Software Requirements Specification

for

Oldie2GenZ

Version <X.X>

Prepared by

Group Name: Fusion Force

Anjani Komaravelli	SE22UCSE029	se22ucse029@mahindrauniversity.edu.in
Chaitanya Deepthi	SE22UCSE049	se22ucse049@mahindrauniversity.edu.in
Shreeya Konda	SE22UCSE249	se22ucse249@mahindrauniversity.edu.in
Nikita Reddy	SE22UCSE001	se22ucse001@mahindrauniversity.edu.in
Rishita Chava	SE22UCSE071	se22ucse071@mahindrauniversity.edu.in
Akshay Palutla	SE22UCSE192	se22ucse192@mahindrauniversity.edu.in
Karthik Pataneni	SE22UCSE200	se22ucse200@mahindrauniversity.edu.in

Instructor: Vijay Rao

Course: Software Engineering

Lab Section:

Teaching Assistant: Murali Krishna

Date: 10/03/2025

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft Type and Number	Full Name	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.	00/00/00

1. Introduction

This Software Requirements Specification (SRS) document outlines the requirements for "Oldie2GenZ," a web application designed to assist senior citizens in accessing online services. It provides features such as government service access, digital learning, and health reminders, ensuring a simple and accessible user experience.

1.1. Document Purpose

This document specifies the requirements for "Oldie2GenZ," created by Fusion Force. It serves as a guide for developers, testers, and project managers. The software helps senior citizens with government service access, medication schedules, and digital literacy.

It ensures that all stakeholders understand the system's functionality and design. The document acts as a roadmap to ensure the software meets expectations and user needs

1.2. Product Scope

"Oldie2GenZ" is a user-friendly web application for senior citizens. It provides access to government services, health reminders, and digital learning. The platform is designed to be simple and accessible.

The system helps users stay informed through real-time notifications, tutorials, and reminders. Unlike other platforms, it is specifically tailored to senior citizens' needs, ensuring ease of use.

1.3. Intended Audience and Document Overview

This document is for developers, testers, project managers, and clients, including the professor reviewing the project. Developers will use it for implementation, testers for validation, and managers for tracking progress.

The document is structured as follows: Section 2 provides a system overview, Section 3 lists requirements, Section 4 details interfaces, and Section 5 covers system attributes. Readers should start with the overview before diving into specific sections.

1.4. Definitions, Acronyms and Abbreviations

Fusion Force: Development team

Jeevan Pramaan: Digital life certificate for pensioners

Oldie2GenZ: Web application for senior citizens

SRS: Software Requirements Specification

UI/UX: User Interface/User Experience

1.5. Document Conventions

This document follows IEEE SRS standards, using **Arial font, size 11 or 12**, with **single spacing and 1-inch margins**. Bold is used for section titles, and italics highlight important points.

Consistent naming conventions are maintained for clarity, ensuring easy navigation and readability.

1.6. References and Acknowledgments

References:

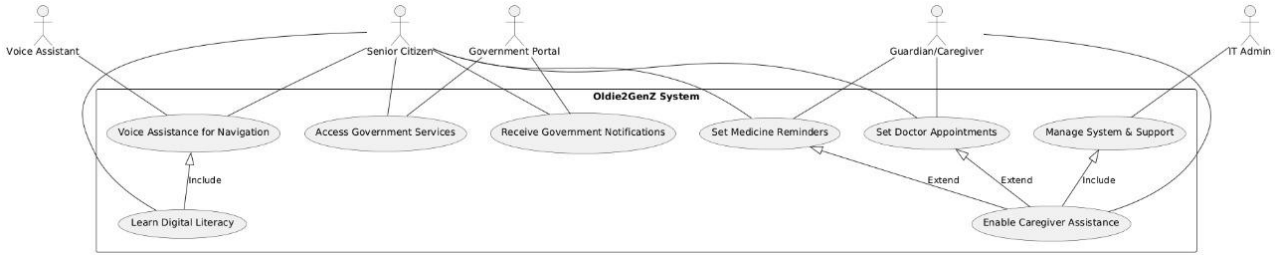
- ***IEEE Software Engineering Standards*** *– Guidelines for software documentation and requirement specifications.*
- ***WCAG (Web Content Accessibility Guidelines)*** *– Standards for making web content accessible for senior citizens.*
- ***Government of India Portals*** *– Official websites for Jeevan Pramaan, pension services, and other senior citizen services.*
- ***User Interface Design Principles*** *– Best practices for accessible and user-friendly digital interfaces.*
- ***Database Management (MySQL)*** *– Guidelines for storing and managing user data securely.*

2. Overall Description

2.1 Product Overview

The "Oldie2GenZ - Bridging Generations, Embracing the Now!" website is a new, self-contained digital platform designed to support senior citizens by providing them with a one-stop solution for essential online services. Many elderly users struggle with digital literacy, keeping track of government deadlines, managing their medication schedules, and navigating social media platforms. This product aims to bridge this digital divide by offering a simple, intuitive, and accessible web application.

The website acts as a hub that integrates multiple services, including access to government portals, digital literacy tutorials, and personalized health reminders. By streamlining these functionalities, "Oldie2GenZ" ensures senior citizens can independently and confidently access the internet without external assistance.



2.2 Product Functionality

The website will include the following key functionalities:
Access to Government Services:

1. Direct links to portals such as Jeevan Pramaan, pension services, and health benefits.
2. Notifications about the remainders which are set.

Smart Reminders for Health & Appointments:

1. Customizable medicine intake and doctor appointment alerts.

Digital Literacy & Internet Education:

1. Step-by-step tutorials with voice guidance for online platforms (e.g., YouTube, WhatsApp, Facebook).

Senior-Friendly Interface with Caregiver Support:

1. Large buttons, simple layouts, and easy navigation.
2. Caregivers can remotely assist in setting up reminders and notifications.

2.3 Design and Implementation Constraints

- **Use of COMET Methodology:** The software design must follow the **COMET (Concurrent Object Modelling and Architectural Design Method)**, ensuring a structured and scalable architecture.
- **UML (Unified Modelling Language) for Modelling:** All system diagrams and designs must be created using **UML (Unified Modelling Language)** for clear and standard representation.
- **Hardware Limitations:** The platform must function efficiently on low-end devices to ensure accessibility for senior citizens.
- **Security Measures:** Strong encryption and authentication methods must be implemented to protect user data.
- **Integration with Government API (Application Programming Interface):** The system should support real-time connections with government portals for seamless service access.
- **Accessibility Compliance:** The platform must meet **WCAG accessibility standards** to cater to senior citizens' usability needs.

2.4 Assumptions and Dependencies

- **Reliable Internet Access:** The application assumes that users will have a stable internet connection for accessing services.
- **Government API Availability:** The system relies on external government APIs to fetch real-time pension and healthcare data.
- **Cross-Platform Compatibility:** The application is designed to work smoothly on both desktop and mobile browsers without additional installations.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The **Oldie2GenZ** website will have a simple and intuitive user interface designed specifically for senior citizens. The interface will include:

- **Large Buttons & Readable Text**: To ensure easy navigation.
- **Voice Assistance**: To help users who have difficulty reading.
- **Interactive Tutorials**: Step-by-step guides with videos on using digital platforms.
- **Notification Panel**: Users will receive important reminders and government updates.

Users will interact with the system via a web browser, using simple menus and guided steps. The system will be designed with **accessibility in mind**, ensuring ease of use for elderly users.

3.1.2 Hardware Interfaces

The system will be accessible through:

- **Desktop Computers & Laptops**

The platform will not require any specialized hardware and will work on any internet-enabled device with a web browser.

3.1.3 Software Interfaces

*The **Oldie2GenZ** system will integrate with:*

- ***Government Service Portals** (e.g., Jeevan Pramaan) for redirection.*
- ***Database (MySQL)** to store user preferences, reminders, and authentication details.*
- ***Notification APIs** to send medicine and appointment reminders.*
- ***Security & Authentication Modules** for user login and data privacy.*

3.2 Functional Requirements

3.2.1 User Authentication & Personalization:

- Users must sign in to view and manage their personalized settings and services.
- Caregivers can create accounts to assist senior users.

3.2.2 Access to Government Services:

- Users can access and get redirected to relevant government portals.
- The system will provide notifications about remainders which are sent

3.2.3 Reminder System for Health & Appointments:

- Users can set up **medication and appointment reminders** (configurable by a caregiver if needed).

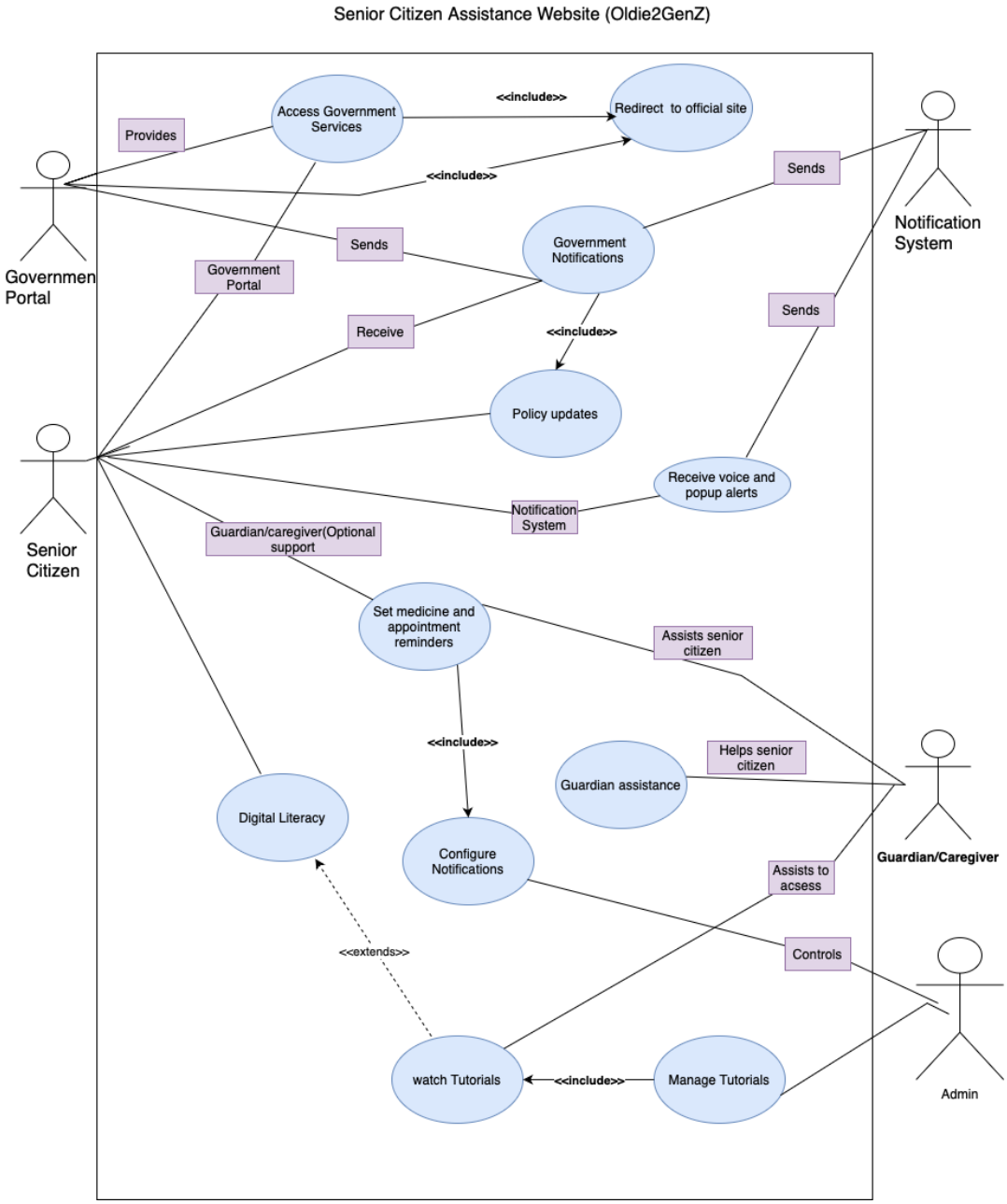
3.2.4 Digital Literacy & Learning:

- Interactive **step-by-step tutorials** on using social media, YouTube, WhatsApp, and other online services.
- A **help section** for common internet-related queries.

3.2.5 User-Friendly Interface:

- Large font sizes, simple layout and **high-contrast themes** for accessibility.

3.3 Use Case


Senior Citizen Assistance Website (Oldie2GenZ)
Access Government Services
<<include>>
Redirect to official site
Provides
<<include>>
Sends
Notification System
Sends
Government Notifications
Government Portal
Receive
<<include>>
Sends
Policy updates
Receive voice and popup alerts
Notification System
Guardian/caregiver(Optional support
Set medicine and appointment reminders
Assists senior citizen
<<include>>
Helps senior citizen
Guardian assistance
Guardian/Caregiver
Digital Literacy
Configure Notifications
Assists to acsess
<<extends>>
Controls
watch Tutorials
<<include>>
Manage Tutorials
Admin
Senior Citizen

Model:

4. Other Non-functional Requirements

4.1 Performance Requirements

The system should meet the following performance expectations:
1. Page Load Time:
The website should load within 3 seconds under normal internet conditions (at least 4G or broadband).
Government service redirections should complete within 5 seconds to ensure seamless navigation.
2. Reminder Notifications:
Medicine and appointment reminders should trigger at the exact scheduled time with a maximum delay of 1 second.
should be played within 2 seconds of notification activation.
3. Tutorial and Media Loading:
Video tutorials should start playing within 2 seconds after clicking, with buffering not exceeding 3 seconds for the first load.
Images and text content should render completely within 2 seconds of accessing a page.
4. Concurrent User Handling:
The system should support at least 100 concurrent users without noticeable performance degradation.
Server response time should not exceed 1.5 seconds under normal load conditions.
5. Database Query Performance:
Retrieval of user settings (reminders, preferences) should take less than 1 second.
Government policy updates and notifications should be fetched every 6 hours and stored locally for faster access.

4.2 Safety and Security Requirements

1. User Authentication and Data Protection:
Users (or their guardians) must authenticate using a secure login system (OTP-based or password-protected).
User data, including reminders and personal information, must be encrypted using AES-256 before storage.
Data transmission must use HTTPS to prevent unauthorized access.

2. Access Control:
Caregivers should have limited access to the user's account only for managing reminders and government service tracking.
Users must be notified whenever a caregiver makes changes to their reminders.
3. Privacy Compliance:
The system must comply with GDPR (General Data Protection Regulation) and India's IT Act (2000) regarding data privacy.
Users must be able to delete their data permanently upon request.
4. Fail-Safe Mechanisms:
If the website crashes, it should automatically restart within 5 seconds.
In case of server downtime, a backup service should provide essential reminders via SMS or email.

4.3 Software Quality Attributes

4.3.1 Usability

The system will have a simple interface with large buttons and text for easy navigation.
Tutorials will have step-by-step guidance with visual and audio aids for seniors.

4.3.2 Reliability

The system should have 99.9% uptime, ensuring users can access services at all times.
Data should be automatically backed up every 24 hours to prevent loss.
The application should handle sudden internet disconnections and resume services once connectivity is restored.

4.3.3 Maintainability

The codebase will follow modular design, making it easy to update and debug.
Third-party APIs (e.g., government services) will be loosely coupled, allowing easy replacements if needed.
Regular security patches will be applied every 3 months.

4.3.4 Adaptability

The system should work across mobile, tablet, and desktop devices.
Language settings should be configurable, with support for at least 3 regional languages.
Future updates should allow integration with voice assistants like Alexa and Google Assistant.

5. Other Requirements

1. Database Requirements:

The system will use MySQL for storing user profiles, reminders, and preferences.
Government service updates will be fetched via APIs and stored in a cache database (Redis) for faster access.

2. Internationalization & Accessibility:

The platform should support multiple languages, starting with English, Hindi, and Telugu.
Users with vision impairment should have a text-to-speech feature for navigation.

3. Legal Requirements:

Compliance with Indian government regulations on digital data storage and processing.
Must adhere to GDPR if the platform expands internationally.

Appendix A – Data Dictionary

The system includes various data elements necessary for providing a seamless experience to senior citizens accessing different online services. Below is a description of key variables, their roles, and related functionalities:

- **User ID**: A unique integer assigned to each user for identification. This is auto-generated and used for login and profile management.
- **User Name**: A string representing the name of the senior citizen. It is displayed on the profile and used for personalization.
- **Email**: A string storing the user's email address in a valid format. It is required for login authentication and sending notifications.
- **Password**: A secure string stored in an encrypted format, used for authentication purposes.
- **Service Category**: A string defining the type of service accessed, such as healthcare, banking, grocery delivery, or social engagement. This helps categorize services for easy navigation.
- **Service Name**: A specific service within a category, such as "Doctor Consultation" or "Online Banking." It tracks user preferences and helps in service recommendations.
- **Appointment Date**: A date field in YYYY-MM-DD format that stores scheduled service appointments. It is required for booking confirmations.
- **Location**: A string representing the user's preferred service location, such as a city or PIN code. It is used to filter location-based services.
- **Feedback**: A text input storing user feedback on services used. It is utilized for service improvement and user experience enhancement.
- **Session Status**: A string representing the current state of the user’s session, which can be "Active," "Inactive," or "Expired." This helps in managing session timeouts and user activity tracking.
- **Support Request ID**: A unique integer assigned to each customer support request. It helps track user complaints and inquiries effectively.

Appendix B - Group Log

Meeting 1: Project Kickoff

- ***Date & Time:*** *20-02-2025, 5:30 pm*
- ***Attendees:*** *All*
- ***Agenda:***
  - *Understanding the problem statement*
  - *Defining the scope of the website for senior citizens*
  - *Identifying initial functional requirements*
  - *Assigning roles and responsibilities*
- ***Decisions Made:***
  - *Website will provide services like healthcare, banking, grocery delivery, and social engagement*
  - *Each member will research relevant online services for senior citizens*
  - *Next meeting scheduled for requirement gathering*

Meeting 2: Requirement Gathering

- ***Date & Time:****01-03-2025,3:30 pm*
- ***Attendees:*** *All*
- ***Agenda:***
  - *Reviewing research findings*
  - *Listing key functionalities for the website*
  - *Discussing user-friendly design considerations*
- ***Decisions Made:***
  - *Major functionalities finalized (user authentication, service selection, appointment booking, feedback, etc.)*
  - *Website will have a simple, accessible UI with large fonts and voice-assist features*
  - *Next meeting scheduled for SRS documentation*

Meeting 3: Drafting the SRS

- ***Date & Time:*** *03-03-2025, 3:30 pm*
- ***Attendees:*** *All*
- ***Agenda:***
  - *Structuring the Software Requirements Specification (SRS)*
  - *Assigning sections to team members for writing*
- ***Decisions Made:***
  - *Each member assigned a section (Product Overview, Functionality, Constraints, Assumptions, etc.)*
  - *Use UML modeling for design representation*
  - *Next meeting scheduled for SRS review*

Meeting 4: Review & Finalization

- ***Date & Time:*** *04-03-2025,4:30 pm*
- ***Attendees:*** *All*
- ***Agenda:***
  - *Reviewing SRS document*
  - *Making final edits based on feedback*
  - *Ensuring completeness and accuracy*
- ***Decisions Made:***
  - *Document formatted and reviewed for clarity*
  - *Final version submitted*

Group Activities Summary:

- *Conducted research on senior citizen services*
- *Defined key features and user requirements*
- *Developed UML diagrams and wireframes*
- *Drafted, reviewed, and finalized the SRS document*

This log provides a detailed record of the team's collaboration and effort in producing the document.