**PAYROLL MANAGEMENT AND PAYMENT RECONCILIATION SYSTEM FOR SOCIAL PENSION PROGRAM WITH INTEGRATED DATA QUALITY ASSURANCE**

**Title**

Proponents

Proponent1

Proponent2

**Notre Dame of Marbel University**

**MIT 251 – Software Engineering**

Brenda M. Balala, MIT

December 2023

**Table of Contents**

Contents

[Introduction 4](#_heading=h.1fob9te)

[Background of the Study 4](#_heading=h.3znysh7)

[Statement of the Problem 4](#_heading=h.2et92p0)

[Objectives of the Study 5](#_heading=h.tyjcwt)

[Concept of the Study 6](#_heading=h.3dy6vkm)

[Conceptual Framework 7](#_heading=h.1t3h5sf)

[System Architecture 8](#_heading=h.17dp8vu)

[Data Flow and Processing 8](#_heading=h.3rdcrjn)

[Security and Authentication 8](#_heading=h.26in1rg)

[Deployment and Scalability 8](#_heading=h.lnxbz9)

[Methods 9](#_heading=h.35nkun2)

[Hardware 9](#_heading=h.1ksv4uv)

[Software 9](#_heading=h.44sinio)

[Frontend 9](#_heading=h.2jxsxqh)

[Backend 9](#_heading=h.z337ya)

[Database 9](#_heading=h.3j2qqm3)

[Procedures 10](#_heading=h.1y810tw)

[Design Methodology 10](#_heading=h.4i7ojhp)

[Development Model 11](#_heading=h.1ci93xb)

[Requirements Analysis 11](#_heading=h.3whwml4)

[Functional Requirements 11](#_heading=h.2bn6wsx)

[Non-Functional Requirements 11](#_heading=h.qsh70q)

[Requirements Documentation 12](#_heading=h.3as4poj)

[System Use Case diagram 12](#_heading=h.1pxezwc)

[User Stories 12](#_heading=h.49x2ik5)

[Acceptance Criteria 12](#_heading=h.2p2csry)

[System Design 12](#_heading=h.147n2zr)

# Introduction

## Background of the Study

The social pension for itinerant senior citizens is a government initiative designed to address the needs of elderly individuals, particularly those in vulnerable sectors. This program focuses on providing financial support to senior citizens who have not had the opportunity to benefit from programs like the Social Security System (SSS) or the Government Service Insurance System (GSIS). It aims to ensure that even the most marginalized and economically disadvantaged seniors, especially those without formal pension coverage, receive the assistance they need to live with dignity and security in their later years.

Meanwhile, the process of disbursing social pensions remains entirely manual, with no technological systems in place to streamline or expedite the procedure. As a result, disbursement officers frequently encounter significant challenges, leading to delays and sometimes the failure to complete disbursements within the designated time frame. One major issue arises during the identity verification process, which often takes longer than expected, particularly when beneficiaries lack the necessary documentation. In some cases, disbursing officers also face difficulties in verifying the legitimacy of individuals attempting to claim benefits despite not being included on the official list. These challenges are inherent in the current manual system and are unlikely to improve without the integration of technology to automate and simplify the process.

In addition to the problems mentioned, grievances are also increasing due to various reasons, such as those who did not receive grants or received but not enough. There are also beneficiaries who receive double which is very worrying for the agency because it greatly affects, not only the overall accomplishment of the agency but also to the thrust of the public to the agency. The disbursing officers will also have difficulty in liquidating the payroll especially since the COA is focusing on program operation.

To solve these problems, the agency needs a centralized database system with strict policies (database contains and roles). through this, the process will be accelerated, the integrity of the data will be protected, and possible problems will be mitigated in the present. The proposed Comprehensive Payroll Management and Payment Reconciliation System will address key issues in the social pension program, such as data inconsistencies, delays, and challenges in tracking beneficiary statuses. By automating data validation, the system will ensure accurate beneficiary information, reduce errors and fraud, and streamline payroll processing and payment reconciliation for timely disbursements. Real-time monitoring and regular data updates will keep beneficiary information up to date, improving payroll accuracy. Scalable technologies like Redis, Celery, and Docker will handle growing data loads, while security measures like role-based access control and data encryption will protect sensitive information.

.

## Statement of the Problem

1. Lack of data quality assurance in the newly endorsed beneficiary from the LGU would result in data inconsistencies like duplicate and fraudulent entries that could result in larger problems like increase in grievances due to lower or higher entitlement.
2. Manual data processing is susceptible for delay in government’s deliverables due to limited timeline particularly in the disbursement and liquidation processing.
3. The agency is experiencing problems in tracking their beneficiary status that hinders in the preparation of the listing of beneficiaries for payroll preparation.

## Objectives of the Study

General Objective:

To develop a web-based application system that would automate the Social Pension Disbursement Processes and provide real-time tracking of beneficiaries.

Specific Objectives:

1. To Develop and implement a data validation and quality assurance mechanism.
2. To automate the payroll processing, disbursement, and liquidation process.
3. To Integrate real-time monitoring and realtime reporting features to facilitate payroll process.

## Conceptual Framework

**Figure1** Conceptual Framework



**Users:**

Project Development Officers

* Encodes waitlisted pension beneficiaries into the system.
* Validates the identities of beneficiaries.
* Assists in the disbursement process.

Quality Assurance Focal

* Ensures the integrity and accuracy of beneficiary data.
* Verifies the payroll before processing.

Monitoring and Evaluation Officer

* Oversees overall program performance.
* Ensures the system effectively tracks beneficiaries and disbursements.

Regional Program Coordinator

* Supervises the program's implementation and operations.
* Provides recommendations to the Regional Director for approvals.

Financial Analyst

* Prepares financial reports for transparency and accountability.
* Supports payroll reconciliation to ensure financial accuracy.

Regional Director

* Holds final authority for approving beneficiary lists and payroll processing.
* Makes key program-related decisions to ensure effective implementation.

**Data Process:**

The conceptual framework for payroll processing, data quality assurance, payroll approval, disbursement, and liquidation. This framework is built on the interaction between three key entities: system users, inputs, and processes—all working together to ensure efficiency, transparency, and accountability.

First, we have the six key users in the system. Project Development Officers handle the encoding and validation of potential beneficiaries. Quality Assurance Focal ensures data integrity by identifying inconsistencies and verifying payroll details. Regional Program Coordinator oversees the entire process and makes recommendations for approval. Financial Analyst is responsible for preparing financial reports and reconciling payroll records. Regional Director holds the authority for final approvals, while the Monitoring and Evaluation Officer continuously assesses program performance to ensure everything runs smoothly.

Next, we have the inputs, which serve as the foundation for payroll processing. These include the registration of potential beneficiaries, the results of field validation to confirm senior citizen identities, and the notice of approved payroll actions.

Now, let’s talk about the process. It all starts with encoding and validating potential beneficiaries, followed by updating their data after validation. Then, quality assurance measures—such as deduplication and identifying inconsistencies—are carried out by the Quality Assurance Focal. Once verified, the Regional Program Coordinator makes a recommendation, and the Regional Director gives approval for payroll processing. The Financial Analyst reviews and reconciles payroll data, submits it for approval, and then the funds are obligated and disbursed. Finally, disbursement and liquidation take place to ensure beneficiaries receive their pensions properly.

Throughout this entire process, the Monitoring and Evaluation Officer plays a crucial role in analyzing data, tracking progress, and ensuring smooth program operations. This structured approach guarantees that pension disbursements are processed accurately, securely, and in compliance with program guidelines.

### System Architecture

* Diagram or description of the microservices-based architecture
* Explanation of how different services (e.g. product management, user authentication) communicate and interact.

### Data Flow and Processing

* Description o how data flows through the system, from user input to database storage.
* Diagram to represent data flow.

### Security and Authentication

* Overview of security measures (e.g. OAuth 2.0, JWT, data encryption)
* Authentication flow and mechanisms from ensuring secure transactions.

### Deployment and Scalability

* Explanation of deployment methods (e.g. AWS, Kubernetes)
* Scalability considerations (e.g. horizontal scaling of microservices)

# Methods

## Hardware

* *Specifications of the hardware used*
* Asus Vivo Book Intel i5, 8gb DDR4 RAM, Nvidia 940MX
* Desktop Intel i7, 16GB DDR4 RAM, Nvidia RTX 3060

## Software

* *Software used in the design and development of the system. Discuss how the software is used.* ***DO NOT*** *give the definition of the software*
* Sublime Text
* XAMPP
* SQLyog
* Celery
* Redis

### Frontend

### 1. Bootstrap 5

* Bootstrap 5 is the latest version of the popular front-end framework for developing responsive and mobile-first websites. It provides a collection of CSS and JavaScript tools for designing web layouts, forms, buttons, modals, and more. Unlike Bootstrap 4, Bootstrap 5 **removes jQuery dependency**, improves grid system flexibility, and introduces enhanced utility classes.

### 2. HTML5

* HTML5 is the latest version of HyperText Markup Language, the standard language for creating web pages. It introduces new semantic elements (<article>, <section>, <nav>, etc.), improved multimedia support (<audio> and <video>), and enhanced APIs for better interactivity, such as Web Storage, Geolocation, and Canvas.

### 3. jQuery

* jQuery is a fast, lightweight JavaScript library that simplifies HTML DOM manipulation, event handling, animations, and AJAX interactions. It allows developers to write concise code for complex tasks, reducing browser compatibility issues. Although widely used, modern frameworks like React and Vue are replacing jQuery in some projects.

### 4. Select2.js

* Select2.js is a jQuery-based plugin that enhances HTML <select> elements by adding features like search, multi-select, tagging, remote data loading (via AJAX), and customizable styling. It improves user experience when handling long lists of selectable items.

### 5. SweetAlert2.js

* SweetAlert2.js is a modern replacement for JavaScript's alert() function. It allows developers to create customizable, visually appealing, and user-friendly alert dialogs with animations, confirmation messages, and asynchronous support. Unlike its predecessor SweetAlert, SweetAlert2 is lightweight, modular, and highly configurable.

### 6. Xterm.js

* Xterm.js is a JavaScript library that provides a **terminal emulator** for web applications. It allows developers to embed a fully functional command-line interface within a webpage, making it useful for web-based SSH clients, developer tools, and cloud-based coding environments. It supports Unicode, ANSI colors, and customizable theme

### Backend

* Description of backend technologies (Node.js, Express.js)
* Explanation of microservices architectures and RESTful APIs.

RESTFul, Django, MySQL

* Explanation of microservices architectures and RESTful APIs.

django-RESTful Api

django framework

MySQL Database

### Database

* Overview or relational (PostgreSQL and NoSQL (MongoDB) databases
* Why both types are used form different purposes (structured vs unstructured data)
* *Sources and composition of data*

*{ERD}*

*{description of the ERD}*

# Procedures

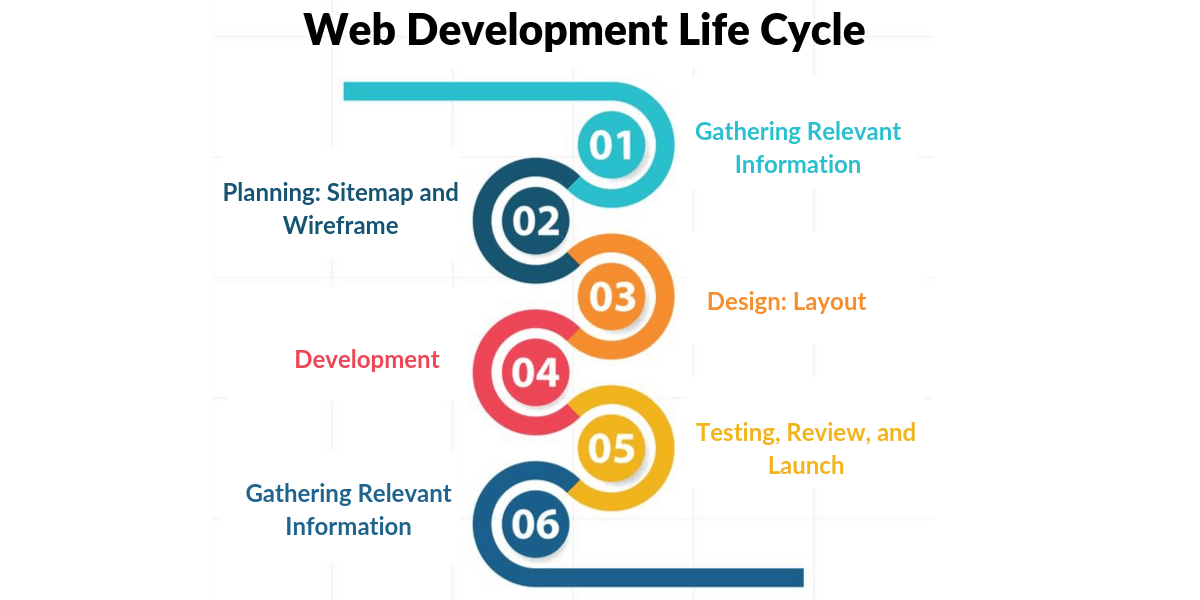
This section describes the process of system development, from design to implementation.

* *Must be sequential based on the Conceptual framework*
* *Indicate the various steps that must be taken to attain the Specific Objectives*

## Design Methodology

* *provides a logical and systematic means of proceeding with the design process as well as a set of guidelines for decision-making. The design methodology provides a sequence of activities, and often uses a set of notations or diagrams.*
* *Create a diagram of the methods to be used in software development based on the* ***Specific Objectives***
* *Example: SDLC, Web Development Life Cycle, Game Development Life Cycle*

Example

**Figure 2** Web Development Life Cycle

## Development Model

* *The model to be used in the development of the system*
* *Discuss the activities done in each stage.*
* *Explanation of the agile or waterfall development model adopted for the project.*
* *Iterative development cycles with milestones and reviews.*

## Requirements Analysis

* *Discuss the details of the current system (how the current procedures are performed)*
* *Create a* ***USE case diagram of the current system***

### Functional Requirements

1. Mobile Field Validation

* User-Agent Detection to check if the login is from a mobile device.
* Webcam/Camera Access using HTML5 MediaStream API.
* Image Upload & Storage in Django
* Verification for the beneficiary identified during disbursement.

1. Realtime Dashboard Data

* Use Django API to load information in the dashboard.
* Leverage Celery + Redis to process updates in the background.
* Using Jquery to send POST request to API

1. Automatic Data Analytics for DQA (Data Quality Assessment)

* Implement y-data and pandas dataframe for analysis.
* Use Django ORM queries or SQLAlchemy for data aggregation

1. Fuzzy-Matching Capability Using Jaro-Winkler algorithm

* Leverage the RapidFuzz library with parallel computation to efficiently capture potential duplicates, significantly improving the speed and accuracy of duplicate detection.

1. Task Queuing & Background Processing System using Celery and Redis

* Configure Celery with Redis as a message broker.
* Use task retries to handle failed tasks.
* resume task capability if the the service interupted

### Non-Functional Requirements

* Input Validation and Sanitization: Django validates and sanitizes user inputs to prevent injection attacks.
* Protection Against SQL Injection: Django ORM uses parameterized queries to eliminate SQL injection risks.
* Cross-Site Request Forgery (CSRF) Protection: Django enforces CSRF tokens to prevent unauthorized form submissions.
* Cross-Site Scripting (XSS) Prevention: Django automatically escapes template output to block malicious scripts.
* Clickjacking Protection: Django sets X-Frame-Options headers to prevent embedding in iframes.
* Brute Force and Dictionary Attack Mitigation: Django supports rate limiting and enforces strong password policies.
* Secure Authentication and Authorization: Django provides role-based access control (RBAC) and multi-factor authentication.
* Session Hijacking Prevention: Django secures session cookies and supports automatic session expiration.
* **Encryption of Sensitive Data:** Django encrypts passwords using PBKDF2, bcrypt, or Argon2.
* **Logging and Intrusion Detection:** Django logs security events and integrates with intrusion detection tools.
* **Automated Security Updates:** Regular updates and dependency checks keep Django protected from vulnerabilities.
* **Security Headers Implementation:** Django supports CSP, Referrer Policy, and other security headers.
* **Denial-of-Service (DoS) Attack Mitigation:** Django caching and WAF integration help defend against DoS attacks.

### 

## Requirements Documentation

* *All software features are enumerated in detail by providing storyboard showing how the proposed system will look like.*
* *Requirements documentation is the description of what a particular software does or shall do. It is used throughout development to communicate how the software functions or how it is intended to operate. It is also used as an agreement or as the foundation for agreement on what the software will do. Requirements are produced and consumed by everyone involved in the production of software, including:  end-users, customers, project managers, sales, marketing, software architects, usability engineers, interaction designers, developers, and testers.*

### System Use Case diagram

* Diagram representing system use cases, including interactions between users and the system

### User Stories

* User stories detailing the needs of different user personas (e.g. buyer, vendor, admin)

### Acceptance Criteria

* Criteria used to determine if the system meets the requirements (e.g. response time, error rates)

## System Design

### Architectural Design

* ***Detailed breakdown of system components: frontend, backend, database, and security***

### Database Design

### Application Flow

* ***,***