**A PROPOSED OFFERING OF ENHANCING BARANGGAY SERVICES IN NBBS PROPER: THE DEVELOPMENT OF A REQUEST SYSTEM**

**VALENZUELA BRANCH**

A Project Proposal Presented to the

Faculty of Datamex College of Saint Adeline, Inc.

In Partial Fulfillment of the Requirements for the

Degree of Bachelor of Science in Information Technology

By:

Tingson, Emerald Maja

Pore, Kim

Espinosa, Angelica

Cañada Jessie Ver

August 2025

**REQUIREMENT SPECIFICATION**

**ADVISER’S CERTIFICATE**

This is to certify that the thesis entitled **“Web-based Enhancing Baranggay Services in NBBS Proper: The Development of a Request System”** prepared and submitted by Tingson, Emerald Maja, Pore, Kim, Espinosa, Angelica, Cañada Jessie Ver in partial fulfillment of the requirements for the degree of Bachelor of Science in Information Technology has been examined and is hereby recommended for approval and acceptance.

Prepared under my supervision.

Mr. Gabriel Thomas Torneros

Thesis Adviser

**ACKNOWLEDGMENTS**

First and foremost, I give all praise and glory to Almighty God, whose grace, wisdom, and strength have guided me throughout this journey. Without His divine guidance and blessings, this thesis would not have been possible.

I would like to express my sincere gratitude to my advisor, [Prof. Gabriel Thomas Toneros ], for their exceptional support, patience, and insightful advice throughout the course of my research. Their mentorship has played a crucial role in the completion of this work.

To my peers, colleagues, and friends at [Department or Institution Name], thank you for your camaraderie and for creating an inspiring academic environment. Your support and shared laughter made the tough days easier.

I am deeply grateful to my family for their unconditional love, prayers, and support. To my [parents/spouse/siblings], thank you for always believing in me and reminding me of my purpose when things got difficult.

This thesis is a reflection of all the people who have contributed to my growth, and I remain forever thankful.

**TABLE OF CONTENTS**

**TITLE PAGEi**

**REQUIREMENT SPECIFICATIONii**

**ADVISER’S CERTIFICATEiii**

**ACKNOWLEDGMENTiv**

**TABLE OF CONTENTSv**

**INTRODUCTION1**

**FUNCTIONAL REQUIREMENTS2-3**

**NON-FUNCTIONAL REQUIREMENTS4**

**USE CASES5**

**DATA REQUIREMENTS5**

**ASSUMPTION AND CONSTRAINTS6**

**INTRODUCTION**

Barangay NBBS Proper in Navotas City, like many barangays in the Philippines, plays an important role in the daily lives of its residents. It provides essential services such as issuing certificates, permits, and barangay clearances. However, the current way of delivering these services is mostly manual—residents have to go to the barangay hall, fill out paper forms, wait in line, and hope their documents are processed on time.

This traditional system often causes delays, lost paperwork, and long waiting times, which can be stressful for both residents and barangay staff. As the population grows and the demand for services increases, it becomes harder for the barangay to manage requests efficiently using manual processes.

In today's digital age, technology can offer a solution to these challenges. This study proposes the development of an online request system for Barangay NBBS Proper. The goal is to make barangay services more accessible, organized, and efficient by allowing residents to submit requests online, track their status in real time, and reduce the need for physical visits.

This system aims to:

Shorten waiting times, Reduce errors and misplaced documents, Help barangay staff process requests more efficiently, and Improve overall transparency and satisfaction among residents.

Ultimately, this project supports the idea that even small local governments can benefit from digital transformation. By improving how services are delivered, the barangay can better serve its people and build a more responsive and organized community.

### **FUNCTIONAL REQUIREMENTS**

These are the tasks the system needs to do.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID** | **Requirement Description** | **Priority** | **Dependencies** | **Acceptance Criteria** |
| FR-001 | Staff can add resident profiles in the system. | High | None | Staff can save resident details like name, address, and contact. |
| FR-002 | Staff can record and process document requests (like Barangay Clearance, Certificate). | High | FR-001 | Staff can select a resident and type the request in the system.. |
| FR-003 | Staff can check and update the status of a request. | High | FR-002 | Staff can mark a request as pending, approved, or released. |
| FR-004 | The system can make and print requested documents. | High | FR-002 | Staff can print an official copy of the requested document. |
| FR-005 | The system has a search option for resident records and requests. | Medium | FR-001, FR-002 | Staff can find a resident or request by name, ID, or request number. |
| FR-006 | The system can make summary reports (daily, weekly, or monthly).. | Medium | FR-003 | Staff can print or view reports for monitoring and record-keeping.. |

### **NON-FUNCTIONAL REQUIREMENTS**

**Performance**

* **The system should respond to user actions (page load, search, submit, etc.).**
* **During peak hours, it should maintain acceptable response times without significant degradation.**
* **It should include monitoring tools to measure response time, throughput, and resource usage.**
* **Performance testing (e.g., load and stress tests) should be part of deployment to verify speed and stability.**

**Usability**

* **The interface should follow a consistent layout (fonts, colors, button placement).**
* **Use accessible design: readable fonts, proper color contrast, keyboard navigation.**
* **Support two devices Laptop or Computer**

**Reliability**

* **There should be automatic backups of important data (e.g., weekly or monthly).**
* **The system should have error handling so users receive clear messages instead of crashes.**
* **Failover or recovery mechanisms should be in place (e.g., redundant servers).**

**Security**

* **All communication between client and server must use HTTPS encryption.**
* **User accounts should require strong passwords and support multi-factor authentication if needed.**
* **Apply role-based access control so staff and residents only see what they’re authorized to.**
* **Store sensitive data (e.g., personal details, passwords) with proper encryption and hashing.**
* **Regularly update and patch software to reduce vulnerabilities.**
* **Keep detailed audit logs of logins and changes.**

**Scalability**

* **The architecture should allow horizontal or vertical scaling (e.g., adding servers or upgrading hardware).**
* **Use a modular design so new features can be added without major changes to the whole system.**
* **Plan for growth specify the maximum number of users, data records, or transactions the system.**

**Maintainability**

* **Source code should follow clean coding standards and be well-commented.**
* **Maintain clear documentation for developers and support staff (setup, deployment, troubleshooting).**
* **The system should have an easy way to apply patches, configuration changes, or feature updates without long downtime.**
* **Automated testing and continuous integration can help catch bugs early.**

### **USE CASES**

Examples of how the system will be used by barangay staff (Online).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID** | **Use Case Name** | **Description** | **Actors** | **Preconditions** | **Alternate Flows** |
| UC-001 | Add Resident | The staff will add a new resident to the system. | Barangay Staff, System | The system is open and ready. | The resident info is not complete, so it cannot be saved. |
| UC-002 | Record Request | The staff will enter a request for a document. | Barangay Staff, System | The resident profile is already in the system. | The request cannot be saved if details are missing. |
| UC-003 | Process the Request | The staff will check the request and mark it as pending, approved, or released. | Barangay Staff, System | There is a request recorded in the system.. | The staff cannot approve if needed papers are missing. |

### **DATA REQUIREMENTS**

This is the information that needs to be saved in the system.

**Admin Account Information**

**Admin ID (unique): A special code that identifies each admin.**

**Full name: The admin’s complete name.**

**Email address: The admin’s email for messages or login.**

**Username & password (securely stored): A safe way for admins to sign in.**

**Role/permissions level: What tasks the admin is allowed to do.**

**Contact number: The admin’s phone number.**

**Resident Account Information**

**Resident ID (unique): A code that identifies each resident.**

**Full name: The resident’s full name.**

**Address/unit number: Where the resident lives.**

**Email and/or phone number: Ways to contact the resident.**

**Username & password (hashed): A secure login for residents.**

**Account status (active, pending, suspended): Shows if the account is ready, waiting, or blocked.**

**Authentication & Access**

**Tools and info that confirm who a user is and keep them signed in.**

**Login credentials: Username and password.**

**Session tokens/cookies: Keep users logged in as they browse.**

**Password recovery data: Helps reset a forgotten password.**

**Audit logs: Records of who logged in, out, or made changes.**

**System & Operational Data**

**Information that keeps the system running well.**

**Records of updates (audit trail): History of admin edits.**

**Backup and recovery logs: Notes about saved data and restores.**

**Error or crash logs: Reports when something goes wrong.**

**System performance metrics: How fast and reliable the system is.**

**Content / Functional Data**

**The visible parts and settings of the system.**

**Home page content: Banners, notices, and quick links.**

**Menus/navigation: Buttons and lists to move around.**

**Configuration settings: Options for how features appear.**

**Security Data**

**Information that protects users and data.**

**Encryption keys: Secret codes that lock data safely.**

**Access control lists: Rules about who can see or use things.**

**Multi-factor authentication records: Details for extra login steps (like codes or apps).**

**Assumptions and Constraints**

**Assumptions:**

* Residents will have access to the internet so they can use the online request system.
* All document and service requests will be submitted and processed only through the system (no more paper-based forms).
* Barangay staff will have computers/laptops and basic training to operate the system.
* The internet connection in the barangay office will be stable enough for daily operations.
* Data provided by residents (like names and addresses) will be accurate and up to date.
* System users (residents and staff) will keep their login details private and use the system responsibly.

**Constraints:**

* Time Constraint: The project must be finished within the school/academic deadline (limited time frame).
* Budget Constraint: There is only a small budget for design, development, and testing of the system.
* Technology Constraint: The system is intended to run only on laptops or desktop computers (no mobile app for now).
* Scope Constraint: The project will focus only on processing requests (e.g., barangay clearance, certificates) — not on other barangay activities.
* Resource Constraint: The team is composed of students with limited manpower, tools, and technical resources.
* Data Security Constraint: Personal data of residents must be stored and handled according to security rules (e.g., encryption, limited access).

**Glossary**

|  |  |
| --- | --- |
| Term | Meaning (Simple Definition ) |
| Barangay | The smallest unit of local government in the Philippines; similar to a village or neighborhood. |
| NBBS Proper | A barangay (community) in Navotas City, Philippines where the project will be implemented. |
| Request | A resident’s application for a document or service (e.g., barangay clearance, certificate). |
| Request System | The online tool where residents can submit, track, and receive services or documents. |
| Resident | A person living in the barangay who uses the system to request services. |
| User | Anyone using the system — this can be a resident or barangay staff member. |
| Barangay Staff | Employees or officials in the barangay office who manage resident requests and data. |
| Admin Account | A staff account with full access to manage the system and its users. |
| Resident Account | A user account created for residents so they can log in and request services. |
| Admin ID / Resident ID | A unique number or code assigned to each admin or resident. |
| Audit Log | A record of actions (like logins, changes, or errors) that helps track system activity. |
| Authentication | The process of confirming a user’s identity before allowing access (e.g., login). |
| Authorization | The level of access or actions a user is allowed to do after logging in. |
| Backup | A saved copy of system data to prevent loss during errors or failures. |
| Encryption | A method of turning data into a secret code to protect it from unauthorized access. |
| Error Log / Crash Log | A file that keeps details of problems or failures in the system. |
| HTTPS | A secure version of HTTP that protects data sent between a browser and a server. |
| Multi-factor Authentication (MFA) | An extra step in logging in, like entering a code sent to a phone. |
| Scalability | The system’s ability to grow and handle more users, data, or transactions. |
| Maintainability | How easy it is to fix, update, or improve the system. |
| Performance | How fast and responsive the system is while handling tasks. |
| Usability | How simple and user-friendly the system is for residents and staff. |
| Reliability | The system’s ability to work well without crashing or losing data. |
| Functional Requirements | Tasks the system must be able to perform (e.g., add resident, print clearance). |
| Use Case | A short story that shows how a user interacts with the system to complete a task. |
| Throughput | The amount of work or number of requests the system can handle in a given time. |
| Patch | A small update or fix applied to correct a problem in the system. |
| Module | A separate part or feature of the system, like “Resident Records” or “Reports.” |
| Session Token / Cookie | Small pieces of data stored in a browser to keep a user logged in. |
| Hashing | A way of transforming passwords into unreadable text so they can’t be stolen. |
| Data Security | Protecting stored or transmitted data from unauthorized access. |
| Clearance / Certificate | Official documents issued by the barangay for various purposes. |
| Digital Transformation | Using technology to improve services that were previously done manually. |
| Role / Permission Level | The set of actions (like view, edit, or delete) allowed for a user type. |
| Scope | The specific work or limits of the project (e.g., only for requests, not for other services). |
| Constraint | A limit on time, budget, resources, or tools while developing the system. |
| Assumption | A condition believed to be true to make planning and development easier. |
| Revision History | A list of edits or changes made to the document over time. |

**Revision History**

|  |  |  |
| --- | --- | --- |
| **Date** | **Sections Update** | **Description Of Changes** |
| August 10, 2025 | Introduction, Client Information, Project Scope | Revised to ensure a consistent format across all sections; aligned the structure, style, and layout. |
| August 12, 2025 | Project Approach, Project Team, Project Timeline | Adjusted inconsistent fonts and figure/table formatting; simplified wording for clarity. |
| August 16, 2025 | Project Resources, Risk Management, Communication Plan | Reformatted sections, improved overly complex wording, and created tables for better readability. |
| August 22, 2025 | Project Governance, Approval, Appendix | Refined sentences for clarity and improved text formatting. |