**CEBU INSTITUTE OF TECHNOLOGY**

**UNIVERSITY**

COLLEGE OF COMPUTER STUDIES

Software Requirements Specifications

for

Barangay360

Prepared by:

Omen, Jared Karl A.

Castañeda, Achilles Niño V.

Navarro, Mikhail James P.

Change History

Table of Contents

Change History 2

Table of Contents 3

1. Introduction 4

1.1. Purpose 4

1.2. Scope 4

1.3. Definitions, Acronyms and Abbreviations 4

1.4. References 4

2. Overall Description 5

2.1. Product perspective 5

2.2. User characteristics 5

2.4. Constraints 5

2.5. Assumptions and dependencies 6

3. Specific Requirements 7

3.1. External interface requirements 7

3.1.1. Hardware interfaces 7

3.1.2. Software interfaces 7

3.1.3. Communications interfaces 7

3.2. Functional requirements 7

Module 1 7

Module 2 8

3.4 Non-functional requirements 8

Performance 8

Security 8

Reliability 8

# Introduction

## Purpose

The purpose of **Barangay360** is to empower barangays by creating a connected, informed, and engaged community through a digital platform that bridges the gap between residents and officials. By addressing challenges such as fragmented communication, inefficient service delivery, and low civic participation, Barangay360 aims to streamline processes like event management, document requests, and emergency alerts while promoting transparency and accountability. Ultimately, the platform seeks to strengthen community bonds, enhance the quality of life for residents, and foster a collaborative environment where barangays can thrive as modern, resilient, and inclusive local governance units.

## Scope

* *Identify the software product(s) to be produced by name (e.g., Host DBMS, Report Generator, etc.);*
* *Explain what the software product(s) will, and, if necessary, will not do;*
* *Describe the application of the software being specified, including relevant benefits, objectives, and goals;*
* *Be consistent with similar statements in higher-level specifications (e.g., the system requirements specification), if they exist.*

## Definitions, Acronyms and Abbreviations

* *provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS*

## References

* *Provide a complete list of all documents referenced elsewhere in the SRS;*
* *Identify each document by title, report number (if applicable), date, and publishing organization;*
* *Specify the sources from which the references can be obtained.*

# Overall Description

## Product perspective

**Barangay360** is a comprehensive digital platform designed to address the challenges of fragmented communication, low community engagement, and inefficient service delivery in barangays by providing a centralized hub for essential services and activities. It solves these issues by offering features such as event management for organizing vaccination drives and local seminars, real-time announcements for important updates, streamlined document requests for certificates and permits, and instant emergency alerts for natural disasters or health crises. Additionally, Barangay360 promotes transparency and accountability through resource management tools, a directory for barangay officials, and a feedback system that allows residents to report issues and participate in surveys. By bridging the gap between officials and residents, Barangay360 fosters a stronger sense of community, enhances civic engagement, and improves overall quality of life, ensuring barangays operate as modern, efficient, and transparent communities.

**Why Barangay360 Matters**

Barangay360 is more than just a platform—it’s a movement toward smarter, more inclusive communities. By leveraging technology to solve long-standing challenges, it ensures that every resident, regardless of location or background, has access to the information and services they need. Whether it’s staying informed about local events, accessing critical documents, or responding to emergencies, Barangay360 equips barangays with the tools to operate effectively and residents with the means to participate actively.

In a world where connectivity and efficiency are paramount, Barangay360 stands as a beacon of progress, redefining how barangays function and empowering communities to build a brighter future together.

## User characteristics

* *Describe all user types and their roles and privileges in the system*

## 2.4. Constraints

* *Provide a general description of any other items that will limit the developer’s options.*
* *Regulatory policies;*
* *Hardware limitations (e.g., signal timing requirements);*
* *Interfaces to other applications;*
* *Parallel operation;*
* *Audit functions;*
* *Control functions;*
* *Reliability requirements;*
* *Criticality of the application;*
* *Safety and security considerations.*

## 2.5. Assumptions and dependencies

*This subsection of the SRS should list each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software but are, rather, any changes to them that can affect the requirements in the SRS. For example, an assumption may be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not avail- able, the SRS would then have to change accordingly.*

# Specific Requirements

## External interface requirements

### 3.1.1. Hardware interfaces

*This should specify the logical characteristics of each interface between the software product and the hard- ware components of the system. This includes configuration characteristics (number of ports, instruction sets, etc.). It also covers such matters as what devices are to be supported, how they are to be supported, and protocols. For example, terminal support may specify full-screen support as opposed to line-by-line support.*

### 3.1.2. Software interfaces

*This should specify the use of other required software products (e.g., a data management system, an operating system, or a mathematical package), and interfaces with other application systems (e.g., the linkage between an accounts receivable system and a general ledger system).*

### 3.1.3. Communications interfaces

*This should specify the various interfaces to communications such as local network protocols, etc.*

## Functional requirements

### Module 1: Web Functionalities

#### 1.1 Dashboard for Admins or Officials Overview of announcements, requests, and statistics. Use Case Diagram

**Use Case Description:**

* **View Announcements:** This use case allows both Admins/Officials and Users to view community announcements. The system will display a list of announcements, which may include titles, summaries, and dates.
* **View Requests:** This use case enables Admins/Officials to view pending and processed requests submitted by users. The dashboard will provide an overview of these requests.
* **View Statistics:** As an extension of viewing requests, this use case allows Admins/Officials to access and view various statistics related to the requests, and potentially other platform activities. This helps in understanding trends and workloads.

**Actors:** Admin/Official, User

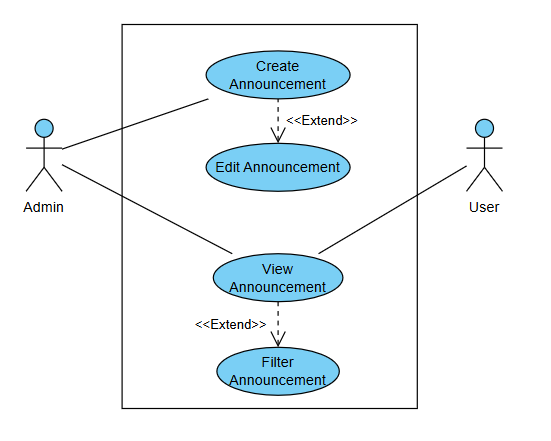
**Activity Diagram**

**A screenshot of a computer

AI-generated content may be incorrect.**

#### 1.2 Community Announcements with Search Filter

Create, edit, and filter announcements.



**Use Case Description:**

* **Create Announcement:** This use case allows Admins/Officials to compose and publish new community announcements. This involves inputting a title, content, and potentially other relevant details like dates or categories.
* **Edit Announcement:** As an extension of creating an announcement, this use case enables Admins/Officials to modify existing announcements. This could involve correcting typos, updating information, or changing the status of an announcement.
* **View Announcement:** This use case allows both Admins/Officials and Users to read community announcements. The system will display the content of selected announcements.
* **Filter Announcement:** As an extension of viewing announcements, this use case allows both Admins/Officials and Users to search for specific announcements based on criteria such as keywords, dates, or categories. This helps in quickly finding relevant information.

**Actors:** Admin/Official, User

**Activity Diagram**

A screenshot of a computer screen

AI-generated content may be incorrect.

#### 1.3 Request Approval System

Approve/reject requests for documents and permits.

##### 

##### Use Case Description

##### Activity Diagram

* Wireframe

**Use Case Description:**

* **Send Request:** This use case allows Users to submit requests for documents or permits. The user will fill out necessary forms and attach any required information.
* **Approve/Reject Request:** This use case enables Admins/Officials to review submitted requests. They can choose to either approve or reject a request based on predefined criteria and submitted information.
* **Send Documents:** This use case is included when an Admin/Official approves a request. It involves the system or the admin sending the requested documents or permits to the User, typically electronically.
* **Notify Rejection:** As an extension of the approve/reject process, if a request is rejected, this use case ensures the User is notified of the rejection, potentially along with the reason for rejection.

**Actors:** Admin/Official, User

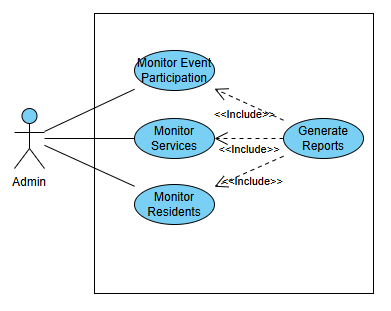
**Activity Diagram**

**A flowchart of a system

AI-generated content may be incorrect.**

#### 1.4 Data Analytics

Monitor residents, services, and event participation.



**Use Case Description:**

* **Monitor Residents:** This use case allows Admins/Officials to view and analyze data related to registered residents. This could include demographic information, trends in new registrations, or other relevant resident data points.
* **Monitor Services:** This use case enables Admins/Officials to track the usage and performance of various services offered through the platform, such as document requests or facility bookings.
* **Monitor Event Participation:** This use case allows Admins/Officials to oversee and analyze data regarding participation in community events, including attendance numbers and engagement metrics.
* **Generate Reports:** This use case is included within the monitoring functions (Residents, Services, Event Participation). It allows Admins/Officials to generate summarized reports from the monitored data, facilitating insights and decision-making.

**Actors:** Admin/Official

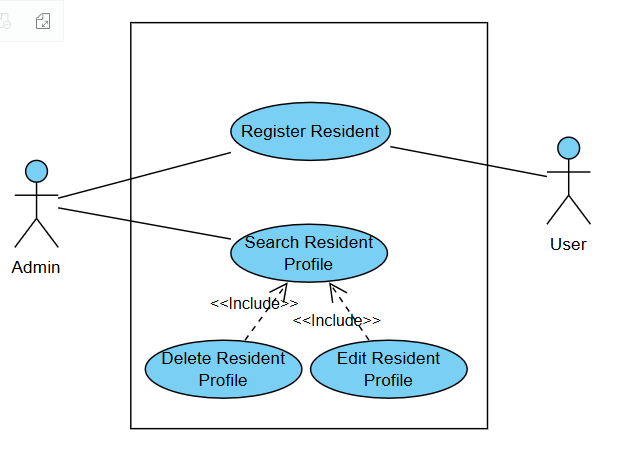
**Activity Diagram**

A screenshot of a diagram

AI-generated content may be incorrect.

#### 1.5 Resident Management

Register, edit, and search resident profiles.



**Use Case Description:**

* **Register Resident:** This use case allows both Admins/Officials and Users (presumably for self-registration or registration of household members) to create new resident profiles in the system. This involves inputting personal details and other required information.
* **Search Resident Profile:** This use case enables Admins/Officials to search for and retrieve specific resident profiles based on various criteria like name, address, or ID number.
* **Edit Resident Profile:** Included as part of managing a searched profile, this use case allows Admins/Officials to update or modify the information within an existing resident's profile.
* **Delete Resident Profile:** Included as part of managing a searched profile, this use case allows Admins/Officials to remove a resident's profile from the system, subject to data retention policies and permissions.

**Actors:** Admin/Official, User

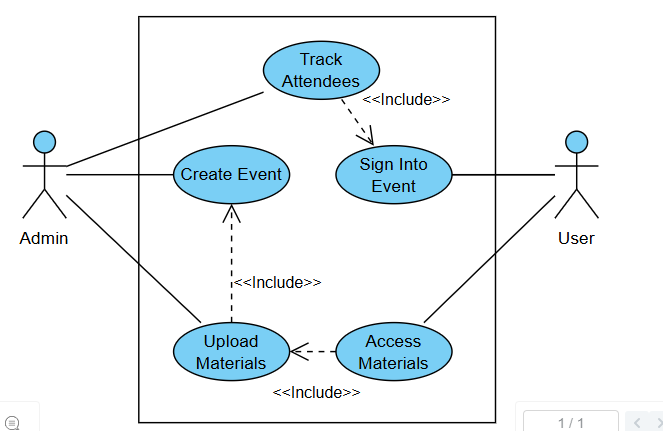
**Activity Diagram**

A diagram of a flowchart

AI-generated content may be incorrect.

#### 1.6 Event Management

Create events, track attendees, and upload materials.

**

**Use Case Description:**

* **Create Event:** This use case allows Admins/Officials to set up new community events. This includes defining event details such as name, date, time, location, description, and capacity.
* **Upload Materials:** Included when creating or managing an event, this use case enables Admins/Officials to upload documents or other materials related to an event (e.g., agendas, presentations, flyers).
* **Track Attendees:** This use case allows Admins/Officials to monitor and manage the list of attendees for an event. This might involve viewing registrations and checking participants in.
* **Sign Into Event:** This use case allows Users to register for or sign into an event, indicating their attendance. This is often a prerequisite for being tracked by the "Track Attendees" use case.
* **Access Materials:** This use case allows both Admins/Officials (likely for management) and Users (attendees or interested parties) to view and download materials that have been uploaded for an event.

**Actors:** Admin/Official, User  
  
**Activity Diagram**  
  
A screenshot of a computer screen

AI-generated content may be incorrect.

### Module 2: Mobile Functionalities

#### 1.1 Push Notifications

Real-time alerts for emergencies, events, and requests.

##### Use Case Diagram

A diagram of a system

AI-generated content may be incorrect.

##### Use Case Description

**Actors**

* **Admin**: Responsible for creating events, uploading materials, and tracking attendees.
* **User**: Participates in events by signing in and accessing materials.

**Use Cases**

* **Create Event**: Admin can create an event. This use case includes:
  + **Upload Materials**: Admin can upload event-related content.
* **Sign Into Event**: Users can sign in to an event. This use case includes:
  + **Track Attendees**: Tracks who attends the event.
* **Access Materials**: Users can access uploaded materials, which is linked to the **Upload Materials** use case.

**Relationships**

* **Includes**:
  + The **Create Event** use case includes the **Upload Materials** use case.
  + The **Sign Into Event** use case includes the **Track Attendees** use case.
  + The **Access Materials** use case includes the **Upload Materials** use case.

**Summary**

The system allows admins to create events, track attendees, and manage materials, while users can sign into events and access shared materials. The use of <<Include>> relationships highlights the dependencies between use cases for streamlined functionality.

##### Activity Diagram

A diagram of a system

AI-generated content may be incorrect.

##### Wireframe

A screenshot of a phone

AI-generated content may be incorrect.

#### 1.2 QR Code Scanning

Scan codes for event check-ins and service access.

##### Use Case Diagram

A diagram of a system

AI-generated content may be incorrect.

##### Use Case Description

**Actors**

1. **Barangay Official**
   * Can view service statistics and approve service access.
2. **System Administrator**
   * Manages the QR system and generates reports.
3. **Resident**
   * Accesses barangay services, displays their QR code, and registers for a QR ID.
4. **Barangay Staff**
   * Scans QR codes, verifies resident identity, processes service requests, and records transactions.

**Use Cases and Relationships**

* **View Service Statistics** (Barangay Official)
  + Extends the **Approve Service Access** use case, indicating that statistics may require approval conditions.
* **Manage QR System** (System Administrator)
  + Allows the administrator to manage the overall system that tracks QR IDs.
* **Access Barangay Services** (Resident)
  + Includes:
    - **Display Personal QR Code** (a mandatory part of accessing services).
* **Register for QR ID** (Resident)
  + A standalone process for residents to acquire a QR code.
* **Scan Resident QR Code** (Barangay Staff)
  + Includes:
    - **Verify Resident Identity** (to ensure accurate identification).
* **Process Service Request** (Barangay Staff)
  + Includes:
    - **Record Transaction** (documenting each service transaction).

**Key Relationship Notes**

* **<<Include>>**: Shows that certain actions are necessary for completing a primary use case (e.g., accessing services requires displaying a QR code).
* **<<Extend>>**: Indicates optional functionality or conditional actions (e.g., approving service access extends viewing service statistics).

**Summary**

This system efficiently integrates various roles with QR-based identity management, ensuring streamlined barangay service access, identity verification, and transaction recording. The clear use of <<Include>> and <<Extend>> relationships helps outline dependencies and optional features.

##### Activity Diagram

A diagram of a software system

AI-generated content may be incorrect.

##### Wireframe

A screenshot of a phone

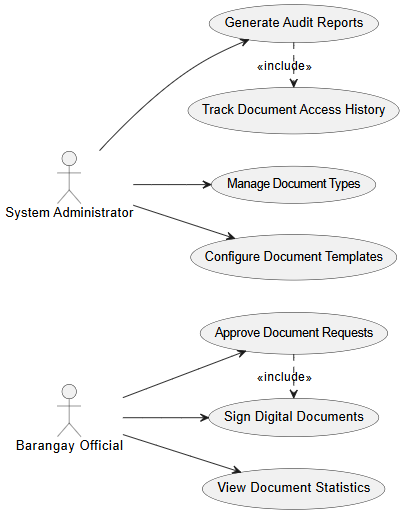
AI-generated content may be incorrect.A blue rectangular object with white lines

AI-generated content may be incorrect.

#### 1.3 Request Downloadable Documents

Request, download, and track barangay documents.

##### Use Case Diagram



A diagram of a document

AI-generated content may be incorrect.

##### Use Case Description

**Actors**

1. **System Administrator**
   * Manages system configurations and document types.
2. **Barangay Official**
   * Approves requests, signs documents, and views statistics.
3. **Barangay Staff**
   * Manages document processing and uploads.
4. **Resident**
   * Requests, downloads, verifies, and tracks documents.

**Use Cases and Relationships**

1. **System Administrator**

* **Generate Audit Reports**
  + Includes:
    - **Track Document Access History** (ensures audit reports contain access data).
* **Manage Document Types**
  + Independent task to organize document categories.
* **Configure Document Templates**
  + Manages predefined templates for documents.

1. **Barangay Official**

* **Approve Document Request**
  + Includes:
    - **Sign Digital Documents** (required to confirm document authenticity).
* **View Document Statistics**
  + Independent task for data analysis and reporting.

1. **Barangay Staff**

* **Process Document Requests**
  + Includes:
    - **Upload Documents** (mandatory to fulfill requests).
* **Review Document Requests**
  + Standalone process to assess document requests.

1. **Resident**

* **Request Document**
  + Includes:
    - **Browse Available Documents** (to select needed files).
* **Download Document**
  + Includes:
    - **Verify Document Authenticity** (ensures files are verified before download).
* **Track Document Request**
  + Includes:
    - **View Request Status** (for real-time updates).

**Key Relationship Notes**

* **<<Include>>**: Shows necessary steps for the primary use case to function properly.
* For instance:
  + **Track Document Access History** is crucial for generating accurate **Audit Reports**.
  + **Sign Digital Documents** is essential to **Approve Document Request**.

**Summary**

This system effectively supports a structured document management process with clear responsibilities for administrators, officials, staff, and residents. The detailed relationships highlight crucial dependencies ensuring security, document integrity, and efficient service delivery.

##### Activity Diagram

A flowchart of a software application

AI-generated content may be incorrect.

##### Wireframe

A screenshot of a phone

AI-generated content may be incorrect.

## Non-functional requirements

### Performance

##### Details

### Security

##### Details

### Reliability

##### Details