



**EULOGIO "AMANG" RODRIGUEZ INSTITUTE OF SCIENCE AND TECHNOLOGY
INFORMATION AND TECHNOLOGY DEPARTMENT**

**E-Serbisyo: Web-Based Document Request and Information System using QR Code
Authentication**

A Project

Presented to the

Information Technology Department

In Partial Fulfillment

of the Course Requirements for

System Integration and Architecture 1

for the first semester of the academic year 2025–2026.

Members:

Chua, Reymart M. – Project Leader

Morales, John Rey F. – Technical Leader

Tomas, TJ C. – Quality Assurance Leader

Kamilan, Diana F. – Documentation Leader

Taña, Lara Mhay Claire L. – Communication Leader

Regualos, Chris Tokiyo T. – Communication Leader

Submitted to:

Prof. Ernanie M. Carlos Jr., MIT

Section

BSINFOTECH 3C

Date of Submission:

1/16/2026



CHAPTER 1	7
1.1 Project Name	7
1.2 Project Members	7
1.3 Project Duration and Activities	7
1.4 Background	8
1.5 Scope	8
1.5.1 Included	8
1.5.2 Excluded	9
1.6 Definitions, Acronyms, and Abbreviations	9
CHAPTER 2	14
2.1 Functional Requirements	14
2.1.1 Functional Requirements Descriptions	15
2.2 Non-Functional Requirements	30
2.3 Use Cases	31
2.3.1 Use Case Diagrams	34
2.4 System Requirements	35
2.4.1 Software Requirements	35
2.4.2 Hardware Requirements	36
2.4.3 Network and Cloud Service Requirements	36
2.5 Business Rules	37
2.6. Constraints	38
2.7 Assumptions	39
2.8 Dependencies	40
CHAPTER 3	42
3.1 Architecture Diagram	42



3.1.1 Presentation Layer	43
3.1.2. Business Logic Layer: The Processing Engine	45
3.1.3. Data Access Layer (DAL): The Data Gateway	48
3.1.4 Database Layer	49
3.2 Module Descriptions	51
3.3 Data Design	53
3.3.1. Data Model	53
3.3.2 Data Flow Diagram	54
3.3.3. Activity Diagram	55
3.3.4 Database Schema	56
3.3. Physical Arrangement of Devices	59
3.4. Conceptual Design	61
3.5. System Prototype	63
Appendices	82
Work Breakdown Structure	82
Minutes of Meeting 1	83
Weekly Status Report	88
Letter for Beneficiary	91
Weekly Status Report 2	94
Minutes of Meeting 2	97
Minutes of Meeting 3	102
Weekly Status Report 3	107
Initial Proposal Document	114
Minutes of Meeting 4	122
Weekly Status Report 4	126
Minutes of Meeting 5	130



Weekly Status Report 5	135
Presentation of Initial Proposal Document & Weekly Status Report 6	138
Weekly Status Report 7	145
Weekly Status Report 8	148
Minutes of Meeting 7	151
Weekly Status Report 9	154
Weekly Status Report 10	157
Request Letter for Sample Copies of Document	160
Weekly Status Report 11	163
Weekly Status Report 12	166
Weekly Status Report 14	172
Weekly Status Report 15	175
Weekly Status Report 16	177
Weekly Status Report 17	179
Weekly Status Report 18	182
Weekly Status Report 19	185
Weekly Status Report 20	188
List of Figures.....	191
Figure 2.1: Use Case Diagram	34
Figure 3.1: Four-Layered Architecture.....	42
Figure 3.2: Entity-Relation Diagram.....	53
Figure 3.3: Data Flow Diagram.....	54
Figure 3.4: Activity Diagram.....	55
Figure 3.5: Hybrid network	59
Figure 3.6: Proposed Conceptual Design	61
Figure 3.7: Login Form	63



Figure 3.8: Registration Form	63
Figure 3.9: Request Document Forms	64
Figure 3.10: Add New Admin / Resident Form	65
Figure 3.11: Resident Dashboard.....	66
Figure 3.12: Resident Requests Page.....	66
Figure 3.13: Resident's Notifications Page	67
Figure 3.14: Resident's Profile Page	67
Figure 3.15: Admin Dashboard	68
Figure 3.16: Admin Request Queue Page	69
Figure 3.17: Admin Documents Page.....	69
Figure 3.18: Admin's Residents Page.....	70
Figure 3.19: Admin's Announcements Page.....	71
Figure 3.20: Admin's Audit Trail Page.....	72
Figure 3.21: Super Admin Dashboard	73
Figure 3.22: Super Admin Account Management Page.....	74
Figure 3.23: Super Admin Document Monitoring Page.....	74
Figure 3.24: Super Admin's Announcement Page	75
Figure 3.25: Super Admin's Audit Trail Page	76
Figure 3.26: Super Admin's System Settings Page.....	77
Figure 3.27: Landing Page / Home Page	79
Figure 3.28: Dialogs/Popups/Modals.....	80
Figure 3.29: Whole Figma Canvas.....	81
 List of Tables	 192
Table 2.1 Functional Requirements	14
Table 3.1: User, Interface Descriptions, and Context	43



Table 3.2: Core Logic Component, System Action, and Proposal Benefit	46
Table 3.3: Data Access Layer Functions, Technical Action, and Proposal Benefit	48
Table 3.4: Data stored, access requirement and proposal benefit	49
Table 3.5: Module Description.....	51
Table 3.6: List of tables and Attributes	56
 Curriculum Vitae	 193



CHAPTER 1

Project Overview

1.1 Project Name

E-Serbisyo: Web-Based Document Request and Information System using QR Code Authentication

1.2 Project Members

- Chua, Reymart M., Project Leader
- Morales, John Rey F. Technical Leader
- Tomas, TJ C., Quality Assurance Leader
- Taña, Lara Mhay Claire L., Communication Leader
- Kamilan, Diana F., Documentation Leader
- Regualos, Chris Tokiyo T., Communication Leader

1.3 Project Duration and Activities

The Data Gathering, Planning, and Designing phase of the project will take place during the first semester of School Year 2025–2026, during this phase, the team will conduct requirements analysis, gather data through interviews or surveys, and create detailed plans for the project. This will include defining the project scope, creating system designs such as wireframes and database schemas, and outlining the overall architecture of the system.

The Development, Implementation, Testing, and Integration phase will be carried out during the first semester of School Year 2026–2027. In this phase, the actual coding and system development will take place, followed by implementation of the system. After development, the system will undergo rigorous testing, including unit testing, integration testing, and user acceptance testing. The final step in this phase will involve integrating the system into the intended environment and preparing documentation for final evaluation or presentation



1.4 Background

The current system for document requests and information dissemination at Barangay Malaya is characterized by significant inefficiency, which is a common issue in local governance relying on manual processes. Specifically, residents endure long queues and limited accessibility due to the entirely manual processing of official documents, while the barangay staff suffers from the time-consuming and repetitive task of manually editing each certificate and individually searching through disorganized physical folders to retrieve resident records. Furthermore, the absence of a centralized official website severely restricts communication, preventing the timely dissemination of important announcements and programs to the residents and contributing to the potential spread of misinformation; this project proposes a Web-based Document Request and Information System to fully digitize the document request-and-verification workflow, establish a centralized communication channel, and integrate QR Code Authentication to secure documents against fraud, thereby replacing the current slow, error-prone manual process with a fast, secure, and modern digital solution accessible to both staff and residents on any modern web browser.

1.5 Scope

1.5.1 Included

- Admin (Barangay Official) Login System
- Residents Login/Registration System
- Full CRUD support for Super Admin
- Database For Storing of User Data and Information
- Advanced analytics and report generation tools for performance monitoring
- Record History Request
- Audit Trail for logging all activities on the system for future report
- Email Authentication and Notification
- Backup and Restore System for Fallback
- Responsive UI Design
- Encrypted Data and Password Hashing using AES256 and SHA256
- Super Admin Settings Configuration for Data and Account Management
- Import/Export Features for Document and Data Management



- System Feature Toggle (ON/OFF Feature) And System Configuration/Settings
- Automatic Document Generation for Official Documents
- QR Code Document Verification / QR Code Authentication
- Online Website for Realtime Announcements and Programs

1.5.2 Excluded

- Garbage Collection Schedule
- Online Payment
- SMS Announcements
- Mobile application of the system
- Real-time messaging or live chat support between residents and officials
- Complaints Forms and Resolutions Documents
- Digital or electronic signature implementation for document approval
- AI Integration

1.6 Definitions, Acronyms, and Abbreviations

- **QR Code** (Quick Response Code): A two-dimensional barcode that can be rapidly scanned by a smartphone camera, functioning as a compact data carrier often used for storing URLs or other critical information, which, in this project, is central to the **instant and secure verification** of official documents.
- **Personal Computer (PC)**: An electronic digital device intended for use by one person at a time, encompassing devices like desktops, laptops, smartphones, or tablets, and serving as the primary hardware through which both residents and staff will securely access the web-based system.
- **MySQL/MariaDB**: A widely-used, open-source Database Management System (DBMS) that acts as the central repository for all critical system records, including resident profiles, document request statuses, and official data, allowing for quick digital storage, retrieval, and organized management of information.
- **Cloudflare**: A global, integrated cloud network service designed to enhance the security and performance of websites by operating as a Content Delivery Network (CDN) to cache content closer to users and as a **security shield** to protect the system from online threats.



- **CDN (Content Delivery Network):** a group of geographically distributed servers that work together to deliver web content more quickly by caching copies of the content closer to users.
- **Node.js:** An open-source, cross-platform JavaScript runtime environment that permits developers to execute JavaScript code on the server-side (outside of a web browser), which is fundamental to building the project's powerful and scalable **back-end** application.
- **MariaDB Connector:** A specific driver or library that establishes and manages the read and write connection between the system's back-end logic and the central MariaDB database, ensuring that application functions can access, modify, and store data accurately.
- **React.js:** A popular, open-source JavaScript library maintained by Facebook, which is utilized for efficiently building the system's responsive and dynamic user interfaces (UIs), forming the technological core of the Front-End.
- **JavaScript:** A versatile, high-level programming language used to create interactive and dynamic content on the web, serving as the core language for both the Front-End (React.js) and the Back-End (Node.js/Express.js) of the application.
- **Material-UI (MUI):** A comprehensive, open-source UI Component Library for React that implements Google's Material Design guidelines, providing a set of pre-built, aesthetically pleasing, and functional UI elements to ensure a consistent, modern, and user-friendly appearance for the system.
- **Express.js:** A minimal and flexible Node.js web application framework that provides a robust set of features specifically for building the server-side logic and Application Programming Interfaces (APIs) of the system.
- **Web Application Framework:** A structured software platform that provides the standard organization, tools, and pre-written code (or libraries) necessary to streamline the development of complex web applications, such as the proposed system.
- **Front-End:** The client-side portion of the application that users directly see and interact with via a web browser, encompassing all visual elements, interactive features, and presentation logic.



- **Back-End:** The server-side infrastructure that operates behind the scenes, responsible for crucial operations such as managing the MySQL/MariaDB database, executing server-side logic, processing user requests, and handling security/authentication.
- **Application Programming Interface (API):** A set of defined rules and protocols that enables the Front-End and Back-End of the system to securely and efficiently communicate with each other, allowing the UI to send data requests to the server and receive information in return.
- **Portable Document Format (PDF):** A file format developed by Adobe that guarantees a document's layout, fonts, and formatting remain consistent and preserved across virtually any software, hardware, or operating system, making it the standard output format for all automatically generated official documents.
- **Website:** A related collection of web pages that are hosted under a single domain name, which, in this project, will serve as the official, centralized platform for Barangay Malaya's document requests, announcements, and key information.
- **System:** A collection of interconnected and interdependent components (software, hardware, and processes) that work together cohesively to achieve a defined objective, which is to digitize and streamline the barangay's administrative services.
- **Runtime Environment:** The complete software layer, including necessary libraries, services, and the Node.js engine, that establishes the foundational context required for a program to execute smoothly and manage all system-level operations like memory and network access in real-time.
- **UI Component Library:** A curated collection of ready-made, modular user interface elements (e.g., buttons, forms, navigation bars) that developers use to accelerate development and maintain a uniform design language across an application.
- **Database Management System (DBMS):** A software suite, like MySQL or MariaDB, specifically designed to allow authorized users and applications to efficiently create, manage, store, organize, and retrieve vast amounts of data from a database.
- **Secure Tunneling:** A networking security practice that involves encapsulating and encrypting data packets to create a private, authenticated "tunnel" for transmission over a public network like the internet, ensuring data privacy and integrity during communication between the application and external services.



- **Object Storage:** A modern data storage architecture that manages unstructured data as self-contained objects (each with data, metadata, and a unique identifier), offering a highly scalable, durable, and cost-effective way to store large amounts of non-database files, such as generated PDF documents.
- **SSL/TLS Certificate** (Secure Sockets Layer/Transport Layer Security Certificate): A digital security file installed on the web server that serves two key functions: authenticating the website's identity and enabling the necessary encryption for a secure HTTPS connection between the resident's browser and the system.
- **Database Server:** A powerful, networked computer or system specifically dedicated to hosting, managing, and providing highly available access to the central MySQL/MariaDB databases and their associated DBMS software.
- **Automatic Document Generation:** A core, time-saving functionality that enables the system to dynamically create complete, official documents (like clearances or certificates) by instantly merging a predefined PDF template with the most current resident data, request details, and official signature information fetched directly from the central database.
- **CRUD (Create, Retrieve, Update, Delete):** the four basic and essential operations for interacting with persistent data in databases and applications. These operations are fundamental for managing data, allowing users to add new records, retrieve existing ones, modify them, and remove them.
- **AI (Artificial Intelligence):** the ability of computer systems to perform tasks that typically require human intelligence, such as learning, problem-solving, and decision-making
- **Hashing:** a process of converting data of any size into a fixed-length string of characters, called a hash or hash value, using a mathematical algorithm called a hash function
- **Encryption:** the process of converting readable data (plaintext) into unreadable code (ciphertext) to prevent unauthorized access.
- **SHA256:** a cryptographic hash function that takes any input data and produces a fixed-size 256-bit (32-byte) "hash" or "fingerprint"



- **AES256:** the Advanced Encryption Standard with a 256-bit key, a highly secure symmetric encryption algorithm used to scramble data into an unreadable format



CHAPTER 2

2.1 Functional Requirements

Table 2.1 Functional Requirements

ID	Requirement Description	Priority
FR1	The system should provide secure login for Barangay Officials.	High
FR2	The system shall manage Resident accounts (Login/Registration).	High
FR3	The system shall store all User Data and Information in a secure database.	High
FR4	The system shall Automatically Generate Official Documents.	High
FR5	The system shall publish Realtime Announcements on a public website.	High
FR6	The system should use AES256 for data encryption and SHA256 for password hashing.	High
FR7	The system shall enable QR Code Verification for document authenticity.	High
FR8	The system shall allow the Super Admin Full CRUD capability over all data.	High
FR9	The system should provide Super Admin Settings for data and account management.	High
FR10	The system should have a Responsive UI Design for all devices.	Medium
FR11	The system shall perform Data Backup and Restore operations.	Medium
FR12	The system shall maintain a Record History of all document requests.	Medium
FR13	The system should maintain an Audit Trail for all users and system activities.	Medium



ID	Requirement Description	Priority
FR14	The system should use Email for Authentication and Notifications.	Medium
FR15	The system shall generate Advanced Analytics and Performance Reports.	Low
FR16	The system shall support Import and Export of Documents and Data.	Low
FR17	The system shall allow administrators to Toggle System Features (ON/OFF).	Low

2.1.1 Functional Requirements Descriptions

Item	Description
Req ID	FR1
Description	The system shall provide secure login for Barangay Officials
Priority	High
Functional Requirement Description	<ul style="list-style-type: none">• The system must authenticate the admin's username and password.• The system must assign specific roles/permissions upon successful login (e.g., Secretary, Treasurer).• The system must restrict access to system settings and resident data until login is successful.• All login attempts (success/failure) must be registered in the logs.
Acceptance criteria	<ol style="list-style-type: none">1. Authentication: A designated test user (e.g., 'Secretary' role) must successfully log in with their credentials and be correctly redirected to their role-specific dashboard.2. Account Provisioning (Creation): The Super Admin must successfully create a new Official account, assign a role



	<p>(e.g., 'Treasurer') and verify the new user can log in with the initial credentials.</p> <p>3. Validation: The system must validate login input fields to prevent SQL injection or other injection attacks, and the system must deny access to the official dashboard URL when credentials are invalid or missing (unauthenticated access).</p>
--	--

Item	Description
Req ID	FR2
Description	The system shall manage Resident accounts (Login/Registration).
Priority	High
Functional Requirement Description	<ul style="list-style-type: none">• The system must allow new users to register with mandatory personal details (Name, Address, Contact).• The system must require Barangay Official approval before a registered resident account is fully active.• The system must allow registered residents to log in to access the document request module.
Acceptance criteria	<p>1. Authentication (Login): A resident whose account has been verified and approved by an Official must successfully log in with their credentials and be granted access to the resident portal and request forms.</p> <p>2. Account Creation (Registration): A new resident must successfully complete the registration form and, upon submission, be assigned a 'pending verification' status.</p> <p>3. Validation: The system must prevent the creation of a duplicate account when a primary unique identifier (such as a combination of full name and address) is detected during the registration attempt.</p>



Item	Description
Req ID	FR3
Description	The system shall store all User Data and Information in a secure database.
Priority	High
Functional Requirement Description	<ul style="list-style-type: none">• The system must maintain separate tables for Residents, Barangay Officials (Admins), and Super Admin.• The system must store resident profile information (e.g., Name, Address, Contact Info) and request details.• The system must enforce data type and length constraints on all input fields to ensure integrity.
Acceptance criteria	<p>1. Authentication (Data Retrieval): A Barangay Official must successfully retrieve and view a resident's complete and accurate profile information from the database via the secure Admin panel, demonstrating authorized access to the stored data.</p> <p>2. Account Creation (Storage): New resident profile data entered through the registration form is correctly and permanently saved to the designated 'Residents' table in the database, with fields mapping correctly to the data model.</p> <p>3. Validation (Integrity): The system must enforce data integrity by preventing the saving of a record where a mandatory constraint is violated (e.g., a required field like 'Address' is blank, or the data exceeds the defined field length).</p>

Item	Description
Req ID	FR4



Description	The system shall Automatically Generate Official Documents.
Priority	High
Functional Requirement Description	<ul style="list-style-type: none">• The system must populate document templates (e.g., Barangay Clearance, Certificate of Residency) with resident data and official details.• The system must generate the document in a standard, non-editable format (e.g., PDF).• The system must attach the required digital signatures and the verification QR code upon final approval.
Acceptance criteria	<ol style="list-style-type: none">1. Authentication (Trigger): The system must automatically trigger the document generation process immediately upon a Barangay Official successfully executing the 'Approve Request' action in the system.2. Creation (Generation): The system must successfully generate a final, non-editable PDF file that incorporates the standard Barangay letterhead, the required digital signatures, and the unique QR code.3. Validation (Data Accuracy): The generated PDF must accurately populate all variable fields (e.g., resident name, address, request purpose, and issue date) using the latest, verified data retrieved from the request record and the system's official settings.

Item	Description
Req ID	FR5
Description	The system shall publish Realtime Announcements on a public website.
Priority	High



Functional Requirement Description	<ul style="list-style-type: none">• The system must provide an Admin interface for creating, editing, and publishing announcements/programs.• The system must display published announcements on the public-facing website immediately.• The system must allow officials to set a start and end date for each announcement's visibility.
Acceptance criteria	<ol style="list-style-type: none">1. Authentication (Authorization): A user with the Official or Super Admin role must successfully create an announcement and set its status to 'Published' via the secure Admin interface.2. Account Creation (Visibility): The newly published announcement must appear on the public-facing website within five (5) seconds of the publishing action being completed by the authorized user.3. Validation (Time Constraint): An announcement must automatically disappear from the public website view immediately after its designated End Date and Time has passed, demonstrating the time-based validation of visibility.

Item	Description
Req ID	FR6
Description	The system should use AES256 for data encryption and SHA256 for password hashing.
Priority	High
Functional Requirement Description	<ul style="list-style-type: none">• The system must encrypt sensitive resident data (e.g., contact numbers) using AES256.• The system must hash all user passwords (for Admin and Residents) using SHA256 before storage.



	<ul style="list-style-type: none">The system must use HTTPS/SSL for all data transmission between the client and the server.
Acceptance criteria	<ol style="list-style-type: none">1. Authentication (Password Hashing): Upon login, the system successfully verifies the user's password against the stored database value, which must be stored as a SHA256-hashed string and not plain text.2. Account Creation (Data Encryption): A newly registered resident's sensitive data (e.g., contact number or birthdate) is verified to be stored in the database in a reversible AES256-encrypted format.3. Validation (Security Protocol): All system portals (public website, resident login, and admin dashboard) must load exclusively over HTTPS/SSL and attempts to access the site via HTTP are automatically redirected.

Item	Description
Req ID	FR7
Description	The system shall enable QR Code Verification for document authenticity.
Priority	High
Functional Requirement Description	<ul style="list-style-type: none">The system must embed a unique, verifiable QR code on every final official document generated.The system must provide a public-facing page or utility for scanning the QR code.The system must display the document's true status (e.g., 'Valid and Issued', 'Revoked', 'Invalid') when the QR code is scanned.
Acceptance criteria	<ol style="list-style-type: none">1. Authentication (Verification): Scanning the QR code on a valid, issued document must successfully link to the public



	<p>verification page and display the document's Unique ID, Issue Date, and a 'Valid and Issued' status.</p> <p>2. Account Creation (Generation): Every document generated by the system (FR4) must contain a unique, scannable QR code that encodes the document's specific verification URL or ID.</p> <p>3. Validation (Status Change): For a document that has been officially revoked by the Super Admin, scanning the QR code must correctly display the status of 'Revoked' or 'Invalid' to the public user.</p>
--	--

Item	Description
Req ID	FR8
Description	The system shall allow the Super Admin Full CRUD capability over all data.
Priority	High
Functional Requirement Description	<ul style="list-style-type: none">The Super Admin must be able to Create, Read, Update, and Delete any record in the system (e.g., Admin accounts, resident profiles, announcements).This capability must be secured and restricted solely to the Super Admin role.
Acceptance criteria	<p>1. Authentication (Authorization): A user successfully logged in with the Super Admin role must be able to access the restricted data management modules (e.g., system settings, user account lists) that are unavailable to other officials.</p> <p>2. Account Creation (CRUD): The Super Admin must successfully demonstrate full CRUD functionality (Create, Read, Update, and Delete) on a target dataset, such as</p>



	<p>creating a new Official account, editing its details, and then permanently deleting it.</p> <p>3. Validation (Access Control): A user logged in with a non-Super Admin role (e.g., Secretary or Treasurer) must be blocked with an explicit authorization error when attempting to access the restricted Super Admin management modules.</p>
--	---

Item	Description
Req ID	FR9
Description	The system should provide Super Admin Settings for data and account management.
Priority	High
Functional Requirement Description	<ul style="list-style-type: none">The Super Admin must be able to configure system-wide settings, such as Barangay Name and Contact Details, which appear on official documents.Super Admin must be able to manage Admin and resident accounts (e.g., block/unblock users).
Acceptance criteria	<p>1. Authentication (Access): A user logged in as the Super Admin must successfully access the dedicated 'System Configuration' panel, which must be inaccessible to all other Official roles.</p> <p>2. Account Creation (Configuration): The Super Admin must successfully update a core system variable (e.g., the official contact number), and this change must be immediately and correctly reflected on newly generated documents and the public website.</p> <p>3. Validation (Account Enforcement): The Super Admin must successfully deactivate a resident's account, and the</p>



	system must immediately prevent that specific resident from logging in or submitting new requests.
--	--

Item	Description
Req ID	FR10
Description	The system should have a Responsive UI Design for all devices.
Priority	Medium
Functional Requirement Description	<ul style="list-style-type: none">• The user interface must automatically adjust to fit screen sizes ranging from mobile devices (320px width) to large desktop monitors.• The system must ensure all buttons and form fields remain accessible and usable on touch-based devices.
Acceptance criteria	<ol style="list-style-type: none">1. Authentication (Access): A user (either Resident or Admin) must be able to access and successfully log in to their respective portal using a mobile device (under 450px width) without encountering horizontal scrolling.2. Account Creation (Usability): The document request form must be fully functional and error-free on a tablet device in both portrait and landscape modes, ensuring all input fields and buttons are tap-accessible.3. Validation (Interface Integrity): The main Admin Dashboard must be correctly displayed on a wide desktop monitor (1920px width) and must not exhibit misplaced or overlapping elements when resizing the browser window between breakpoints.

Item	Description
Req ID	FR11



Description	The system shall perform Data Backup and Restore operations.
Priority	Medium
Functional Requirement Description	<ul style="list-style-type: none">• The system must allow the Super Admin to manually initiate a full database and file backup.• The system must allow the Super Admin to restore the system from a selected backup file.
Acceptance criteria	<ol style="list-style-type: none">1. Authentication (Authorization): A user logged in as the Super Admin must successfully access and initiate the dedicated 'Backup and Restore' utility, which must be secured and inaccessible to all other Official roles.2. Creation (Backup Generation): The system must successfully execute the backup utility, resulting in a compressed, time-stamped file that demonstrably contains the full database schema and data as well as all associated system files.3. Validation (Restore Capability): After purposefully modifying a core system setting (e.g., changing the Barangay Name), restoring the system from the latest backup file must correctly revert that core system setting back to its original state.

Item	Description
Req ID	FR12
Description	The system shall maintain a Record History of all document requests.
Priority	Medium



Functional Requirement Description	<ul style="list-style-type: none">• The system must record every state change for a document request (e.g., Pending, Processing, Approved, Rejected).• The system must capture the date/time and the official responsible for each status change.• The system must keep the record information for as long as it is stored• The system must be ready to fetch the information in case of reprint or viewing the details
Acceptance criteria	<ol style="list-style-type: none">1. Authentication (Access): A Barangay Official must successfully access the details of any completed request and view a secure, chronological history log that includes the timestamp and the ID of the official who executed each step.2. Action Creation (Logging): When any status is changed (e.g., 'Payment Validated' to 'Approved'), the system must automatically create a new, permanent record in the history log that accurately captures the new status and the authorized official's identity.3. Validation (Completeness): For a rejected request, the history log must enforce and display a mandatory rejection reason recorded by the official, ensuring the completeness of the transaction record.

Item	Description
Req ID	FR13
Description	The system should maintain an Audit Trail for all users and system activities.
Priority	Medium



Functional Requirement Description	<ul style="list-style-type: none">The system must log on to every action performed by Admin and Super Admin users (e.g., creating a user, deleting a record, changing settings).Each log entry must include the User ID, Action Performed, Timestamp (PH time), and IP Address.
Acceptance criteria	<ol style="list-style-type: none">1. Authentication (Logging Trigger): An Admin or Super Admin must successfully perform a critical data modification action (e.g., editing a resident's profile), and the system must immediately and successfully record this action in the Audit Trail.2. Action Creation (Log Content): The recorded log entry must correctly and permanently contain the unique User ID, the specific Action Performed, the IP Address, and the accurate Timestamp in PH Time (GMT+8).3. Validation (Forensic Search): The Super Admin must successfully access and search for the Audit Trail to retrieve all activities performed by a specific User ID or within a specific date and time range.

Item	Description
Req ID	FR14
Description	The system should use Email for Authentication and Notifications.
Priority	Medium
Functional Requirement Description	<ul style="list-style-type: none">The system must send an email to residents when their document request status is updated.The system must send a password reset link to a user's registered email address for account recovery.
Acceptance criteria	<ol style="list-style-type: none">1. Authentication (Password Reset): A user who initiates a password recovery must successfully receive a unique,



	<p>single-use, time-limited token/link via email to their registered address, allowing them to authenticate and change their password.</p> <p>2. Account Creation (Notification): When a Barangay Official performs a key status update (e.g., changing the status to 'Approved'), the system must automatically generate and send a structured email notification to the residents detailing the change.</p> <p>3. Validation (Token Expiration): The password reset token/link sent via email must be validated by the system and become invalid if it is not used within the defined security time frame (e.g., 60 minutes) or if it has been successfully used once.</p>
--	--

Item	Description
Req ID	FR15
Description	The system shall generate Advanced Analytics and Performance Reports.
Priority	Low
Functional Requirement Description	<ul style="list-style-type: none">The system must generate reports on key performance indicators (KPIs) like average document processing time and total requests per month.The system must provide data visualizations (e.g., charts, graphs) of resident demographics and service usage.
Acceptance criteria	<p>1. Authentication (Access): A user with the Official or Super Admin role must successfully access the restricted 'Analytics and Reporting' dashboard.</p> <p>2. Account Creation (Report Generation): The system must successfully generate a quantified report showing a key performance indicator, such as the number of 'Approved'</p>



	<p>vs. 'Rejected' requests for a user-specified date range (e.g., the last quarter).</p> <p>3. Validation (Visualization Accuracy): The dashboard must correctly display a data visualization (e.g., a bar chart or pie chart) that accurately reflects a resident demographic breakdown (e.g., by age group) based on the current data in the database.</p>
--	--

Item	Description
Req ID	FR16
Description	The system shall support Import and Export of Documents and Data.
Priority	Low
Functional Requirement Description	<ul style="list-style-type: none">• The system must allow the export of resident master lists and request history into a standard format (e.g., CSV, Excel).• The system must allow the Super Admin to import resident data via a structured file format.
Acceptance criteria	<p>1. Authentication (Authorization): A user logged in as the Super Admin must successfully access and initiate both the 'Export Data' and 'Import Data' utilities, which must be secured and restricted.</p> <p>2. Account Creation (Export): The Super Admin must successfully export the complete list of all registered residents into a standard, structured format (e.g., a CSV or Excel file) that accurately reflects the database columns.</p> <p>3. Validation (Import): The system must successfully import a batch of new resident records from a structured import file, creating the records in the database and logging in any errors for records that violate data integrity rules.</p>



Item	Description
Req ID	FR17
Description	The system shall allow administrators to Toggle System Features (ON/OFF).
Priority	Low
Functional Requirement Description	<ul style="list-style-type: none">• The system must provide a configuration screen where Super Admin can enable or disable major system modules (e.g., Resident Registration).• Disabling a module must immediately hide the feature from the relevant user interface.
Acceptance criteria	<ol style="list-style-type: none">1. Authentication (Toggling): A user logged in as the Super Admin must successfully access the feature configuration panel and change the state of a major module (e.g., set 'Resident Registration' to OFF).2. Account Creation (Toggling Effect): Immediately after the Super Admin toggles a feature to OFF, the corresponding feature (e.g., the public registration link) must be instantly hidden or removed from the relevant user interface.3. Validation (Access Denial): While the feature is toggled OFF, any user attempting to directly access the module's URL (e.g., the registration endpoint) must be blocked and redirected to the public homepage or a non-access page.



2.2 Non-Functional Requirements

ID	Requirement Description	Priority
NFR1	The system shall be available 99.98% of the time.	High
NFR2	The system shall be ready for offline mode	High
NFR3	The system shall respond to user requests within 2 seconds.	High
NFR4	The system shall protect all resident personal information (PI) and official records in compliance with data privacy regulations.	High
NFR5	The system shall utilize HTTPS/SSL for all data transmission between the client and the server.	High
NFR6	The system interface shall be intuitive for non-technical staff (Barangay Officials).	Medium
NFR7	The system documentation shall allow new developers to efficiently maintain and extend the codebase.	Medium
NFR8	The system documentation shall allow new developers to efficiently maintain and extend the codebase.	Medium
NFR9	The system shall support the latest two major versions of Chrome, Firefox, and Edge browsers.	Medium
NFR10	The system shall provide meaningful and helpful error messages that guide the user to resolution.	Medium
NFR11	The system shall employ a consistent color palette, typography, and layout across all Admin and Resident portals.	Medium
NFR12	The system shall be able to support a 50% increase in resident accounts and document requests within one year without performance degradation.	Low



NFR13	The system shall be able to be deployed and run on any standard Linux-based or Windows-based web server environment (Portability).	Low
NFR14	The system shall support storing and displaying dates and times in the PH Time (PST/GMT+8) format.	Low
NFR15	The system shall comply with WCAG 2.1 Level AA standards to ensure PWD Accessibility.	Low

2.3 Use Cases

Use Case ID	Name	Brief Description	Actors
UC01	Register	Allows a resident to create a new account.	Resident
UC02	Login	Authenticates user credentials to access the system.	Resident, Admin, Super Admin
UC03	View Website	View public contents and general information.	Resident
UC04	Request Document	Submit a request for an official document.	Resident
UC05	Cancel Active Request	Cancel a previously submitted document request.	Resident
UC06	Retrieve Rejection Reason	View the reason why a request was rejected.	Resident



**EULOGIO "AMANG" RODRIGUEZ INSTITUTE OF SCIENCE AND TECHNOLOGY
INFORMATION AND TECHNOLOGY DEPARTMENT**

UC07	Track Request Status	Monitor the progress of a submitted request.	Resident
UC08	Verify Document Authenticity	Validate if a document is legitimate.	Resident
UC09	View Requested Documents	Display list of documents previously requested.	Resident
UC10	Receive Cancel Request	Admin receives and reviews cancellation requests.	Admin
UC11	Update Verification of Rejection Status	Admin updates verification or rejection status of a request.	Admin
UC12	Receive Requested Documents	Admin receives and views new document requests.	Admin
UC13	Process Requested Documents	Admin evaluates and processes document requests.	Admin
UC14	Filter and Prioritize Queue	Organize, sort, and prioritize request queue.	Admin
UC15	Generate OR Number for Paid Document	Generate an OR (Official Receipt) number for paid requests.	Admin



**EULOGIO "AMANG" RODRIGUEZ INSTITUTE OF SCIENCE AND TECHNOLOGY
INFORMATION AND TECHNOLOGY DEPARTMENT**

UC16	Generate and Issue Document	Create, finalize, and officially issue requested documents.	Admin
UC17	Record Document Claimed	Admin records when residents pick up their documents.	Admin
UC18	Manage Public Contents	Create, update, or delete public website contents.	Admin, Super Admin
UC19	Revoke/Correct Issued Document	Modify or revoke documents previously issued.	Admin
UC20	Manage Staff User Account	Add, edit, or remove admin user accounts.	Super Admin
UC21	System Configuration Management	Manage system-wide settings and configurations.	Super Admin
UC22	Perform Data Backup/Restoration	Backup or restore system data.	Super Admin
UC23	Import/Export System Data	Import or export system records and databases.	Super Admin
UC24	Review System Report	View generated reports and analytics.	Super Admin
UC25	Review Document Audit Trail	View logs of document-related transactions.	Super Admin

2.3.1 Use Case Diagrams



Figure 2.1: Use Case Diagram



2.4 System Requirements

2.4.1 Software Requirements

Category	Component	Version / Details
Backend Runtime	Node.js	v24.10.0 or higher (Active LTS) with npm and Express.js v5.1.0 (or latest stable) framework.
Frontend Library	React.js	v19.2
Frontend UI	UI Component Library	Material UI v7.3.4 (or latest stable)
Database (Primary)	Database Management System	MariaDB Server 11.8 (Latest GA) or MySQL 8.0.43.
Database Connector	Node.js Driver	MariaDB Connector-Node.js 3.5.0 (Use latest available stable version).
Web Browser (Client)	Supported Browsers	Chrome, Firefox, Edge (latest 2 versions)
Operating System (Dev)	Development Environment OS	Windows 11 Home Edition (24H2) or 10 LTS Home Edition
Operating System (Server)	Hosting Environment OS	Windows 11 Home Edition (24H2). <i>Requires disabling automatic updates and sleep mode for 24/7 server operation.</i>



Networking Utility (Internal DNS)	DNS Service	Technitium DNS Server is REQUIRED and must be configured for Split-Horizon DNS functionality.
--	-------------	---

2.4.2 Hardware Requirements

Component	Version / Details	
Server CPU (Minimum)	CPU	4 Cores / 4 Threads (e.g., Intel Core i3 10th Gen equivalent or higher) with a Base Clock of 2.0 GHz or faster.
Server RAM (Minimum)	RAM	8GB of DDR4 RAM minimum (16GB recommended for stability) to run the OS, DNS Service, Node.js, and Database concurrently.
Server Storage		256GB SSD (Solid State Drive) minimum for OS, application, and data storage.
Client Device RAM (Minimum)	RAM	4GB for web browser and operating system function.
Server Stability (Non-Data Center)		Uninterruptible Power Supply (UPS) and a reliable cooling system are mandatory for continuous 24/7 operation on a desktop OS.

2.4.3 Network and Cloud Service Requirements

Component	Version / Details	
Internet Connection (Server)	Connection	Stable, high-speed connection (5 Mbps upload minimum) with low latency for Cloudflare Tunnel reliability.



Cloud Services (Domain/Security)	Registered Domain Name (https://gomanilahost.net/) hosted/managed via Cloudflare (for DNS and proxy/CDN).
Cloud Services (Access)	Cloudflare Tunnel for secure, external ingress to the local Windows 11 PC.
Cloud Services (Storage)	Cloudflare R2 for Object Storage (for external file backups and non-database assets).
Network Protocol	Split-Horizon DNS configuration using Technitium DNS is mandatory to route traffic to the LAN IP address (for offline access) and the Cloudflare Tunnel (for online access).

2.5 Business Rules

- Only Barangay Officials (Secretary, Treasurer, and Punong Barangay) and Super Admin are authorized to access, review, and approve or reject document requests.
- Residents must submit complete and valid information, including required identification details (e.g., CTC number, ID details), and a clearly stated purpose of request, before the request can proceed to the verification stage.
- All issued official documents must bear the seal, name, and wet signature of the Punong Barangay to be considered valid.
- Every generated official document must contain the Barangay logo, the Authorized official's name and signature, the Issue Date, and a Unique Identifier (QR Code) for verification.
- All standard fees, processing times, and exemptions will strictly follow Barangay Malaya's Citizen's Charter and Barangay Ordinance No. 002, Series of 2019.
- The system shall not allow the same applicant to have more than one *active* request of the same document type at the same time to avoid duplication.
- All document requests must follow Barangay's standard fees and rules stated in Barangay Ordinance No. 002, Series of 2019 and the Citizen's Charter.



- All payments for documents must be made in person to the Barangay Treasurer, who will issue an official receipt. The treasurer shall then mark the request as paid in the system.
- Requests marked as "Free of Charge" (such as indigency certificates for medical purposes or first-time job seekers) are automatically exempted from payment confirmation.
- Once a document is officially issued, it becomes read-only and cannot be modified, except by the Super Admin for legitimate corrections or revocations.
- Residents and staff must comply with the barangay's official service hours (8:00 AM–5:00 PM) when handling requests, consistent with the Citizen's Charter.
- The barangay shall conduct periodic reviews and updates of fees, templates, and workflows to ensure that the system remains consistent with current local ordinances.
- An issued document is considered valid for a period of six (6) months from the Issue Date, unless otherwise specified by the Citizen's Charter for specific document types (e.g., permits).
- In case of discrepancies between resident information and submitted documents, the barangay staff shall have the right to hold or reject the request pending further validation.
- Residents are responsible for reviewing their information and requesting details before final submission to avoid clerical errors.

2.6. Constraints

- Limited Operating Hours: Transactions can only be handled during official working hours (8:00 AM–5:00 PM) in compliance with the Citizen's Charter, restricting after-hours processing.
- Hardware Dependence: The number of users that can access the system at once depends on the available computers and equipment in the barangay office.
- Cash-Only Payment System: The process currently supports only in-person cash payments confirmed by the treasurer or an authorized staff member; no online payment options are integrated.



- Semi-Automated Process: Certain tasks such as resident verification and payment checking are not yet automated and still need staff intervention.
- Uneditable Issued Documents: Once an official document has been generated, it cannot be modified, except in cases approved and corrected by the Super Admin.
- Internet Stability Issues: The system's performance relies on the barangay's current internet connection, which may slow down or disconnect at times, affecting user access.
- Hardware Dependence: The number of users that can access the system at once depends on the available computers and equipment in the barangay office.
- Access Control Limitation: Only designated officials (Secretary, Treasurer, and Punong Barangay) are permitted to approve or reject requests, delaying processing when they are unavailable.
- Single Active Request Constraint: The system must be constrained to prevent any applicant from having more than one active request for the same document type simultaneously.
- Compliance Requirements: All personal data collected must strictly follow the Data Privacy Act of 2012, which limits how information can be stored and shared.
- Server and Power Interruptions: The system may temporarily go offline during power failures or maintenance activities affecting the barangay's local server.
- Internet Stability Issues: The system's reliable performance is directly dependent on the barangay's current internet connection, which is known to be unstable or slow down at times, potentially affecting real-time user access and data processing.

2.7 Assumptions

- All users (barangay staff and residents) have access to a stable internet connection and a modern web browser (Chrome, Firefox, or Edge).
- Not all residents are familiar with using digital systems or web-based applications; therefore, barangay staff may assist walk-in residents in encoding their requests into the system.
- Some residents may have limited or no access to internet or digital devices, so manual submission of request forms at the barangay office will still be accommodated.



- Barangay officials and staff have basic computer knowledge and can operate the system with minimal training.
- The barangay will maintain a dedicated official email account for sending notifications, updates, and password recovery links.
- The barangay's server or hosting computer will be kept operational during office hours and protected by a reliable power supply.
- Future integration with a barangay resident database or mobile application will be possible without major system redesign.

2.8 Dependencies

- The backend operation depends on a properly installed and configured Node.js runtime (version 24.10.0 or higher) together with npm and the Express.js framework. Any missing or outdated dependencies may prevent the application from running correctly.
- The user interface requires React v19.2 and the Material UI v7.3.4 component set. System stability and design consistency rely on these libraries being compatible with the installed Node.js and browser versions.
- All records and requests are stored in MariaDB 11.8 (or MySQL 8.0.43). The system depends on this database service to be continuously available and properly linked to the backend through the MariaDB Node.js Connector v3.5.0.
- Both the development and hosting machines must use Windows 10 LTS or 11 (24H2). The environment must remain active, with automatic updates and sleep mode disabled to maintain continuous access.
- Internal routing depends on Technitium DNS Server set up with Split-Horizon DNS. This configuration allows staff to access the system via local network addresses while external users connect through the Cloudflare Tunnel.
- A stable internet connection with at least 5 Mbps upload speed is required for Cloudflare services, remote access, and real-time synchronization between the local server and the cloud environment.



- The domain and proxy setup rely on Cloudflare for DNS, security, and content delivery. File backups and attachments are stored in Cloudflare R2 Object Storage, making both services essential for external accessibility and data protection.
- Reliable operation depends on a host device with at least 4 CPU cores, 8 GB of RAM, and 256 GB SSD storage. Lower specifications may slow down data processing and impact response time.
- Because the server runs continuously, uninterrupted power (via UPS) and proper cooling are required to prevent data loss and hardware damage during prolonged use.
- Users must access the platform through updated web browsers—Google Chrome, Mozilla Firefox, or Microsoft Edge—to ensure full support for React-based components and system scripts.
- System notifications and password recovery depend on an active official barangay email account configured with a reliable mail service provider.
- Proper firewall, antivirus, and backup utilities must remain active to safeguard data integrity and comply with the Data Privacy Act of 2012.

CHAPTER 3

3.1 Architecture Diagram

The proposed Barangay Information and Document Request Management System adopt a Four-Layered Client-Server Architecture. This design separates the system into four distinct logical layers: Presentation, Business Logic, Data Access, and Database.

This layered approach ensures that the "Client" (the web browser) is decoupled from the "Server" (the database and logic), allowing for secure data processing, independent scalability, and the ability to function in both online (Cloudflare Tunnel) and offline (Local LAN) environments.

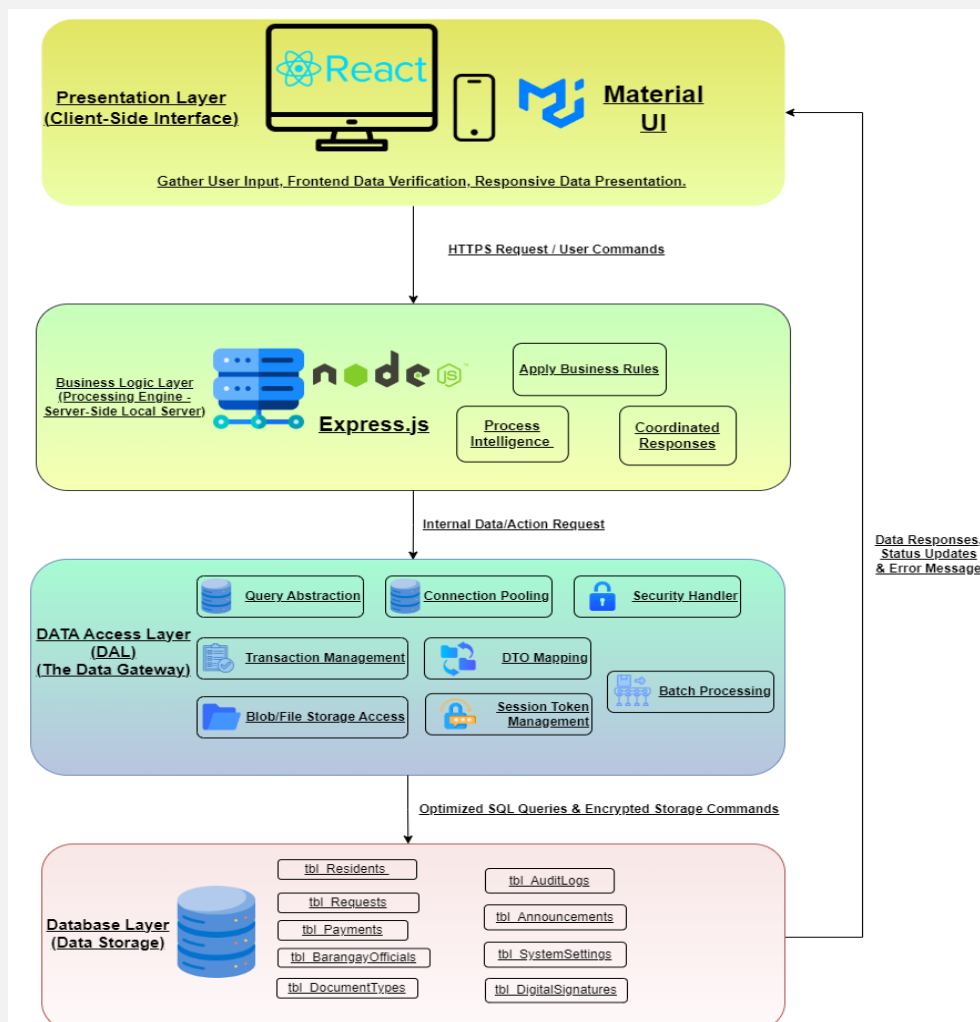


Figure 3.1: Four-Layered Architecture



3.1.1 Presentation Layer

The Presentation Layer acts as the Client-Side interface. It is responsible for gathering user input, verifying data formats on the frontend, and presenting data to the user. It is built using React.js and Material UI, ensuring a responsive experience across mobile devices and desktops.

Table 3.1: User, Interface Descriptions, and Context

User	Interface Descriptions	Context
Resident	A responsive web interface featuring a navigation bar, a "Hero" section with the Barangay logo, and a scrollable timeline for announcement cards. It also includes a grid-layout menu for services and a multi-step form wizard with a status tracker for document applications.	Serves as the primary entry point for residents to navigate the system, view real-time updates on programs or suspensions, and submit or track online document requests.
Resident	A multi-step form interface with progress indicators (e.g., "Step 1: Personal Info," "Step 2: Document Type"). It features dropdown menus for document selection, text fields for personal data, and a file upload area for requirements.	Allow residents to submit requests remotely, ensuring they provide all necessary data (Purpose, ID) before processing.
Resident	A simple search interface with a text input field for the "Reference Number." Upon search, it displays a visual status progress bar (Pending > Processing > Ready).	Allows residents to track the progress of their specific request without needing to log in or visit the hall.
Resident	A public-facing, login-free mobile page. It displays a "Scan Successful" banner followed by the verified document details (Resident	Triggered when a third party scans the QR code on a physical document to verify it is authentic and not forged.



**EULOGIO "AMANG" RODRIGUEZ INSTITUTE OF SCIENCE AND TECHNOLOGY
INFORMATION AND TECHNOLOGY DEPARTMENT**

	Name, Issue Date, Validity) and a green "VALID" checkmark.	
Admin (Secretary)	A desktop-optimized view with a collapsible sidebar navigation and a main content area featuring a "Pending Requests" data table. The table includes columns for Date, Name, Type, and action buttons for "View," "Approve," or "Reject."	The main workspace for staff to manage incoming workflows, verify resident identities, and move requests to the next stage.
Admin (Treasurer)	A secure financial interface displaying a queue of "Approved/Unpaid" requests. It features a transaction form with inputs for "Cash Amount Received" and "OR Number," along with a "Confirm Payment" button and the Treasurer's pre-loaded digital signature before printing.	Dedicated interface for the Treasurer to input face-to-face cash payments and trigger the release of documents.
Admin (Treasurer)	A modal window that renders a digital preview of the Official Receipt, showing the transcription of the amount, the date, and the Treasurer's pre-loaded digital signature before printing.	Ensures the receipt is correct before the physical copy is generated for the resident.
Admin	An internal, high-speed data entry form designed for desktop use. It mimics the resident wizard but allows staff to bypass certain validation steps for quick encoding of offline residents.	It used to accommodate residents who physically visit the hall without an online account.
Super Admin	A secure administrative list displaying all registered system users (Staff and Residents). It includes search filters and	Allows Super Admin to manage access, reset staff



	"Edit/Deactivate" toggle switches for account control.	passwords, or ban abusive resident accounts.
Super Admin	A settings form containing text inputs for "Barangay Name," image uploaders for "Logos/Signatures," and toggle switches to Enable/Disable features like "Online Registration."	Provides control over global system variables and maintenance modes without altering source code.
Super Admin	A read-only grid view displaying a chronological history of system events. Columns include Timestamp, User ID, IP Address, and Action Description (e.g., "Printed Clearance").	A forensic tool to track all activities for security and accountability purposes.
Admin	A data visualization dashboard featuring bar graphs for "Requests per Document Type" and pie charts for "Resident Demographics," with date range pickers for custom reporting.	Visualizes data to help officials analyze trends and service usage

3.1.2. Business Logic Layer: The Processing Engine

The Business Logic Layer represents the Server-Side Application Layer. Hosted on the local server using Node.js and Express.js, this layer processes the "intelligence" of the system. It receives commands from the client, applies business rules (e.g., "Is this resident banned?"), and coordinates responses.



Table 3.2: Core Logic Component, System Action, and Proposal Benefit

Core Logic Component	System Action	Proposal Benefit
Request Validation Engine	Checks if the resident has an active pending request of the same type and validates input data formats.	Prevents spamming of requests and enforces the "One Active Request" rule to avoid backlog.
Fee & Payment Calculator	Determines if a request is "Indigent/Free" or "Paid" based on the resident's profile and document type.	Automates fee assessment, ensuring that indigent residents are correctly exempted from fees as per the Citizen's Charter.
Official Receipt (OR) Generator	Generates a unique, sequential OR number and stamps the Treasurer's digital signature onto a receipt PDF. "Request Received," "Approved," or "Password Reset" events.	Digitizes the cash transaction process, ensuring every payment is tracked and professionally documented
Document Generation Service	Merges resident data into predefined PDF templates (Clearance, Certificate) and embeds a secure QR code.	Automates the typing process, reducing clerical errors and ensuring all issued documents follow the standard format.
QR Code Encryption Service	Encrypts the document's validation URL and unique ID into a QR matrix before stamping it on the PDF.	Ensures that the QR code cannot be forged or tampered with, enabling reliable offline or online verification.



EULOGIO "AMANG" RODRIGUEZ INSTITUTE OF SCIENCE AND TECHNOLOGY
INFORMATION AND TECHNOLOGY DEPARTMENT

Notification Controller	Triggers automated emails via SMTP when a request status changes (e.g., "Pending" -> "Ready for Pickup").	Keeps residents informed in real-time, reducing the need for them to physically visit the hall just to ask for updates.
Authentication & Session Manager	Verifies user credentials (hash comparison) and manages secure login sessions for residents and officials.	Ensuring that only authorized users can access specific features (e.g., only Treasurer can access Payment Dashboard).
Input Sanitization Logic	Scrubs all incoming text fields from the Presentation Layer to remove potential SQL injection scripts.	Protects the database from malicious attacks by neutralizing harmful code before it is processed.
Audit Logging Service	Automatically captures the User_ID, Timestamp, and Action for every critical event (e.g., "Printed Clearance").	Creates an immutable record of all staff actions, promoting accountability and transparency in governance
Backup & Restore Controller	Manages the schedule for database dumps and handles the "Restore" operation from Cloudflare R2 or local disk.	Ensures the system can recover resident data quickly in case of hardware failure or corruption
Content Management Logic	Handles the scheduling, publishing, and automatic expiration of public announcements.	Ensures that old news is automatically removed from the site, keeping the information board current



3.1.3. Data Access Layer (DAL): The Data Gateway

Table 3.3: Data Access Layer Functions, Technical Action, and Proposal Benefit

DAL Function	Technical Action	Proposal Benefit
Query Abstraction	Translates API requests into optimized, parameterized SQL queries.	Shields the core logic from complex SQL syntax, making the code cleaner and easier to update if database rules change.
Connection Pooling	Manages a pool of active database connections to reuse them efficiently instead of opening new ones per user.	Prevents the server from crashing during high traffic by efficiently managing limited database resources.
Transaction Management	Wraps multi-step operations (e.g., Approve Request + Update Log + Deduct Stock) in ACID transactions.	Ensures that if one step fails (e.g., log update fails), the entire action is rolled back to prevent partial data corruption.
Encryption Handler (At Rest)	Encrypts sensitive personal information (PI) like contact numbers using AES256 before insertion.	Ensures that the system adheres to the Data Privacy Act by keeping stored data unreadable to unauthorized access.
Data Sanitization	Escapes special characters in user inputs to prevent SQL Injection attacks at the database driver level.	Provides the final line of defense against injection attacks, ensuring raw SQL commands are never executed directly.
DTO Mapping	Converts raw database rows into clean JSON objects (Data	Strips out unnecessary or sensitive columns (like password



	Transfer Objects) for the Business Layer.	hashes) before sending data up to the logic layer.
Error Handling & Logging	Catches database-level errors (e.g., "Duplicate Entry") and translates them into readable error messages.	Prevents the system from crashing and provides meaningful feedback to the user/admin (e.g., "Resident already exists").
Blob/File Storage Access	Manages the retrieval and storage of binary data like digital signatures and barangay logos.	Ensures that critical assets required for document printing are always accessible to the generator.
Session Token Management	Handles the storage and validation of active user session tokens in a dedicated table or cache.	Allows the system to invalidate the timeout sessions (logout) automatically for security.
Batch Processing	Handles bulk database operations, such as exporting the resident list to CSV/Excel.	Optimizes large data retrievals to prevent system slowdowns during reporting.

3.1.4 Database Layer

Table 3.4: Data stored, access requirement and proposal benefit

Data Stored	Access Requirement	Proposal Benefit
tbl_Residents	High Encryption (AES256); Restricted Admin Access.	Identity Management: Securely stores resident profiles (Name, Address, DOB, Contact) as the single source of truth.



tbl_Requests	Write-Once, Read-Many (WORM).	Transaction History: specific details of every document request (Type, Purpose, Status, Date) for tracking workflow.
tbl_Payments	Treasurer Write Access Only; Linked to Requests.	Financial Audit: Stores Official Receipt (OR) Numbers, Amounts, Payment Dates, and the Treasurer ID who processed it.
tbl_BarangayOfficials	Hashed Passwords; Role-Based Access.	Access Control: Manages staff credentials and defines roles (e.g., Captain, Secretary, Treasurer) and their permissions.
tbl_DocumentTypes	Admin Configurable (Update Pricing/Template).	Standardization: Stores the rules, standard fees, and template layouts for each document (e.g., Clearance = P50, Indigency = Free).
tbl_AuditLogs	Strict Read-Only (No Delete allowed).	Accountability: A permanent security log of every login, update, print job, and payment confirmation performed in the system ³⁰ .
tbl_Announcements	Public Read Access; Admin Write Access.	Communication: Stores the title, body content, images, and validity dates for public news posts ³¹ .
tbl_SystemSettings	Super Admin Access Only.	Configuration: Stores dynamic global settings like "Barangay



		Name," "Logo," "Contact Info," and "Feature Toggles".
tbl_DigitalSignatures	Encrypted Blob Storage.	Security: Securely stores the digital signature images of the Captain and Treasurer to prevent unauthorized usage or forgery.
tbl_PasswordResets	Temporary storage with Expiry.	Recovery: Manages the secure, time-limited tokens sent via email for password recovery, preventing replay attacks.

3.2 Module Descriptions

Table 3.5: Module Description

Module Name	Description	Example Functionality
Resident Portal	The front-end interface for public users to interact with barangay services.	Displays the "Apply for Clearance" button and allows the residents to input their purpose of request.
Admin Dashboard	The control center for Barangay Officials to manage operations.	Displays a list of "Pending Requests" sorted by date, allowing the secretary to click "Approve" or "Reject."
Treasurer Payment Module	A dedicated module for handling financial transactions and receipts.	Allow the Treasurer to search for a resident, input "Paid 50.00", and generate the Official Receipt PDF.
Document Generator	The backend module is responsible for creating the physical file.	Automatically place the Resident's Name and the Barangay Captain's digital signature onto a PDF Certificate.



EULOGIO "AMANG" RODRIGUEZ INSTITUTE OF SCIENCE AND TECHNOLOGY
INFORMATION AND TECHNOLOGY DEPARTMENT

QR Verification Module	Handles the authenticity check of issued documents.	When a QR code is scanned, this module retrieves the document status (Valid/Revoked) from the database and displays it.
Announcement Module	Allows officials to post updates to the public site.	Allows the Secretary to post "No Office on Monday" which immediately appears on the Resident Portal.
User Management Module	Handles registration, login, and role assignment.	Processes the registration of new residents and allows Super Admin to create new staff accounts.
Audit Trail Recorder Module	A background service that logs system activities.	Silently records "Admin J. Dela Cruz printed Clearance for Resident ID 123" into the database.
Reporting & Analytics Module	Generates statistical summaries for decision making.	Create a pie chart showing the percentage of "Indigency" vs "Residency" requests for the month.
Backup Utility Module	Manages data preservation.	Allows the Super Admin to click "Backup Now" to save a copy of the database to the cloud or local drive.
Email Notification Service	Manages outbound communications.	Send the "Your Document is Ready" email to the resident once the Treasurer confirms payment.

3.3 Data Design

The Data Design section is critical; it establishes the formal framework for how the system's information will be structured, managed, and processed. It translates the barangay's conceptual needs into a concrete blueprint that ensures data integrity and consistency.

3.3.1. Data Model

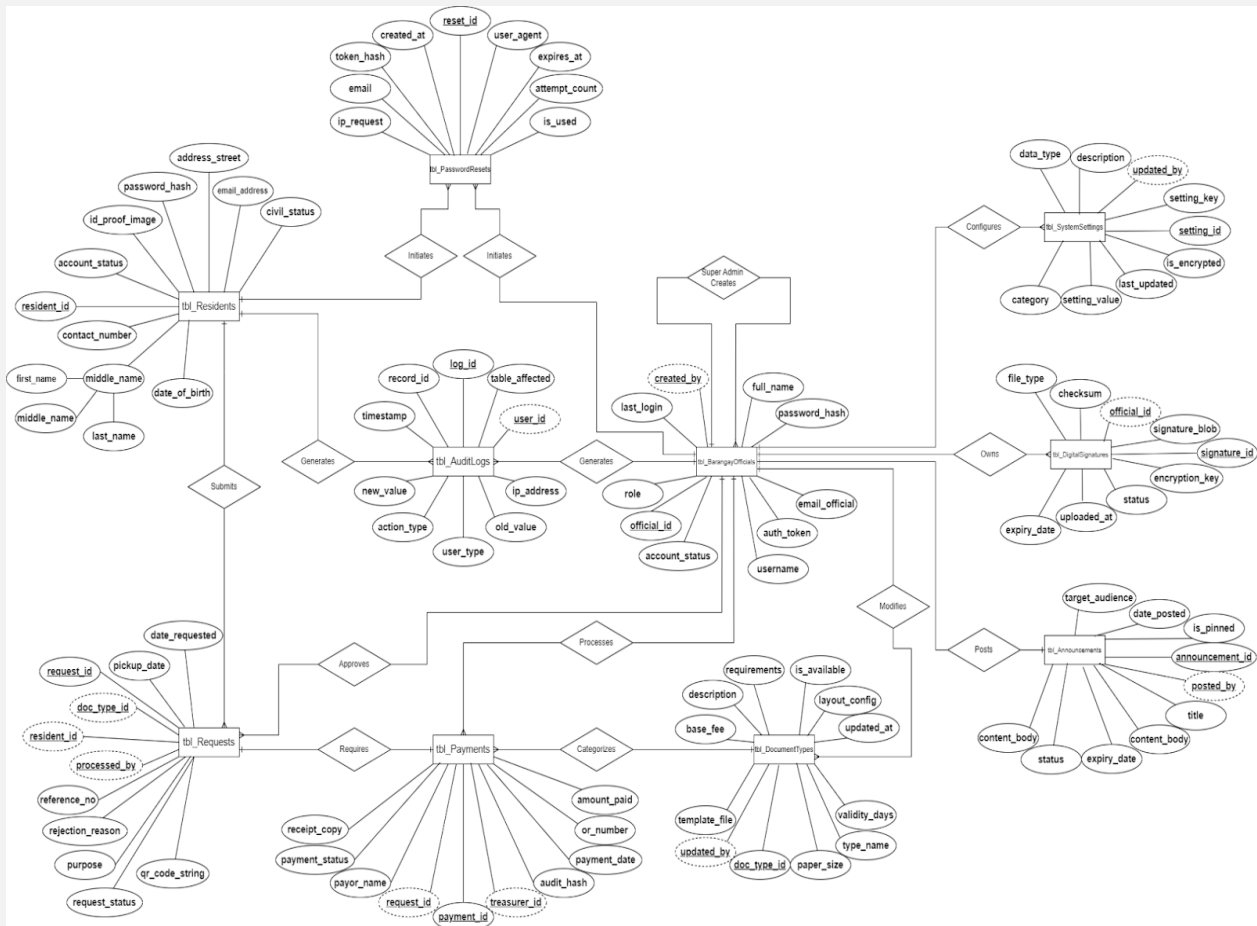


Figure 3.2: Entity-Relation Diagram

Figure 3.2. illustrates the Entity-Relationship Diagram (ERD) for the proposed Web-Based Barangay Document Request and Information System. The schema is anchored by two primary user entities: `tbl_Residents`, which manages constituent demographics and identity verification, and `tbl_BarangayOfficials`, which handles administrative credentials and role-based access. These users interact through a central workflow where residents submit applications via `tbl_Requests`, a table that links specific service requirements defined in `tbl_DocumentTypes` to the corresponding financial transactions recorded in `tbl_Payments`. To

ensure system flexibility and authority, administrative tables such as `tbl_SystemSettings`, `tbl_DigitalSignatures`, and `tbl_Announcements` allow officials to configure settings, manage encrypted signatures, and disseminate public information. Furthermore, data integrity and security are enforced by `tbl_PasswordResets` for account recovery and `tbl_AuditLogs`, which maintains a comprehensive trail of all user actions, timestamps, and data modifications.

3.3.2 Data Flow Diagram

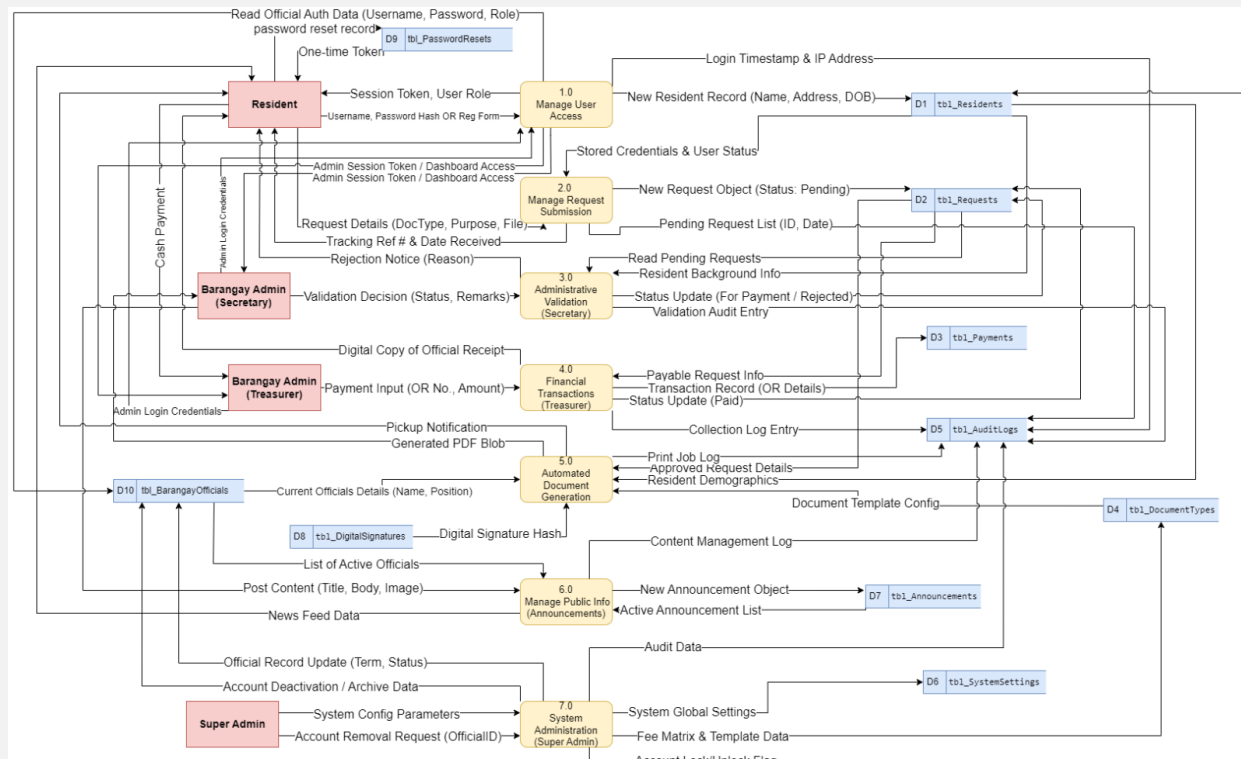


Figure 3.3: Data Flow Diagram

Figure 3.3. illustrates the Level 1 Data Flow Diagram (DFD) of the system, delineating the transformation of data from initial resident access and request submission through the distinct administrative phases of validation by the Secretary and financial processing by the Treasurer. The workflow culminates in Process 5.0 (Automated Document Generation), which synthesizes resident data, payment verification, and encrypted signatures to produce the final document, while simultaneously ensuring system integrity through a centralized Super Admin interface and comprehensive logging into `tbl_AuditLogs` for all transactional activities.

3.3.3. Activity Diagram

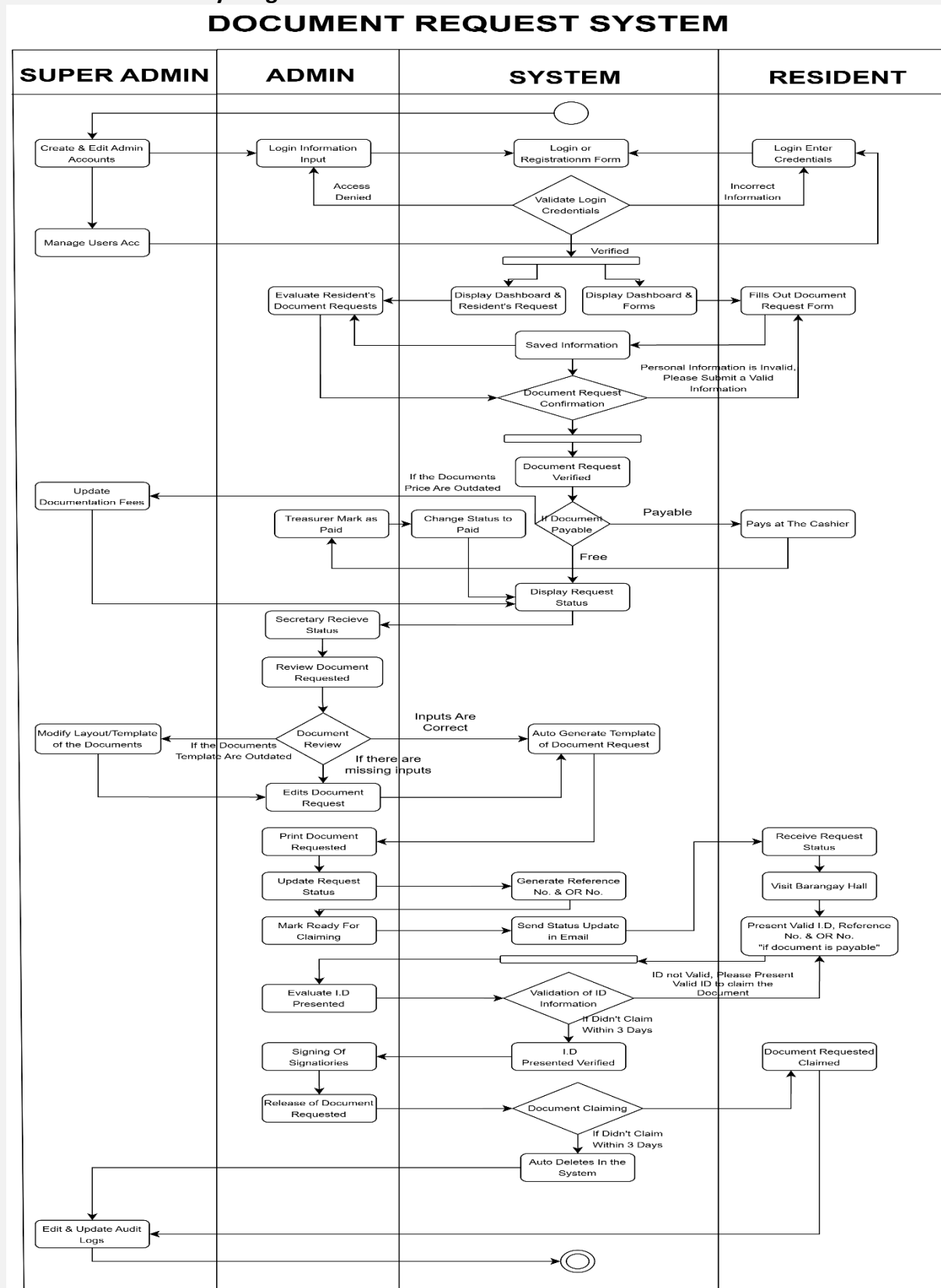


Figure 3.4: Activity Diagram



Figure 3.4. Shows the Activity Diagram model of the Web-Based Document Request and Information System workflow by illustrating the sequence of activities across three responsible entities—the Admin, the Web-Based Information System (the application), and Resident beginning with the Resident opening a browser to access the system and the admin submitting login credentials for authentication. The process continues with the system performing user authentication, which then branches into two parallel actions: displaying the dashboard for the admin and displaying the homepage for the Resident. From the dashboard, the admin can manage and update public information postings within the system, while the Resident simultaneously views accessible public information on the website. The workflow ensures that any updates made by the admin are reflected on the homepage accessed by the Resident. The sequence concludes once the updated public information is fully processed and displayed, marking the end of the activity flow.

3.3.4 Database Schema

Table 3.6: List of tables and Attributes

Table Name	Attributes	Description
tbl_Residents	resident_id (PK), first_name, middle_name, last_name, date_of_birth, address_street, civil_status, email_address, contact_number, password_hash, id_proof_image, account_status	Stores verified resident identity and demographic data.
tbl_Requests	request_id (PK), resident_id (FK), doc_type_id (FK), reference_no, purpose, request_status, rejection_reason, qr_code_string, date_requested, pickup_date, processed_by (FK)	Records resident document requests and their processing workflow.



tbl_Payments	payment_id (PK), request_id (FK), or_number, amount_paid, payment_date, treasurer_id (FK), payment_status, payor_name, receipt_copy, audit_hash	Stores payment transactions linked to specific requests.
tbl_BarangayOfficials	user_id (PK), official_id, full_name, username, email_official, password_hash, role, account_status, auth_token, last_login	Stores barangay staff accounts and role-based access credentials.
tbl_DocumentTypes	doc_type_id (PK), type_name, description, base_fee, requirements, template_file, layout_config, paper_size, validity_days, is_available, updated_by (FK), updated_at	Stores document categories, fees, templates, and configuration rules.
tbl_DigitalSignatures	signature_id (PK), official_id (FK), signature_blob, file_type, checksum, encryption_key, status, uploaded_at, expiry_date	Stores encrypted digital signatures of authorized officials.
tbl_Announcements	announcement_id (PK), title, content_body, target_audience, image_path, is_pinned, status, date_posted, expiry_date, posted_by (FK)	Stores barangay announcements and public advisories.
tbl_SystemSettings	setting_id (PK), setting_key, setting_value, description, category, data_type, is_encrypted, last_updated, updated_by (FK)	Stores system configuration values such as branding and contact info.



tbl_AuditLogs	log_id (PK), user_id (FK), table_affected, record_id, action_type, old_value, new_value, timestamp, ip_address, user_type	Records system activity for security and accountability.
tbl_PasswordReset	reset_id (PK), user_id (FK), token_hash, email, ip_request, user_agent, expires_at, attempt_count, is_used, created_at	Stores temporary password reset tokens with expiration tracking.

Table 3.6 defines the database structure. It centers on tbl_Residents, which connects to tbl_Requests and tbl_Payments for transaction tracking. tbl_BarangayOfficials and tbl_DocumentTypes manage administration, while tbl_DigitalSignatures, tbl_AuditLogs, and tbl_PasswordReset ensure security. Finally, tbl_SystemSettings and tbl_Announcements support general operations.

3.3. Physical Arrangement of Devices

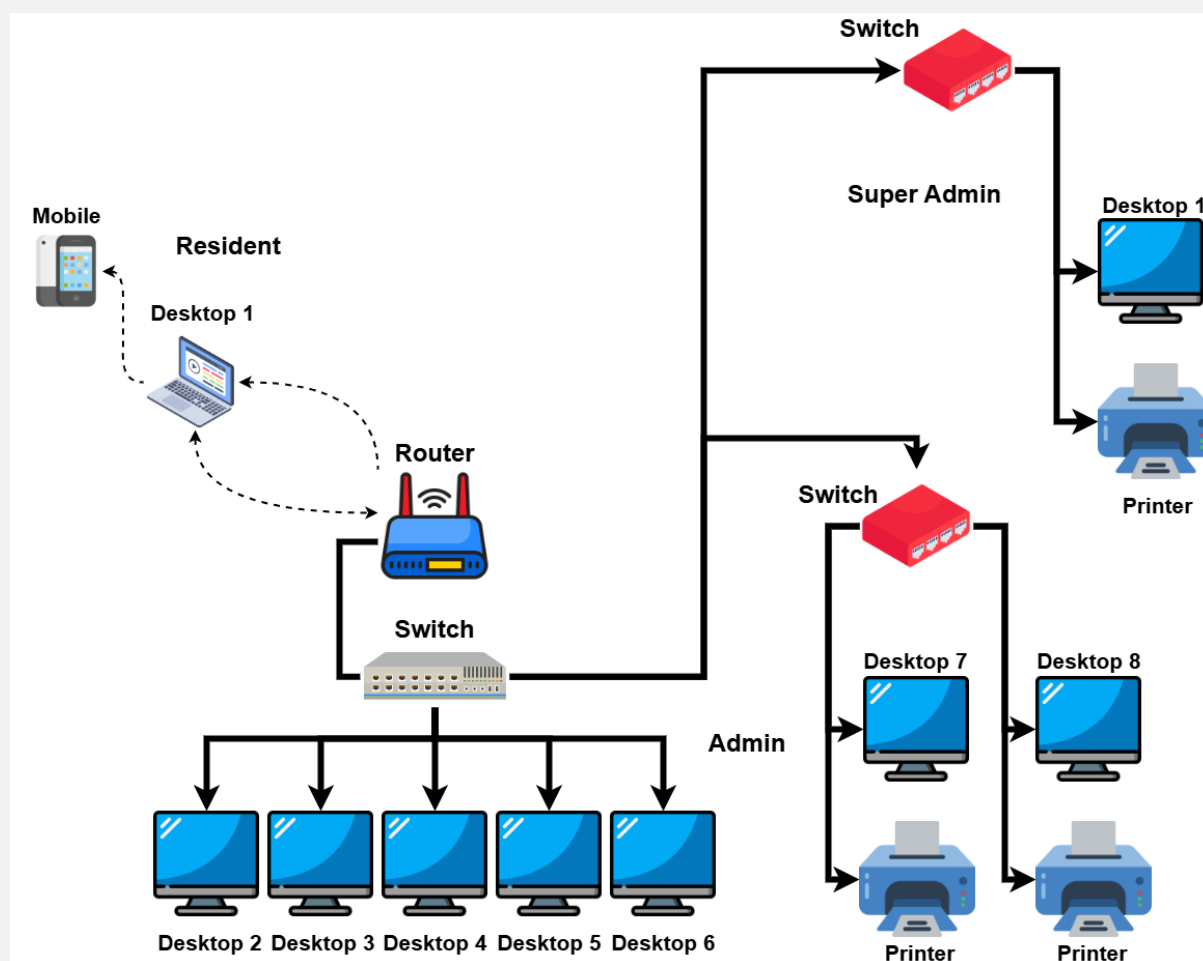


Figure 3.5: Hybrid network

Figure 3.6 above depicts a tiered hybrid network architecture where critical management tasks are segregated into specific administrative zones ("Super Admin" and "Admin") on the wired side, while general user access ("Resident") is handled on the wireless side. All activity converges at the central Switch and Router to manage traffic flow between these distinct user groups.

The figure illustrates a departmentalized Local Area Network (LAN) setup. Here is how its components would function in the context of an organization or school system, distinguishing between high-level administration and general users.

1. **Wired Connection (Main Switch Backbone)** - The central hub of the network is the large 24-port switch connecting the Router to various sub-networks. This serves as the primary distribution point for data across the facility.



2. General Workstations (Desktops 2-6) - These represent a general computer lab or a pool of staff computers connected directly to the main switch.
 - Staff/Lab Use: Users at these stations (Desktops 2 through 6) perform standard daily tasks, such as data entry or accessing the internal knowledge base. Because they are wired directly to the main switch, they share bandwidth equally.
3. Super Admin Network (Top Branch) - This is a dedicated sub-network for high-level management.
 - Secondary Switch: A separate switch filters traffic specifically for the Super Admin, isolating it from the general lab traffic for better security and performance.
 - Super Admin Station (Desktop 1): This workstation is likely used by the IT Director or Head Registrar. They have full access to all system configurations and sensitive records.
 - Dedicated Printer: The Super Admin has a private printer, ensuring that confidential reports or sensitive personnel files printed here are not mixed with general documents.
4. Admin Network (Side Branch) - This represents a specific department, such as Finance or Human Resources.
 - Departmental Switch: Like Super Admin, this group has its own switch to manage local traffic between their devices.
 - Department Staff (Desktops 7 & 8): Two administrative staff members work here, perhaps processing invoices or student records.
 - Dual Printers: This department is equipped with two dedicated printers, suggesting a high volume of paperwork (e.g., printing transcripts or receipts) that requires immediate, redundant printing capabilities independent of the main lab.
5. Wireless Connection (Router Tier) - The "Resident" section represents the wireless side of the network, managed directly by the Router.
6. The Router - The Router acts as the bridge between this internal network and the outside world, while also broadcasting the Wi-Fi signal.

- Traffic Management: It ensures that data requests from the wireless "Residents" do not collide with the heavy data transfer occurring in the wired "Admin" sections.

7. Resident Devices (Mobile & Laptop) - These represent transient or mobile users, such as students in dorms or guests.

- Resident Use: A user connects via their Mobile phone or Laptop (labeled Desktop 1 in the diagram) via Wi-Fi (indicated by dotted lines). They use this connection for flexible access, such as browsing the internet, checking emails, or viewing non-sensitive public portals, without needing physical access to the secure administrative offices.

3.4. Conceptual Design

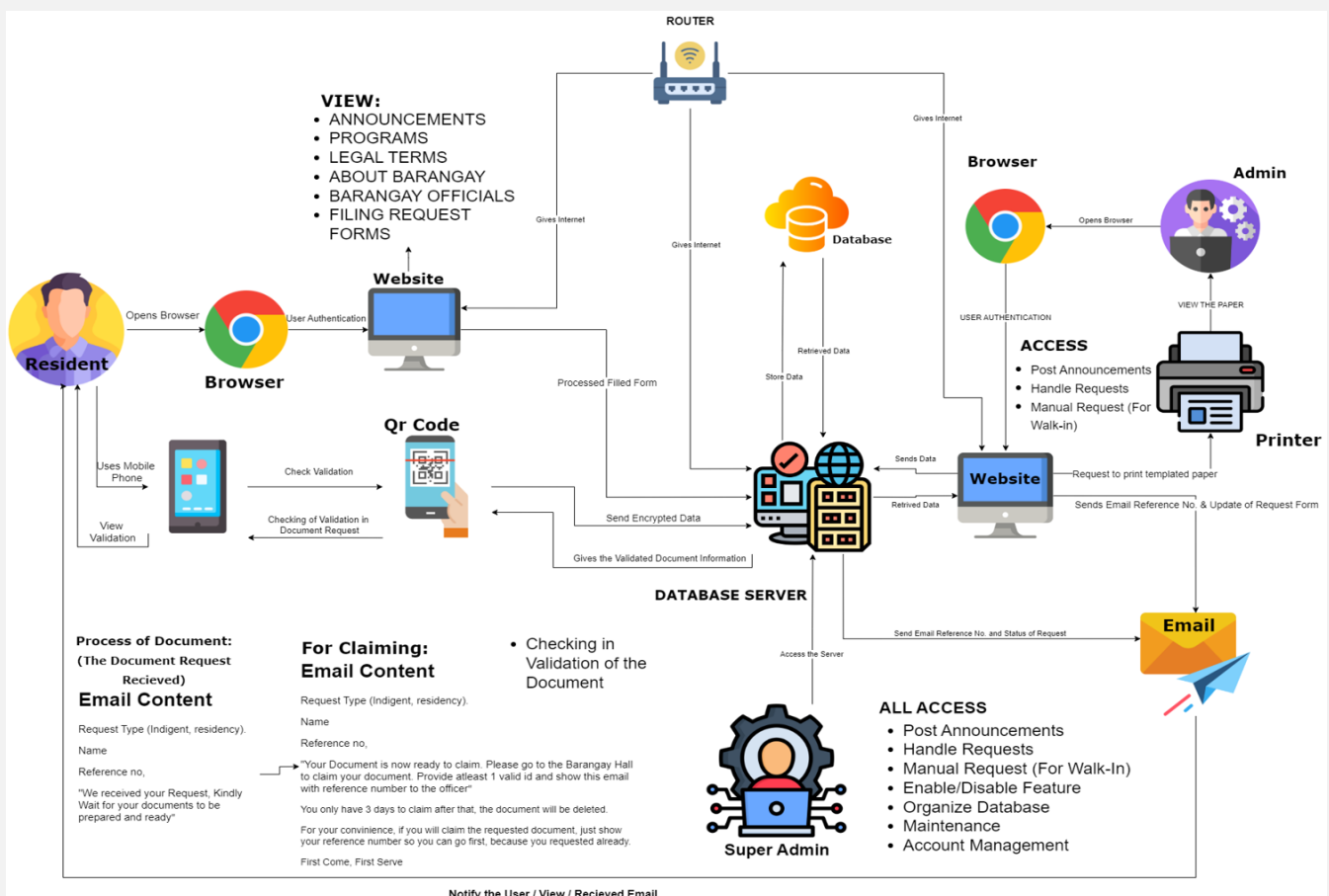


Figure 3.6: Proposed Conceptual Design



Figure 3.6 illustrates the conceptual framework of the Barangay Document Request and Verification System, emphasizing stakeholder collaboration for secure and efficient operations. The design connects Residents, who initiate document requests, with Barangay Administrators responsible for validation and issuance. System governance is maintained by Super Administrators via security configurations and audit logs, while External Verifiers authenticate document legitimacy through encrypted QR code verification.

The proposed system establishes a structured data flow to ensure efficient processing and transparent record management across three key stakeholder groups.

Resident Interaction Residents serve as the primary initiators, beginning with account registration and identity authentication. The system facilitates the submission of document requests, uploading of necessary requirements, and payment processing. In return, the platform provides residents with real-time transparency through automated notifications, allowing them to track request status from submission to approval and access their finalized documents for download or claiming.

Administrative Management & Oversight Barangay Administrators manage the operational workflow by validating submitted requests, verifying requirements, and confirming payment logs via detailed dashboards. Simultaneously, Super Administrators maintain system integrity by overseeing high-level configurations, including user roles, digital signatures, security settings, and audit trails. The system supports these functions by providing administrators with organized data summaries of pending, approved, and completed transactions.

External Verification To ensure document legitimacy, the system extends functionality to external parties such as employers and institutions. By scanning the encrypted QR code on issued documents, verifiers gain access to a secure, read-only record of authenticity, ensuring that all documents remain tamper-proof and traceable.

3.5. System Prototype

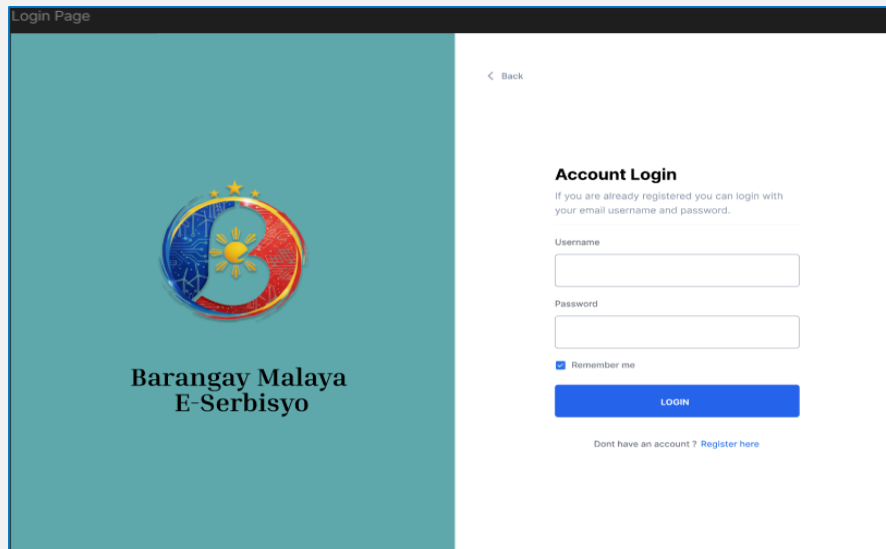


Figure 3.7: Login Form

This is the entry point for all users (Residents, Admins, Super Admins). It contains simple fields for authentication (Username/Email and Password). Its primary purpose is to verify user identity and direct them to their appropriate dashboard based on their role, ensuring data security and system access control.

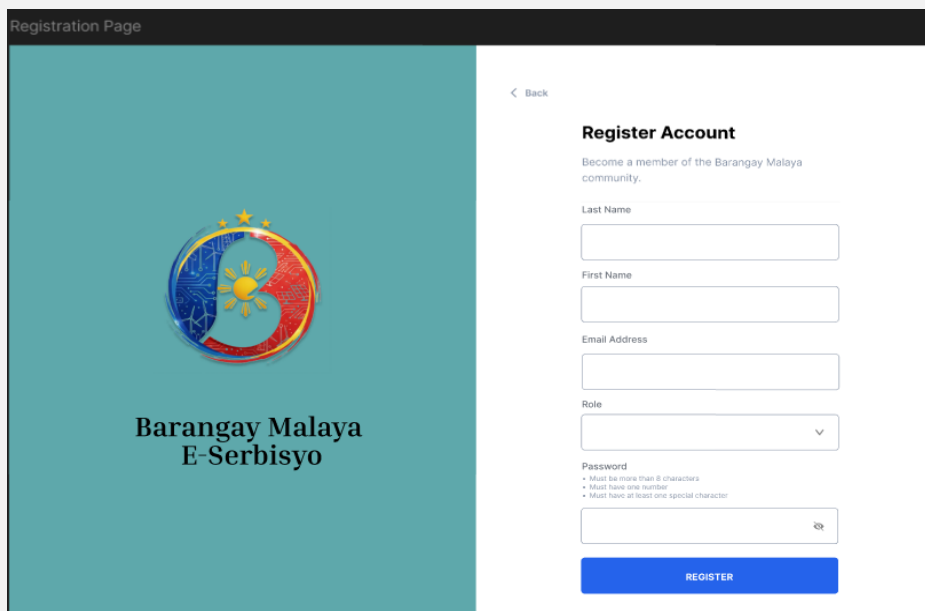


Figure 3.8: Registration Form

This form is used by new residents to create an account and gain access to digital services. It requires collecting essential personal details, a valid email, and a new password. Its function

is to securely onboard new users into the system and record their information in the resident database.

The figure displays two sequential screenshots of a 'Request Document' modal form. The left screenshot, labeled 'Request Document Dialog 3', shows the initial form with fields for Document Type, Purpose, Additional Notes (Optional), Request for Me/Request for Others, and Available Payment Method (Walk-in). The right screenshot, labeled 'Request Document Dialog 4', shows the same form with additional fields for First Name, Last Name, and Email, and a 'SUBMIT REQUEST' button at the bottom.

Figure 3.9: Request Document Forms

This is a modal form that appears over the main screen to allow a resident to initiate a specific service request. It guides the user through selecting the document type and providing required details or attachments. Its purpose is to isolate the requesting task into a focused, sequential workflow.



Add New Resident

First Name
Select Document

Last Name
Select Document

Username
Select Document

Contact Number
Select Document

Password
auto-generated (temporary)

Cancel Save

Add New Admin

First Name
Select Document

Last Name
Select Document

Username
Select Document

Contact Number
Select Document

Birthday
Select Document

Password
auto-generated (temporary)

Cancel Save

Figure 3.10: Add New Admin / Resident Form

This is a critical modal form used by the Super Admin to provision a new staff account. It collects essential staff details and assigns the correct role and initial permissions. Its purpose is to ensure all new administrative accounts are created securely and with the appropriate access levels.

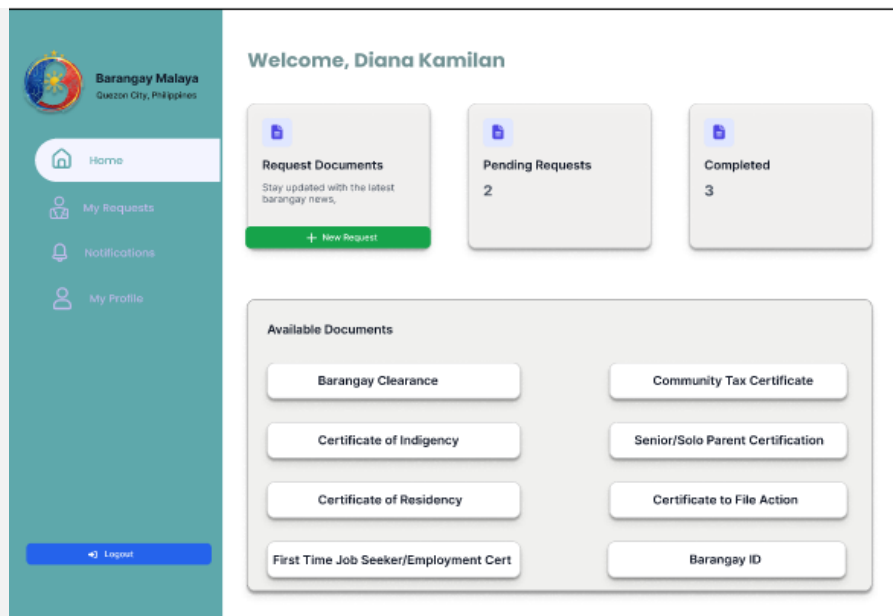


Figure 3.11: Resident Dashboard

This serves as the main entry dashboard for all authenticated residents. It features key operational shortcuts and quick links to manage requests and check profile status. The goal is to provide residents with an easy, centralized hub for accessing common Barangay services and viewing their recent activity.

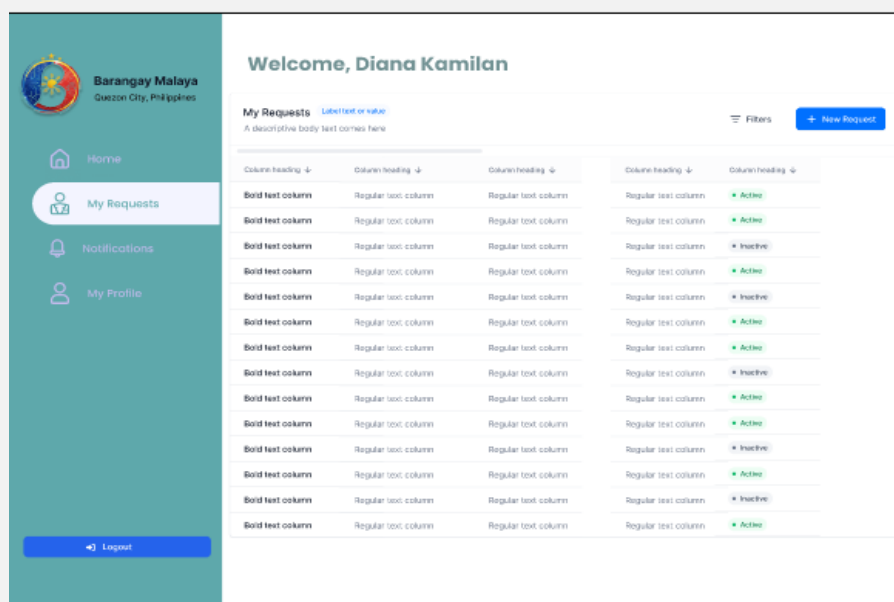


Figure 3.12: Resident Requests Page

This page displays a comprehensive history of all documents or services requested by the resident. Users can track the real-time status of each request (e.g., Pending,

Verified, Ready for Pickup). The purpose is to maintain transparency and provide a clear, chronological history of a resident's interactions with the system.

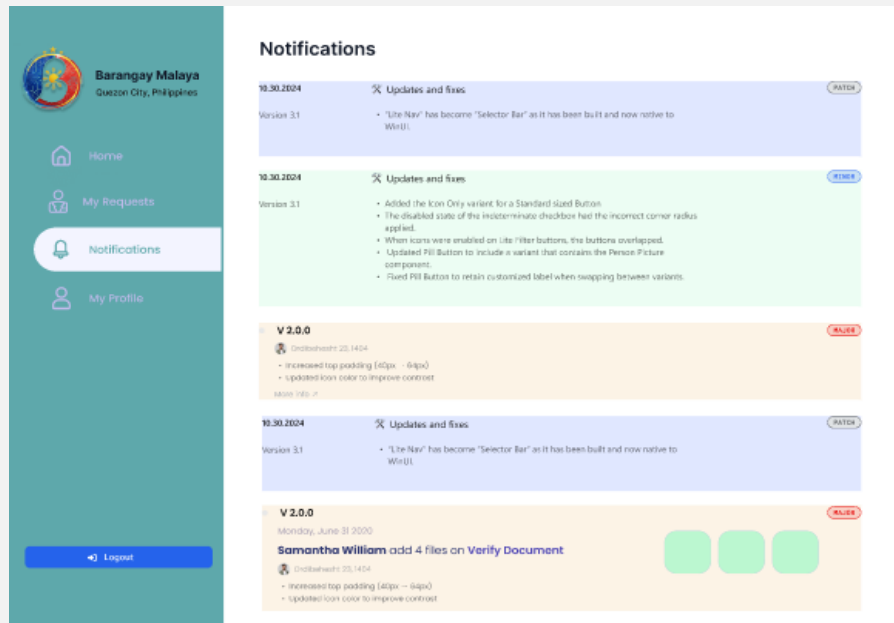


Figure 3.13: Resident's Notifications Page

This page aggregates all system alerts, updates, and direct messages relevant to the resident's account or requests. It is crucial for informing the user about changes in request status and official announcements. The goal is to ensure timely, non-intrusive communication between the system and the resident user.

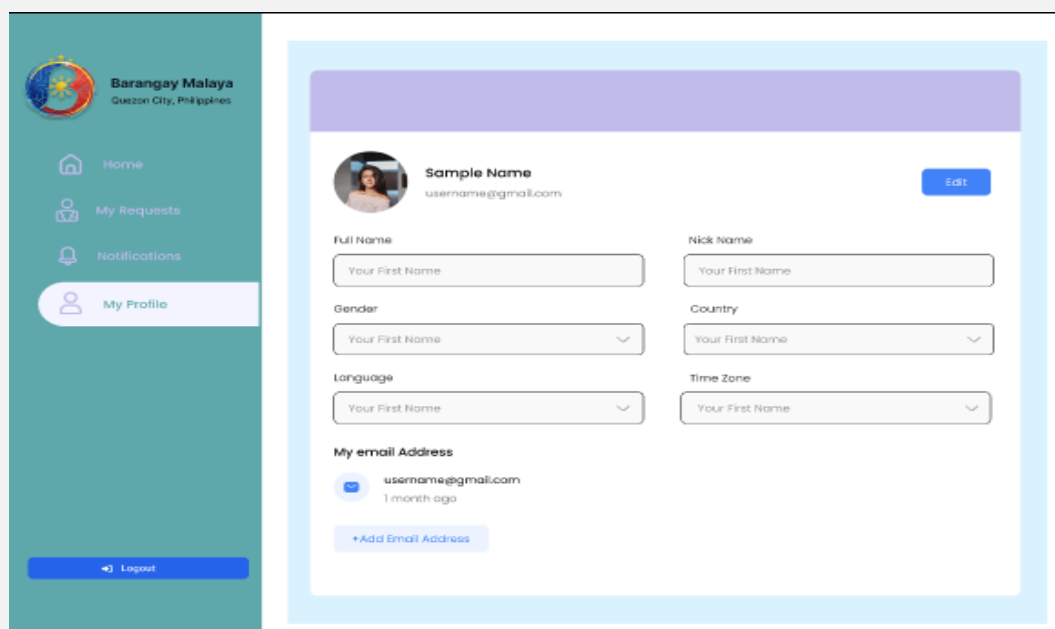


Figure 3.14: Resident's Profile Page

This area allows the resident to view and update their personal information, contact details, and account security settings. It ensures the Barangay maintains accurate and current resident records. Its purpose is to give the resident control over their data and enable self-service profile management.

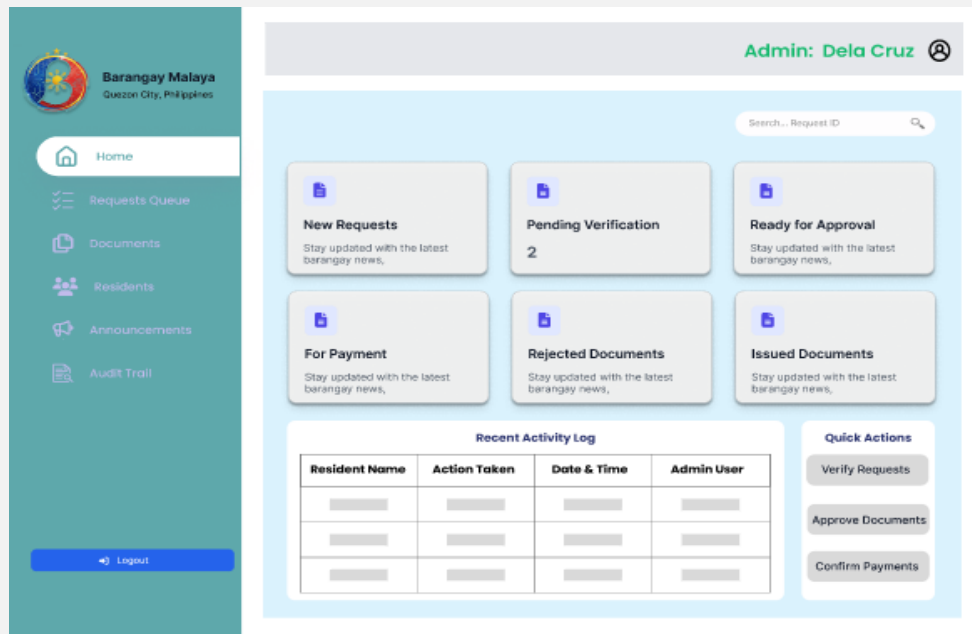


Figure 3.15: Admin Dashboard

This is the main operational hub for the Barangay staff. It features key performance indicators (KPIs) and widgets for current operational load, such as the total number of pending requests and recent documents issued. The primary purpose is to provide a real-time overview of daily tasks and guide the Admin's focus to critical areas

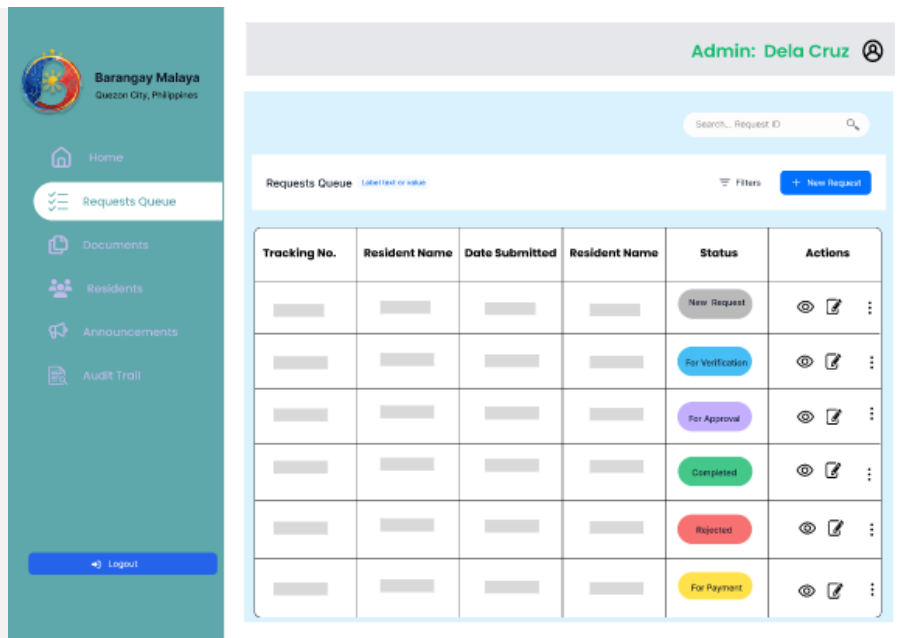


Figure 3.16: Admin Request Queue Page

This page lists all resident document requests requiring staff action. It allows the Admin to filter, sort, and process requests by managing statuses like Verification, Approval, and Payment. Its purpose is to streamline the document issuance workflow and reduce manual processing time.

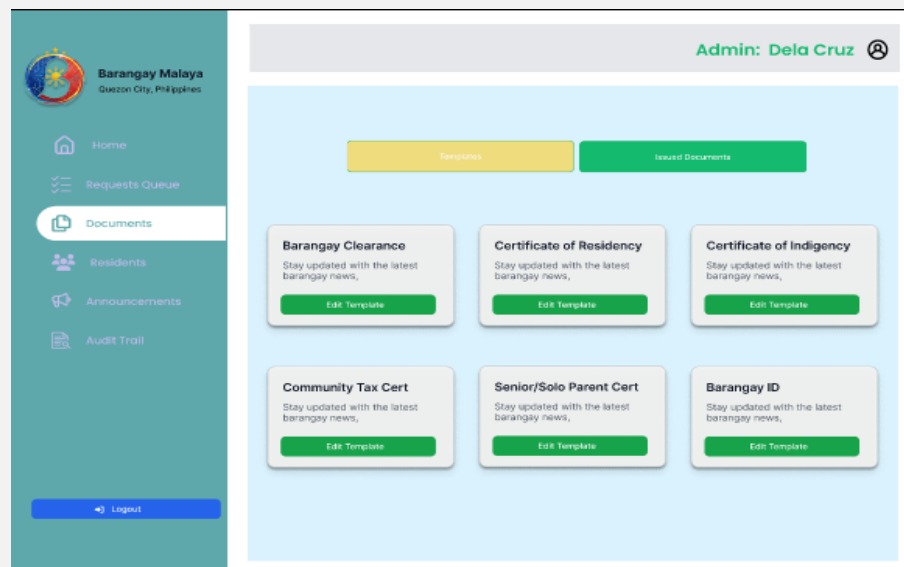


Figure 3.17: Admin Documents Page

This page manages the system's document types, templates, and issuance history. Admins can issue, reprint, or revoke documents, and manage categories and fees

associated with services. This ensures that all official documents are standardized, digitally tracked, and securely managed

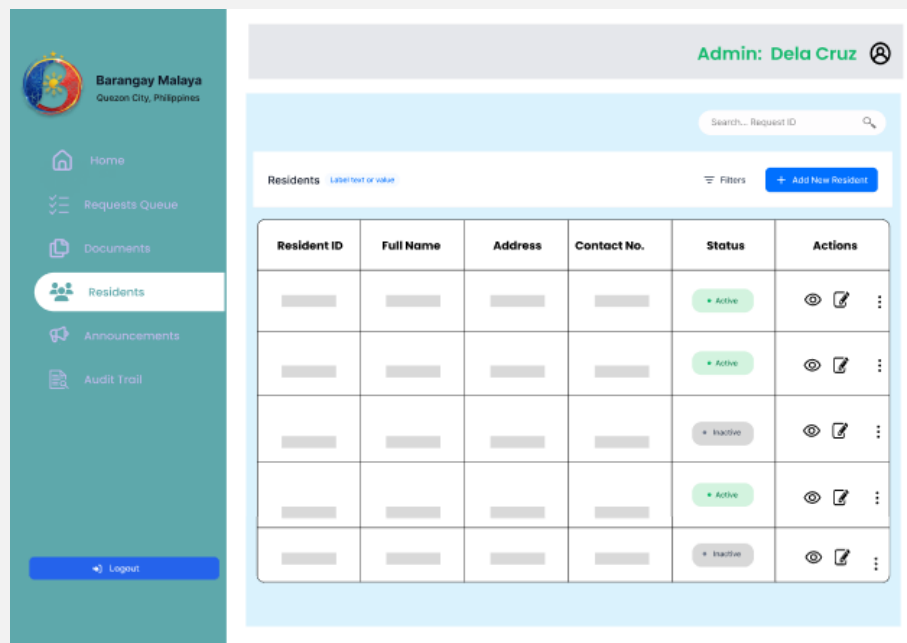


Figure 3.18: Admin's Residents Page

This page provides a comprehensive database of all registered Barangay residents. Admins can search, view, edit, and archive resident accounts, maintaining the integrity and accuracy of the community data. The goal is to give staff full control over resident accounts and verify resident details quickly.

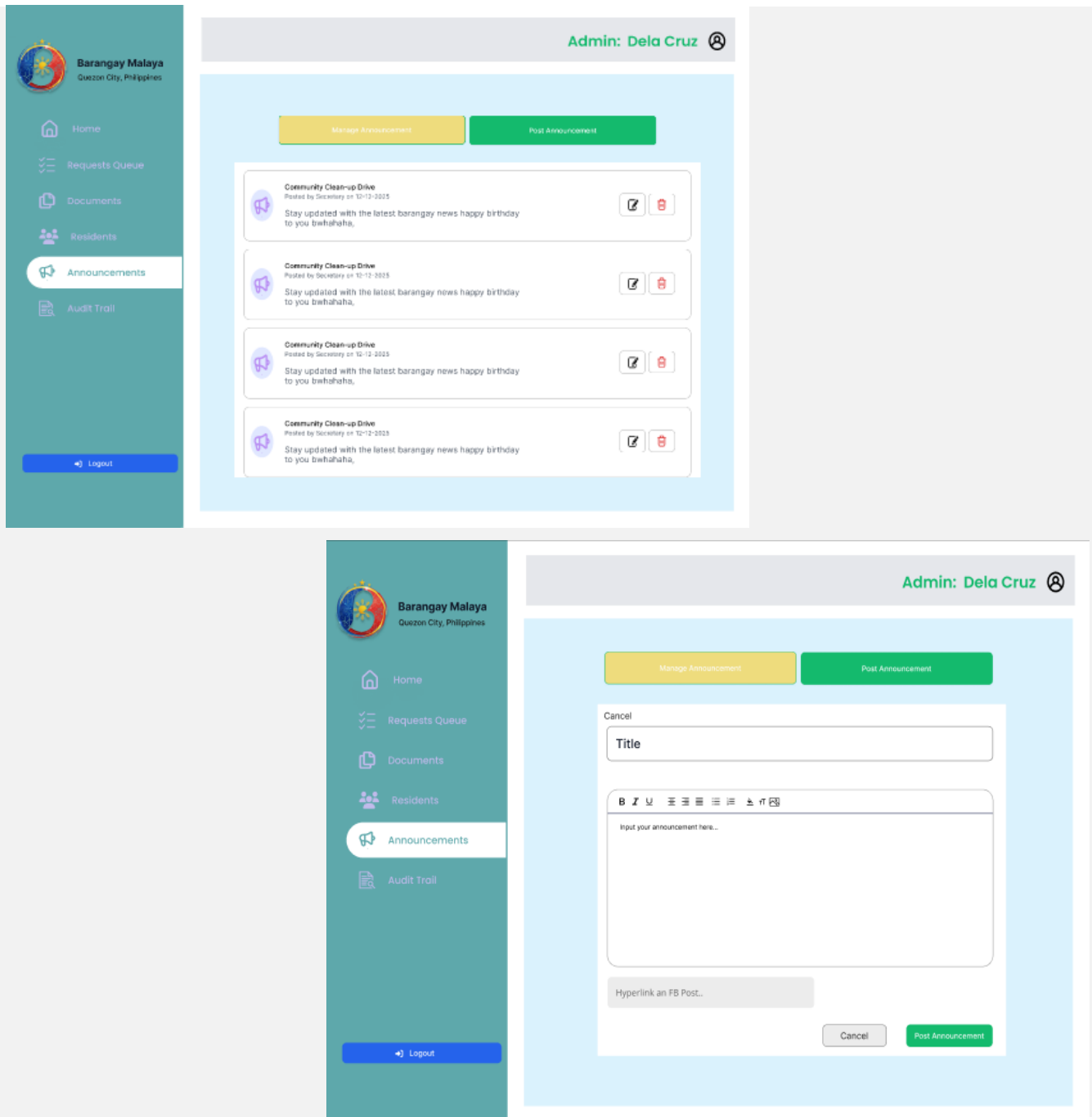


Figure 3.19: Admin's Announcements Page

This page is the management tool for Admins to create, edit, schedule, and publish announcements to the public-facing Homepage and the Resident Notifications page. It includes rich text editing features and controls for setting visibility and publish dates. Its purpose is to facilitate efficient and controlled dissemination of public information.

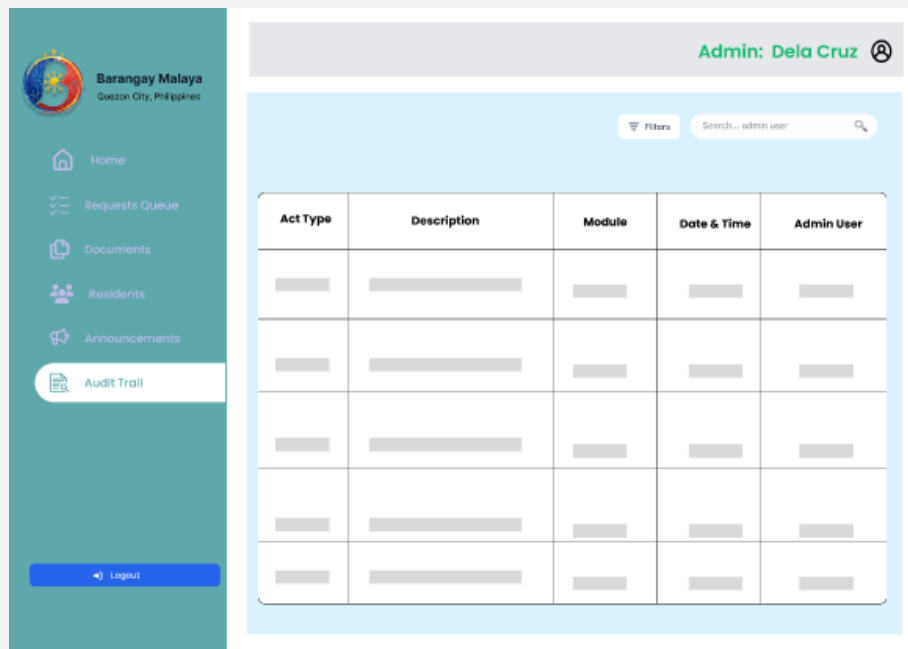


Figure 3.20: Admin's Audit Trail Page

This page is a complete, immutable history of all major user actions within the system, intended for security and accountability. It allows Admins (and Super Admins) to track when a request was processed, which user took the action, and the outcome. The log is essential for meeting compliance and transparency standards.

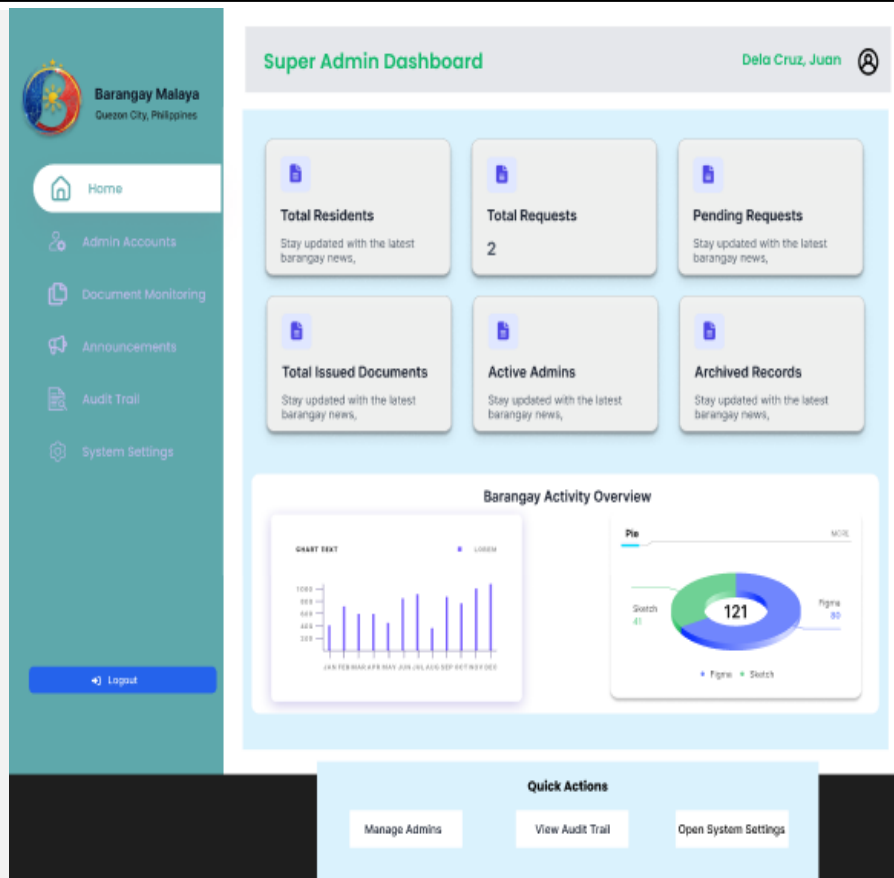


Figure 3.21: Super Admin Dashboard

This is the highest-level entry point, functioning as the system health and administrative oversight dashboard for the Super Admin. It tracks global metrics such as server performance, overall user activity trends, and system resource usage. Its purpose is to provide the system administrator with the necessary tools to monitor and maintain platform stability and performance.

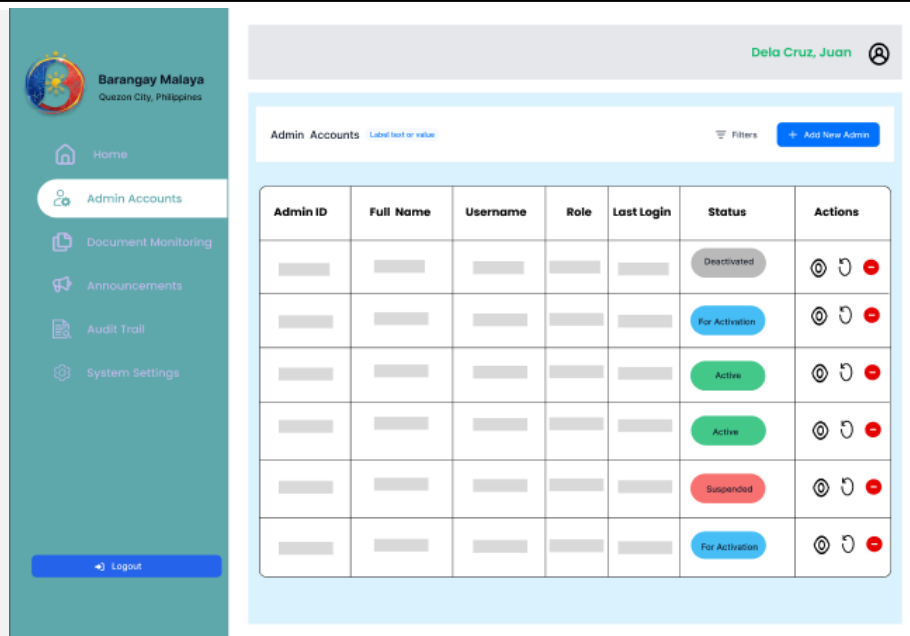


Figure 3.22: Super Admin Account Management Page

This page is exclusively for the Super Admin to manage the entire staff structure, including all Admin and Super Admin accounts. Functionality includes creating new accounts, assigning roles, resetting passwords, and deactivating staff access. This is essential for secure user lifecycle management and access control across the platform.

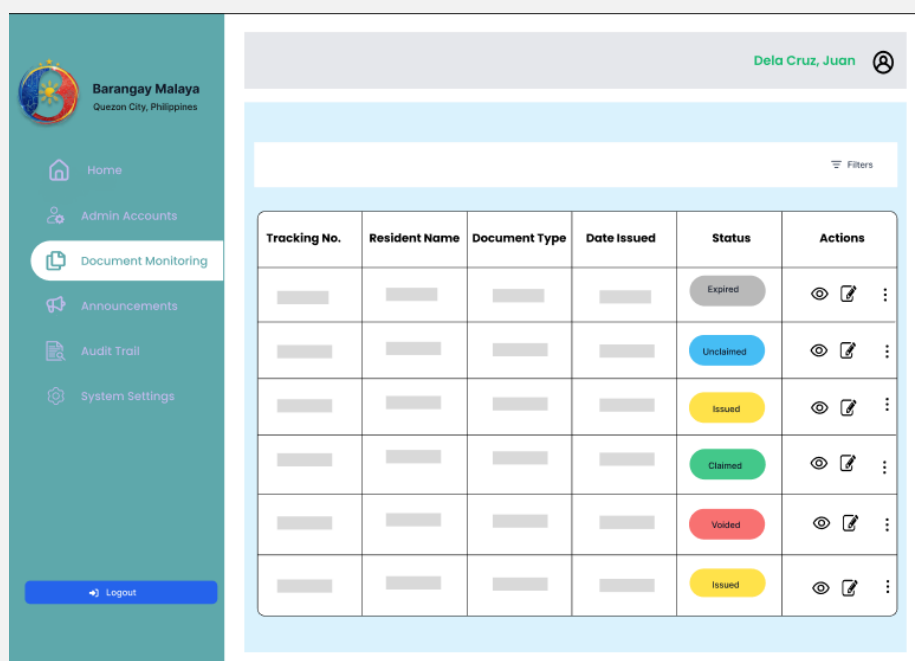


Figure 3.23: Super Admin Document Monitoring Page

This oversight page allows the Super Admin to track the flow of all documents and requests across the system. It provides high-level statistics on processing times,

issuance volume, and bottlenecks in the queue. The purpose is to optimize workflows and identify areas for efficiency improvement within the document management system.

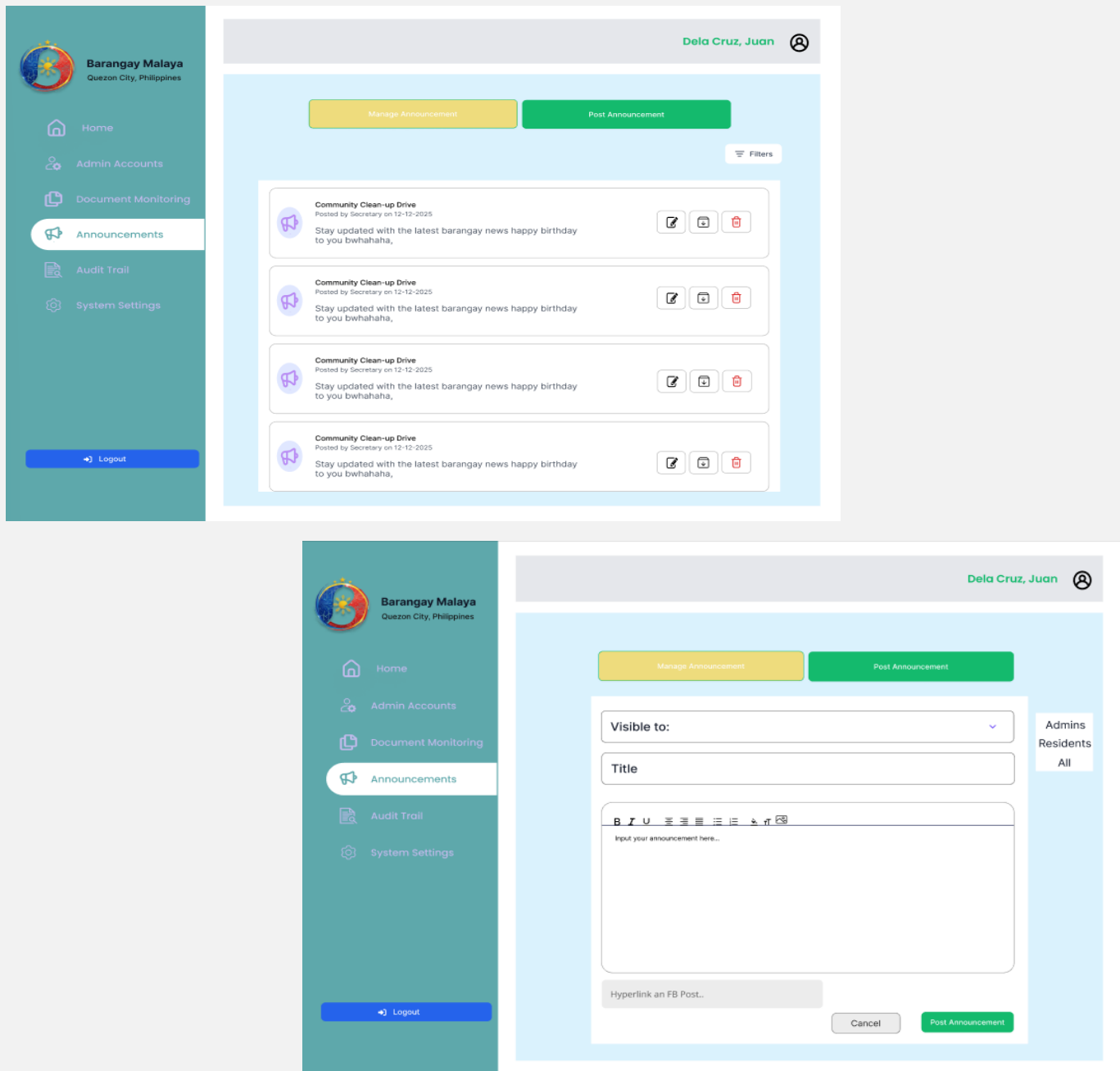


Figure 3.24: Super Admin's Announcement Page

This page is the management tool for Admins (and potentially Super Admins) to create, edit, schedule, and publish announcements to the public-facing Homepage and the Resident Notifications page. It includes rich text editing features and controls

for setting visibility and publish dates. Its purpose is to facilitate efficient and controlled dissemination of public information.

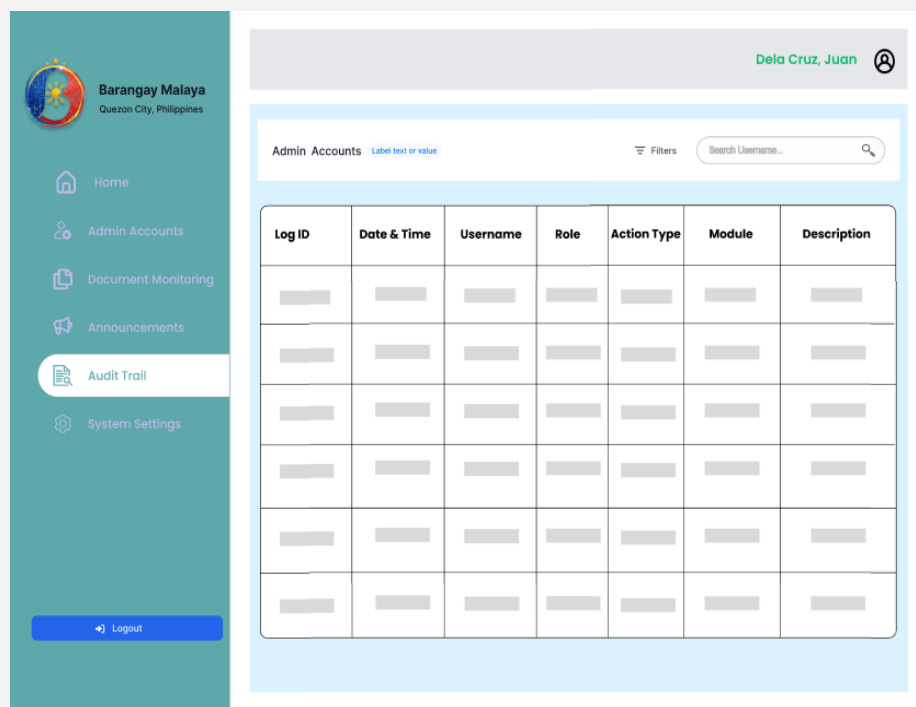


Figure 3.25: Super Admin's Audit Trail Page

This is the Super Admin's version of the Audit Log, featuring greater control, such as the ability to export the full log data and perform archival or purging functions. It maintains the highest level of accountability by securely recording every administrative action. The goal is to provide a forensic tool for system security analysis


Figure 3.26: Super Admin's System Settings Page

This configuration page gives the Super Admin control over global system variables and feature toggles. Settings include enabling/disabling resident registration, integrating payment options, and setting default operational values. This is where the core rules of the platform are maintained and updated.



EULOGIO "AMANG" RODRIGUEZ INSTITUTE OF SCIENCE AND TECHNOLOGY

INFORMATION AND TECHNOLOGY DEPARTMENT


**Barangay Malaya**
Quezon City, Philippines

HomeAbout UsServicesAnnouncementsContact UsVerify DocumentLogin

Welcome to Barangay Malaya


Your trusted community partner for efficient governance, quality services, and sustainable development. Building a better tomorrow, together.


[Request Document](#)[Learn More](#)





Barangay Malaya Officials

Meet our dedicated leaders serving the community

**Juan dela Cruz**
Barangay Captain

**Maria Santos**
Barangay Secretary


**Roberto Garcia**
Barangay Treasurer

**Ana Reyes**
Barangay Kagawad

Services

Our Services


Comprehensive barangay services at your fingertips



Barangay Clearance

Apply for and download your barangay clearance online. Fast, secure, and convenient.


[Learn More](#)



Residency Certificate

Get your certificate of residency for various legal and business requirements.


[Learn More](#)



Community Tax Certificate

Report incidents and concerns directly to barangay officials for immediate action.


[Learn More](#)



Issuance of Barangay ID

Schedule appointments with barangay officials for consultations and services.


[Learn More](#)



Certificate to File Actions

Access health programs, vaccination schedules, and medical assistance.

[Learn More](#)



Barangay Records


Stay updated with the latest barangay news, events, and important announcements.


[Learn More](#)


About Us


About Barangay Malaya

Barangay San Miguel has been serving our community for over 50 years, committed to providing excellent public services and fostering community development. Our mission is to create a safe, prosperous, and sustainable community for all residents.

**Transparent Governance**
Open and accountable leadership for community trust.

**Community-Centered Services**
Programs designed to meet residents' needs and aspirations.


**Digital Innovation**
Modern technology for efficient service delivery.





Contact Us

Get in Touch

Have questions or need assistance? We're here to help you.

 123 Barangay Road, Quezon City


 (02) 123-4567

 info@barangaymalaya.gov.ph

Send us a Message

[Send Message](#)

Verify Document



Verify Documents Here

Document Type

Select Document

[Upload Document Here](#)

[Scan QR Here](#)

[Verify Document](#)

Your uploaded "Certificate of Residency" is 100% Authentic

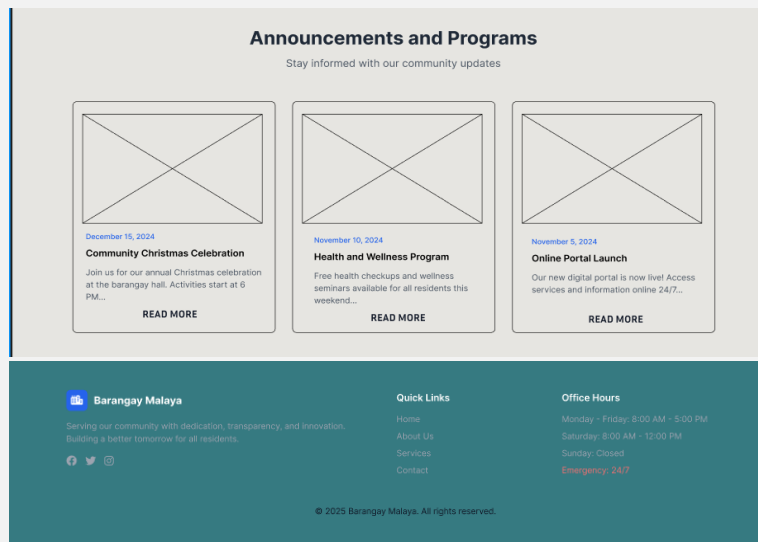


Figure 3.27: Landing Page / Home Page

This is the public-facing main website for the Barangay, accessible without login. It contains general information, service offerings, news, and official announcements. Its primary purpose is to establish the Barangay's digital presence and guide visitors to the system's authentication and document verification services.

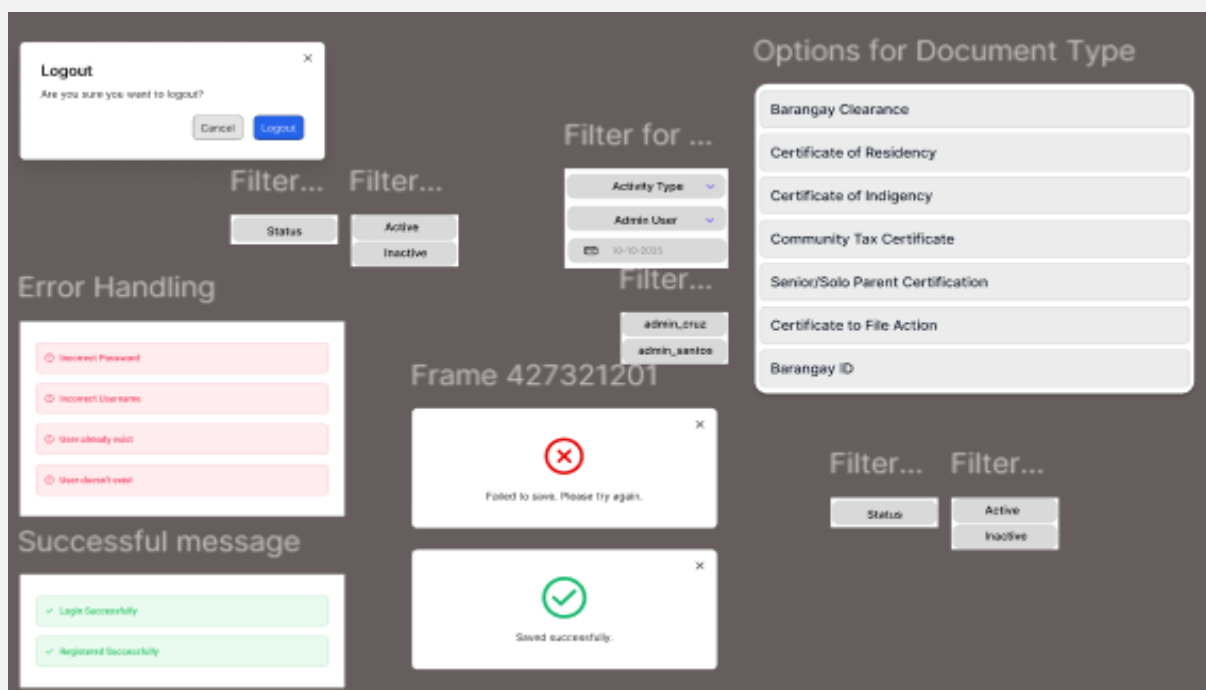




Figure 3.28: Dialogs/Popups/Modals

This category represents all non-page, context-specific overlay user interface elements, including modals, pop-ups, and non-blocking toasts. These components serve to isolate and focus the user on a single, critical task, such as confirming a

destructive action or inputting data into a short form. Their purpose is threefold: to prevent navigation interruptions during a workflow, to enforce critical security confirmations (like rejection or deactivation), and to provide immediate, non-blocking feedback (success or error) regarding an executed action. These dialogs are essential for maintaining data integrity, guiding users through multi-step processes, and ensuring a streamlined, focused user experience.

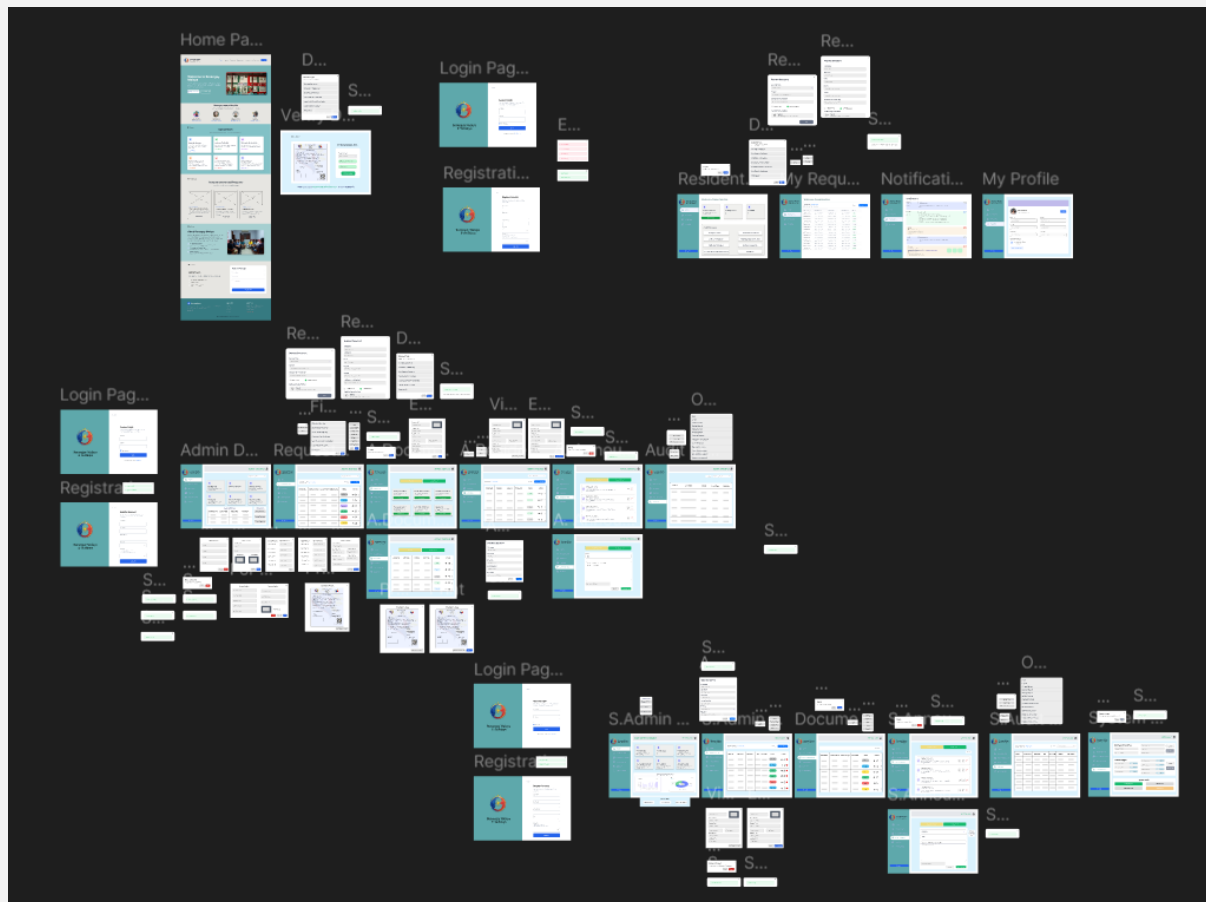


Figure 3.29: Whole Figma Canvas