Software Requirements Specification

for

AgeWise

Version <1.0>

Prepared by MarsCode

Group Name: MarsCode

Chetan Naidu	**SE22UARI137**	**se22uari137@mahindrauniversity.edu.in**
Rahul Kaja	**SE22UARI139**	**se22uari139@mahindrauniversity.edu.in**
PVS Sivamani	**SE22UARI119**	**se22uari119@mahindrauniversity.edu.in**
Nandan	**SE22UARI001**	**se22uari001@mahindrauniversity.edu.in**
Goutham	**SE22UARI010**	**se22uari061@mahindrauniversity.edu.in**

Instructor: **Arun Avinash Chauhan**

Course: **Software Engineering**

Lab Section: ***AI-2***

Teaching Assistant: ***Swapna S***

Date: **11-03-2025**

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1	MarsCode	Initial draft of the Software Requirements Specification (SRS) for AgeWise outlining	11/03/2025

Version	Primary Author(s)	Description of Version	Date Completed
		system functionalities, use cases, and constraints based on the project scope.	

1 Introduction

1.1 Document Purpose

This Software Requirements Specification (SRS) document defines the functional and non-functional requirements for AgeWise, a web-based platform designed to provide senior citizens with an intuitive and accessible way to navigate essential online services. The platform offers features such as easy access to government services, medication reminders, digital literacy tutorials, personalized notifications and other features. The system prioritizes ease of use through accessibility-focused design, including large fonts, high contrast, voice assistance, and simple navigation. This document serves as a guideline for the development team, ensuring that all features align with the intended user experience and system objectives.

This SRS covers the core functionalities, system constraints, and technical specifications necessary for developing a scalable, secure, and user-friendly solution. It outlines the scope of the project, detailing both end-user and administrative capabilities, while also addressing essential performance, security, and usability considerations. While this document primarily focuses on the web application, future expansions to mobile platforms or additional integrations may be considered in later releases.

1.2 Product Scope

AgeWise is a web-based platform designed to simplify internet usage for senior citizens by providing a one-stop solution for accessing essential services, learning digital skills, and managing daily reminders. The platform enables users to access for government services, receive personalized medication and appointment reminders, and access step-by-step tutorials on using the internet, including social media, video calling, and email. To enhance accessibility, AgeWise incorporates features such as large fonts, high contrast, voice assistance, and intuitive navigation, ensuring ease of use for individuals with varying levels of digital literacy.

The primary objective of AgeWise is to bridge the digital gap for senior citizens by providing a secure, intuitive, and efficient platform that enhances their independence and quality of life. The system benefits users by reducing the complexity of accessing essential online services, providing timely reminders for important tasks, and promoting digital literacy. Additionally, it includes a guardian-supported account management feature, allowing family members to assist seniors when needed. By fostering greater self-reliance and confidence in using technology, AgeWise empowers seniors to stay informed, connected, and engaged with the digital world.

1.3 Intended Audience and Document Overview

This document is intended for a variety of stakeholders, each with different roles and responsibilities in the project. Below is a breakdown of the key audiences and their purpose in using this document:

Developers (Frontend & Backend)

- *Developers will use this document to understand the platform's functionality, system behavior, and technical constraints.*

- *The functional requirements will help them implement user authentication, notifications, reminders, and accessibility features.*
- *The system architecture and design section will provide insights into the platform's structure, guiding their choice of frameworks, APIs, and database schema.*
- *The document also specifies the non-functional requirements, such as security, scalability, and performance considerations, which developers must adhere to while building the system.*

Project Managers

- *Project managers will use this document to ensure that the development process aligns with the project's goals, timeline, and budget.*
- *The scope and objectives sections will help them track the system's progress and ensure that all deliverables meet stakeholder expectations.*
- *By referencing the functional and non-functional requirements, they can ensure the team follows a structured approach in system development.*
- *The risk assessment and mitigation strategies (if included) will aid in identifying potential project risks and planning necessary interventions.*

Testers and Quality Assurance (QA) Teams

- *Testers will refer to this document to design comprehensive test cases that verify the system's correctness, reliability, and usability.*
- *The functional requirements will help them ensure that features such as government service access, reminders, tutorials, and voice assistance work as expected.*
- *Non-functional requirements like system performance, security, and accessibility will be validated against industry standards.*
- *The use case scenarios section will help them develop real-world test cases to simulate user interactions with the system.*

Clients

- *While clients (senior citizens and their guardians) are not directly involved in development, they may refer to this document to understand how the platform is structured to meet their needs.*
- *The Product Scope and Purpose sections provide a high-level overview of the system, describing its benefits and goals.*
- *The document will also help guardians or caregivers understand the reminder system, accessibility features, and guardian-assisted accounts, which allow them to support senior users in managing notifications and online services.*

System Administrators

- *Administrators are responsible for managing user accounts, content moderation, and overall system maintenance.*
- *The admin features section outlines their capabilities, such as account approval, content moderation, and system analytics.*
- *Security considerations in the non-functional requirements section help ensure that sensitive user data is handled securely.*

Professor (Avinash Arun Chauhan)

- *Since this project is being developed as part of an academic requirement, the professor evaluating it will use this document to assess its completeness, clarity, and adherence to software engineering best practices.*
- *The document organization, structured definitions, and clarity of requirements will determine the quality of the submission.*
- *The professor may also verify if the system addresses real-world problems effectively, particularly in terms of accessibility and usability for senior citizens.*

The document is organized into the following sections:

Introduction
Provides a high-level overview of the system, including its purpose and intended audience.
Describes why the platform is needed and the problems it addresses for senior citizens.

Product Scope
Defines the core functionalities of the platform, such as reminders, tutorials, and government service redirection.
Outlines the benefits for senior citizens, including improved digital literacy and easy access to essential services.

Specific Requirements
Details the functional requirements, such as authentication, notifications, and AI-driven support.
Defines use case scenarios illustrating how users interact with the system.
Specifies external interface requirements, including user, hardware, and software interfaces.

Non-Functional Requirements
Covers performance, security, and accessibility standards for the platform.
Ensures that the system meets usability criteria for senior citizens with limited digital experience.

Other Requirements
Lists hardware/software constraints, dependencies, and limitations affecting development.
Defines assumptions such as internet access availability and expected user behavior and additional requirements that do not fall into the above categories

Reading sequences for each reader type

General Overview (For All Readers)

Best for: All readers (Developers, Project Managers, Testers, Clients, Instructors) Purpose: Provides a high-level understanding of the system before diving into technical details.

1.1 Document Purpose – Understand what the SRS document is about.

1.2 Product Scope – Get an overview of the system's functionalities and objectives.

1.3 Intended Audience and Document Overview – See who the document is meant for.

1.4 Definitions, Acronyms, and Abbreviations – Familiarize yourself with key technical terms.

2.1 Product Overview – Understand the core concept of AgeWise.

Technical Overview (For Developers & Designers)

Best for: Developers, UI/UX Designers, System Architects
Purpose: Provides insight into system functionalities, architecture, and constraints.

2.2 Product Functionality – Understand the major features the system will offer.

2.3 Design and Implementation Constraints – Learn about system limitations and technology stack.

3.1 External Interface Requirements – Get details on UI, hardware, and software integrations.

3.2 Functional Requirements – Read about each system feature in detail.

3.3 Use Case Model – See how users will interact with the system.

Testing & Validation (For Testers & QA Engineers)

Best for: Testers, QA Engineers
Purpose: Helps in designing test cases and verifying system behavior.

3.2 Functional Requirements – Understand system behaviors and expected outcomes.

3.3 Use Case Model – Review user scenarios and potential test cases.

4.1 Performance Requirements – Understand performance benchmarks for system validation.

4.2 Safety and Security Requirements – See security and privacy measures to be tested.

Project Management & Decision Making (For Project Managers & Clients)

Best for: Project Managers, Clients, Instructors, Government Agencies
Purpose: Helps assess project scope, feasibility, and compliance requirements.

1.2 Product Scope – Understand the project's key objectives.

2.1 Product Overview – Get a broad understanding of how the system functions.

2.4 Assumptions and Dependencies – Review potential risks and dependencies.

4.3 Software Quality Attributes – Ensure system meets usability and reliability standards.

5. Other Requirements – See additional system requirements, including compliance factors

Additional & Reference Sections

Best for: All readers who need deeper insight into system data, logs, and documentation.

Appendix A – Data Dictionary – Review key data elements in the system.

Appendix B – Group Log – See development history and team contributions.

1.4 Definitions, Acronyms and Abbreviations

AI: Artificial Intelligence
API: Application Programming Interface
AWS: Amazon Web Services
CDN: Content Delivery Network
CS: Computer Science
CSRF: Cross-Site Request Forgery (A type of security attack)
DBMS: Database Management System
GDPR: General Data Protection Regulation
GUI: Graphical User Interface
HTML: Hypertext Markup Language
HTTPS: Hypertext Transfer Protocol Secure
ISO: International Organization for Standardization
JSON: JavaScript Object Notation
JWT: JSON Web Token (Used for secure user authentication)
ML: Machine Learning
MFA: Multi-Factor Authentication
MongoDB: NoSQL Database used for storing user and system data
MySQL: Relational Database Management System
NVDA: Nonvisual Desktop Access (Screen reader software for visually impaired users)
NoSQL: Non-Relational Database Management System
OAuth: Open Authorization protocol for secure authentication
OTP: One-Time Password
PWA: Progressive Web Application
RBAC: Role-Based Access Control
React.js: JavaScript library for building user interfaces
REST API: Representational State Transfer Application Programming Interface
SQL: Structured Query Language
SRS: Software Requirements Specification
SSL/TLS: Secure Sockets Layer / Transport Layer Security
TA: Teaching Assistant
UI: User Interface
UML: Unified Modeling Language
UX: User Experience
WCAG: Web Content Accessibility Guidelines
WCAG 2.1: Web Content Accessibility Guidelines version 2.1
XSS: Cross-Site Scripting (Security vulnerability in web applications)

1.5 Document Conventions

This section defines the formatting, typographical, and naming conventions followed throughout this Software Requirements Specification (SRS) document to ensure clarity, consistency, and readability.

- *Formatting Conventions IEEE Standard Compliance: This document adheres to the IEEE formatting requirements for software documentation to maintain uniformity and professionalism. Font Style and Size: The document uses Arial, size 11 for all body text, ensuring readability. Line Spacing: The document is single-spaced to balance readability and compactness. Margins: A 1-inch margin is maintained on all four sides of the document for a clean and structured layout. Section and Subsection Titles: Main section titles (e.g., 1, 2, 3) are bold and numbered for clear hierarchy. Subsection titles (e.g., 1.1, 2.1.1) are also bold but slightly smaller than main sections to indicate subcategories. Sub-subsections (e.g., 1.1.1, 2.3.2.1) are formatted in bold but in an even smaller size for further categorization. Alignment: All text is left-aligned, except for section headings which may be centered when necessary.*
- *Typographical Conventions Italics: Used for comments, placeholder text, and any special notes within the document. Example: This section is to be filled in after the final review. Bold: Used for section headings, key terms, and important information that need emphasis. Example: Functional Requirement F1: The system shall provide a login feature. Underlining: Avoided for regular text but may be used for hyperlinks or references if necessary. Tables and Lists: Tables are used for structured data representation, such as system requirements or comparisons. Lists are used for enhancing readability, categorized into: Numbered lists (1, 2, 3) → Used for sequential steps or ordered information. Bulleted lists (•, –, →) → Used for general information and feature descriptions. Tables follow a clean and readable layout with headers bolded for clarity.*
- *Naming Conventions Acronyms: All acronyms are fully defined when first introduced in the document. A complete list of acronyms and abbreviations is provided in Section 1.4: Definitions, Acronyms, and Abbreviations. Requirement Labels: Functional and non-functional requirements are labeled systematically for easy identification. Functional requirements are labeled as F1, F2, F3, ... Performance requirements are labeled as P1, P2, P3, ... Security requirements are labeled as S1, S2, S3, ... Other non-functional requirements are labeled as N1, N2, N3, ...*

1.6 References and Acknowledgments

2 Overall Description

2.1 Product Overview

AgeWise is a self-contained, web-based platform designed to assist senior citizens in navigating essential online services, managing daily activities, and enhancing digital literacy. Unlike conventional government service portals or healthcare apps that cater to specific needs, AgeWise integrates multiple functionalities into a single, easy-to-use platform. It provides one-click access to government services, medication and appointment reminders, and step-by-step tutorials for using social media, email, and video calling.

The platform is developed from the ground up, focusing on accessibility, simplicity, and inclusivity for elderly users who may have limited digital experience. It incorporates voice assistance, high-contrast UI, large fonts, and a guardian-supported account system to ensure a seamless experience. The system also notifies users about important government deadlines and allows caregivers or family members to assist in configuring reminders and notifications.

The AgeWise ecosystem includes multiple subsystems, such as a reminder management module, government services portal redirection, user education tutorials, and an AI-powered chatbot for assistance. The platform integrates with external APIs, including government databases for service redirection, authentication services for secure logins, and notification services for personalized reminders. The diagram below provides a high-level overview of how AgeWise interacts with its environment.

Context of Usage

- *A senior citizen logs into AgeWise using a simple, voice-assisted interface.*
- *They can navigate easily using large buttons and clear instructions.*
- *If they need to register for a government service (e.g., Jeevan Pramaan), they are directed to the respective portal with easy step-by-step guidance.*
- *The system can send reminders for medication or appointments, ensuring their well-being.*
- *If they need help, the AI chatbot or a family member can assist them in using online services.*
- *For secure transactions (e.g., payment of government fees), OAuth authentication and SSL encryption protect their data.*

2.2 Product Functionality

AgeWise is designed to be an intuitive, accessible, and all-in-one digital companion for senior citizens, enabling them to navigate online services effortlessly. The platform ensures a simple and structured user experience with accessibility-first features, helping seniors engage with technology without frustration.
Core Functionalities of AgeWise

User Registration & Secure Authentication
Simple Sign-Up & Login – Seniors can register using a minimal and easy-to-follow process, assisted by voice prompts and guided inputs.

University/Family Authentication (Optional) – Family members or caregivers can assist in registration and setup.
Multi-Factor Authentication (MFA) – Optional for added security (e.g., OTP via SMS/email).
Forgot Password & Account Recovery – Simple password reset using voice-based or one-click recovery options.

One-Click Access to Government Services
Direct Links to Government Portals – Users can access Jeevan Pramaan registration, pension verification, and welfare schemes through a single click without navigating complex websites.
Step-by-Step Guidance – Each service is accompanied by a simple tutorial explaining how to use it.
Automatic Deadline Notifications – The system sends alerts when government deadlines (e.g., pension renewals) are approaching.

Health & Medication Management
Medication Reminders – Users (or family members) can schedule medicine reminders, with notifications sent via voice alerts, SMS, or emails.
Doctor Appointment Scheduler – Users can set up appointment reminders for medical visits.
Health Check Notifications – Automatic reminders for health checkups based on predefined intervals (e.g., monthly, yearly).
Guardian Alerts – Family members receive updates if the senior citizen misses a reminder.

Learning & Digital Literacy Assistance
Step-by-Step Tutorials – Easy-to-follow guides on email, social media (Facebook, WhatsApp), video calls (Zoom, Google Meet), and safe browsing.
Interactive Help Desk – Seniors can ask questions via chatbot or voice assistant.
Offline Access to Guides – Tutorials can be downloaded for offline learning.

AI-Powered Chatbot & Voice Assistance
Text-to-Speech (TTS) Support – The system can read out website content, notifications, and instructions.
Voice Command Navigation – Seniors can use voice commands to access services (e.g., "Check my pension status").
AI Chatbot for Instant Help – The chatbot answers common questions, guides users through registration, and helps with troubleshooting.

Notifications & Alerts
Government Service Alerts – Automatic notifications about upcoming deadlines (e.g., pension renewal, ID verification).
Health & Medication Alerts – Personalized reminders for medicine and doctor visits.
Customizable Notification Preferences – Users can choose how they want to receive alerts (voice call, SMS, email, app notification).
Emergency Contact System – In case of missed important reminders, the system alerts family members or guardians.

Multi-Language & Accessibility Support
Multi-Language Interface – Users can switch between regional languages for easier understanding.
High-Contrast Mode – Dark mode and large font options improve readability for visually impaired users.
Screen Reader Compatibility – The system is fully compatible with screen readers for blind users.
Simple Navigation – The interface avoids clutter, uses large buttons, and has an easy-to-read layout.

Family & Guardian Assistance
Guardian-Managed Accounts – A family member or caregiver can assist with reminders, notifications, and service registrations.
Emergency Assistance – If a senior misses a critical reminder, the system alerts their guardian.
Activity Monitoring – Family members can track activity logs to ensure seniors are engaging with reminders and services.

Security & Privacy Protection
End-to-End Encryption – All personal data and interactions are secured using TLS 1.3 encryption.
Secure OAuth Authentication – Ensures only verified users can access services.
No Unnecessary Data Collection – The platform strictly follows GDPR and other privacy regulations to protect user data.

Admin & Content Management
User Management – Admins can approve, suspend, or assist user accounts as needed.
Content Moderation – Ensures all government service links, health reminders, and tutorials remain accurate and up to date.
System Analytics – Generates user engagement reports to continuously improve accessibility features.

2.3 Design and Implementation Constraints

The development of AgeWise is subject to several design and implementation constraints that must be considered to ensure accessibility, security, scalability, and integration with external services. These constraints affect the selection of technologies, design methodologies, security protocols, and overall system architecture.

- *Accessibility & Usability Constraints The system must comply with WCAG 2.1 (Web Content Accessibility Guidelines) to ensure senior citizens, including those with vision, hearing, or mobility impairments, can effectively use the platform. The UI must support high-contrast mode, large fonts, and screen-reader compatibility for visually impaired users. Voice navigation and text-to-speech functionalities must be integrated for users who may have difficulty reading text-based instructions.*
- *Hardware Limitations AgeWise must be optimized for low-end devices, including older smartphones and tablets that senior citizens might use. The system must operate efficiently on devices with at least 2GB RAM and 1GHz processor speed. Users may have slow or intermittent internet connections, requiring offline access for tutorials and reminders. The web application must be lightweight, with a maximum page load time of 3 seconds on a 3G network.*
- *Software & Technology Constraints Frontend Development: The UI will be built using React.js for a dynamic and responsive interface. Backend Development: The backend will use Node.js with Express.js to handle API requests efficiently. Database Selection: MongoDB will be used for storing user profiles, health reminders, and system logs due to its flexibility with unstructured data. PostgreSQL will be used for structured data requiring complex queries, such as service interactions. Hosting: The system must be hosted on a scalable cloud platform (AWS, Heroku, or Render) for high availability.*
- *Security & Compliance Constraints End-to-End Encryption: All sensitive user data must be encrypted using TLS 1.3 to prevent unauthorized access. Authentication: Secure login using OAuth 2.0 (Google, University Login, or Mobile OTP verification) is required. Data Privacy Regulations: The platform must comply with GDPR and Indian IT Act (for data protection and user privacy). User data storage and handling must follow minimal*

data collection principles, ensuring no personal data is shared without consent. Role-Based Access Control (RBAC): Admins must have restricted access to user information, and users should only see relevant features based on their roles (Senior Citizen, Family Guardian, Admin).

- *Integration with External Services Government API Constraints: The platform must integrate with government portals such as Jeevan Pramaan for pension verification. However, these APIs may have rate limits, authentication restrictions, or data format inconsistencies. Healthcare System APIs: The system must communicate with hospital/clinic scheduling services for appointment reminders. Notification Services: SMS and email notifications must be sent using third-party services (e.g., Twilio, Firebase Cloud Messaging), which may impose rate limits or regional restrictions.*
- *Design Methodologies & Standards: COMET Method for Software Design. AgeWise must be designed using the COMET (Collaborative Object Modeling and Enterprise Transformation) method, ensuring a modular, scalable, and maintainable software structure. This approach divides the system into separate functional modules (authentication, reminders, government services, AI chatbot, and notifications) for independent development and testing.*
- *All system interactions, including user authentication, service access, and reminders, must be modeled using UML diagrams. UML models to be used: Use Case Diagrams – Illustrate how users interact with AgeWise. Activity Diagrams – Represent the flow of user interactions. Class Diagrams – Define system components and their relationships.*
- *Performance & Scalability Constraints Concurrent Users: The system should support at least 500 concurrent users without performance degradation. Server Response Time: API calls must have a response time of less than 1 second under normal conditions. High-priority requests must be processed within 500ms. System Load Balancing: The backend must support auto-scaling based on traffic demand.*
- *Localization & Multi-Language Support AgeWise must support multiple languages, including English, Hindi, Telugu, Tamil, and Marathi, with an easy toggle option. Translation accuracy must be at least 95%, avoiding literal translations that could confuse senior users.*
- *Parallel Operations & Multi-User Support The system must support parallel operations, allowing a senior user and their family member (guardian) to access the account simultaneously. Notifications and updates should be real-time, ensuring reminders and government alerts are delivered instantly.*

2.4 Assumptions and Dependencies

2.4 Assumptions and Dependencies The development and functionality of AgeWise depend on several external and internal factors. These assumptions and dependencies may impact system performance, integration, and usability. If any of these assumptions change, modifications may be required in the system design.

It is assumed that users have basic familiarity with touch devices or can receive assistance from family members or caregivers. Voice assistance and AI chatbot features will help users with minimal digital literacy. Stable Internet Connection is Available

The platform assumes that users have access to a reliable internet connection for real-time government services, notifications, and updates. However, the system will include offline access

for tutorials and reminders in case of intermittent connectivity. Government Portals and APIs are Available & Functional

The system relies on government APIs to function correctly. It is assumed that these services remain operational and do not undergo frequent API structure changes. If government APIs experience downtime, users will be provided with alternative manual instructions. Users Willing to Provide Basic Personal Information

The platform assumes that senior citizens or their guardians are comfortable providing minimal personal details (e.g., name, date of birth, mobile number) for authentication and personalized services. AgeWise ensures that no sensitive data is shared without user consent and complies with GDPR and Indian IT Act regulations. Family Members or Guardians May Assist in Setup & Use

The system assumes that some senior users may need assistance from family members, caregivers, or university support teams for initial setup, authentication, and learning tutorials. Users Will Engage with Notifications & Reminders

It is assumed that users will check their reminders and notifications for medicine intake, government deadlines, and scheduled appointments. In case of missed reminders, the system will send secondary alerts to a designated guardian. AI Chatbot & Voice Assistance Will Enhance Usability

The platform assumes that integrating text-to-speech, chatbot assistance, and voice commands will significantly improve usability for senior citizens. Users should be able to understand simple voice instructions and chatbot responses. Dependencies Third-Party Authentication Services

The system depends on OAuth-based authentication (Google, University Login, or Mobile OTP verification) for secure access. If external authentication services experience downtime, backup login methods (manual email/password) will be available. Cloud Hosting & Server Availability

AgeWise is dependent on cloud hosting platforms (AWS, Heroku, Render) for deployment. The system must ensure 99.9% uptime to avoid accessibility issues. External Notification Services (SMS, Email, Voice Alerts)

The system relies on third-party notification providers (e.g., Twilio, Firebase Cloud Messaging, SendGrid) for sending reminders and alerts. In case of SMS/email service disruptions, the system will store pending notifications and retry later. Healthcare & Appointment Scheduling Systems

AgeWise integrates with hospital and clinic scheduling APIs for doctor appointment reminders. If healthcare systems do not provide an API, users will have to manually enter appointment details. Web Accessibility Standards Compliance (WCAG 2.1)

The system is designed according to Web Content Accessibility Guidelines (WCAG 2.1) to support users with disabilities. Any changes in accessibility standards may require updates in UI/UX design. Multi-Language Support & Translation Accuracy

AgeWise provides a multi-language interface, which depends on accurate translations. Inconsistent or incorrect translations could impact usability for non-English speakers. AI Chatbot & Speech Recognition Services

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

Designed to be simple, intuitive, and accessible for senior citizens. It features large buttons, high-contrast text, voice assistance, and minimal navigation steps to ensure ease of use.

Key UI Features: Main Dashboard: Provides quick access to government services, medication reminders, and tutorials.

One-Click Navigation: Users can access services with single-tap buttons instead of complex menus. Voice Assistance: Integrated text-to-speech and voice command support.

Notifications Panel: Displays upcoming reminders, alerts, and government deadlines. Multi-Language Support: Users can switch between different languages easily.

User Interaction: Touchscreen-Based Interface: Large, easy-to-press buttons for navigation. Simplified Menus: Clear, structured categories (Government Services, Health Reminders, Tutorials).

AI Chatbot Assistance: Users can ask questions and receive step-by-step guidance.

3.1.2 Hardware Interfaces

Interacts with various hardware devices to provide an accessible and seamless experience for senior citizens. The system is designed to be compatible with touchscreen devices, desktops, and assistive technologies.

Supported Hardware Devices: Smartphones & Tablets

Supports Android and iOS devices with touch input. Uses on-screen touch interactions, voice commands, and notifications.

Desktop & Laptop Computers accessible via web browsers (Chrome, Edge, Firefox, Safari). Supports keyboard shortcuts and screen readers. Voice Assistants & Smart Speakers (Future Integration)

Can integrate with Amazon Alexa or Google Assistant for voice-based navigation. Assistive Hardware

Screen Readers (NVDA, JAWS) for visually impaired users. Braille Displays for text-to-Braille conversion. Notification & Alert Devices

External SMS and Email Services for reminders. Wearable Devices (Smartwatches, Health Monitors) (Future Scope) for health reminders.

Logical Interaction User Input: Touchscreen taps, voice commands, keyboard/mouse interactions.

Data Processing: Cloud-based processing for reminders, notifications, and service interactions.

Output: Visual display on screen, voice feedback via text-to-speech, and alerts via SMS/email.

The system ensures compatibility across multiple devices to make AgeWise accessible and user-friendly for senior citizens.

3.1.3 Software Interfaces

Key Software Connections: Government Services APIs

Connects with Jeevan Pramaan, pension verification, and welfare schemes. Allows users to access government portals without manual navigation. Healthcare & Reminder System

Sends medication and appointment reminders via SMS, email, and push notifications. Integrates with hospital and clinic APIs (where available) to sync appointments. AI Chatbot & Voice Assistance

Uses AI-based chatbots (Dialog flow, IBM Watson, OpenAI) to assist users with navigation. Provides text-to-speech functionality for better accessibility. Authentication & User Management

Secure authentication ensures user verification. Role-based access control (RBAC) differentiates senior citizens, guardians, and admins. Notification & Alert System

Multi-Language Translation Services

Supports multiple languages using Google Translate API or custom localized text for better user accessibility. Cloud Hosting & Database Services

Hosted on AWS, Heroku, or Render for scalability and reliability.

Uses MongoDB for user and health-related data storage and PostgreSQL for structured data.

Logical Flow of Software Integration: User logs in → Authentication. User requests a government service → Redirected to the appropriate government API. User sets a health reminder → Stored in the database & synced with notification services. User asks for assistance → AI chatbot provides text or voice-based guidance. System sends notifications → Alerts are delivered via email, SMS, or push messages. By integrating these software components, AgeWise ensures a seamless, accessible, and secure experience for senior citizens.

3.2 Functional Requirements

User Registration & Authentication
F1: The system shall allow users to register and log in using OAuth authentication (Google, Mobile OTP, or University Login).
F2: The system shall provide an easy-to-use registration form with large fonts, voice assistance, and guided instructions.
F3: The system shall include a password recovery option with simple steps for resetting credentials.

Government Service Access

F4: The system shall provide one-click access to government portals (e.g., Jeevan Pramaan, pension schemes).
F5: The system shall include step-by-step guidance for using government services.
F6: The system shall send notifications for upcoming government deadlines, such as pension renewals.

Health & Medication Reminders
F7: The system shall allow users or guardians to set medication reminders with customizable time and frequency.
F8: The system shall send reminder notifications via SMS, email, and push alerts.
F9: Family members shall receive notifications if a senior user misses a critical health reminder.

AI Chatbot & Voice Assistance
F10: The system shall provide a chatbot assistant to guide users in navigating the platform.
F11: The system shall include text-to-speech functionality to read out content for visually impaired users.
F12: The system shall support voice commands to help users access services without typing.

Learning & Digital Literacy Assistance
F13: The system shall provide interactive tutorials on social media, email, video calling, and online safety.
F14: The system shall allow users to download tutorials for offline access.
Notifications & Alerts
F15: The system shall provide customizable notifications for government deadlines and health reminders.
F16: The system shall allow users to choose their preferred notification method (SMS, email, voice call).
F17: The system shall retry sending notifications if the first attempt fails.

Multi-Language & Accessibility Support
F18: The system shall support multiple languages and allow users to switch languages easily.
F19: The system shall include high-contrast mode and screen reader support for visually impaired users.

Security & Data Protection
F20: The system shall encrypt all user data using TLS 1.3 encryption.
F21: The system shall enforce role-based access control (RBAC) to protect sensitive information.
F22: The system shall comply with GDPR and Indian IT Act for data privacy.

Admin & Content Management
F23: The system shall allow admins to approve, suspend, or remove user accounts if necessary.
F24: The system shall allow admins to update government service links and tutorial content.
F25: The system shall generate user engagement reports for performance analysis.

3.3 Use Case Model

3.3.1 Use Case #1: Set and Receive Reminders (U1)

Author: [Your Name]
Purpose:
Allow a senior citizen or guardian to set reminders for medications, doctor appointments, or government deadlines. The system sends notifications via SMS, email, or app alerts.
Actors:

Senior Citizen (Primary actor)
Guardian (Secondary actor)
Notification System (Automated system)
Preconditions:
The user must be logged in.
Notification preferences (SMS, email, app) must be configured.
Postconditions:
The reminder is stored in the system.
A notification is sent at the scheduled time.
Flow of Events:
Basic Flow:
The user navigates to the "Reminders" section.
The user selects "Set a New Reminder" and enters reminder details.
The system stores the reminder in the database.
The notification system sends an alert at the scheduled time.
Alternative Flow:
If the senior citizen does not configure notifications, the system suggests a default method (e.g., SMS).
If the guardian is setting a reminder, they select the senior citizen's profile before entering details.
Exceptions:
No Internet Connection → The system saves the reminder locally and syncs later.
Invalid Time Entry → The system prompts for correction.
Priority: High
Includes: User Authentication (U2)

3.3.2 Use Case #2: Access Government Services (U2)

Author: [Your Name]
Purpose:
Enable senior citizens to access and apply for government services like Jeevan Pramaan via a guided interface.
Actors:
Senior Citizen (Primary actor)
Government Services API (External system)
Preconditions:
The user must be logged in.
The system must be able to connect to government APIs.
Postconditions:
The user is redirected to the correct government portal.
Flow of Events:
Basic Flow:
The user selects "Government Services" from the main menu.
The system displays a list of services (e.g., pension verification, ID renewal).
The user selects a service.
The system redirects the user to the official government site with instructions.
Alternative Flow:
If government APIs are unavailable, the system shows manual instructions instead.
Exceptions:
Invalid User Authentication → The user is prompted to log in again.
Government Portal Unavailable → The system provides alternative contact information.
Priority: High
Includes: Secure Authentication (U3)

3.3.3 Use Case #3: Watch Tutorials (U3)

Author: [Your Name]

Purpose:
Help senior citizens learn digital skills through video and text-based tutorials.
Actors:
Senior Citizen (Primary actor)
Tutorial System (Automated system)
Preconditions:
The user must be logged in.
The system must have preloaded tutorials.
Postconditions:
The user successfully watches or reads a tutorial.
Flow of Events:
Basic Flow:
The user selects "Digital Learning" from the menu.
The system displays a list of available tutorials.
The user selects a tutorial (e.g., "How to use WhatsApp").
The system plays the video or displays the step-by-step guide.
Alternative Flow:
If a tutorial is unavailable, the system suggests an alternative topic.
Exceptions:
No Internet Connection → The system prompts the user to download tutorials for offline access.
Priority: *Medium*
Includes: *AI Chatbot Support (U4)*

3.3.4 Use Case #4: AI Chatbot Assistance (U4)

Author: *[Your Name]*
Purpose:
Provide instant assistance to senior citizens using an AI-powered chatbot.
Actors:
Senior Citizen (Primary actor)
AI Chatbot (Automated system)
Preconditions:
The user must be logged in.
Postconditions:
The chatbot provides helpful responses.
Flow of Events:
Basic Flow:
The user clicks on the "Help" button.
The chatbot asks what assistance is needed.
The user types or speaks a question (e.g., "How do I set a reminder?").
The chatbot provides step-by-step guidance.
Alternative Flow:
If the chatbot cannot provide an answer, it redirects the user to human support.
Exceptions:
Voice Input Not Recognized → The chatbot prompts the user to type instead.
Priority: *High*

4 Other Non-functional Requirements

4.1 Performance Requirements

P1: The system shall load the main dashboard within 3 seconds on a stable internet connection (5 Mbps).
P2: The system shall handle 500 concurrent users without degrading response time beyond 2 seconds per request.
P3: The system shall ensure that database queries (MongoDB/PostgreSQL) execute within 200ms for optimal performance.
P4: Notifications and reminders (SMS, email, push) shall be delivered within 5 seconds of the scheduled time.
P5: The AI Chatbot shall respond to user queries within 1 second for text and 2 seconds for voice.
P6: The system shall support at least 99.9% uptime, with downtime limited to scheduled maintenance.
P7: Voice commands should be processed and responded to within 1.5 seconds for real-time assistance.
By optimizing backend processing, caching mechanisms, and load balancing, AgeWise ensures a smooth and responsive experience for all users.

4.2 Safety and Security Requirements

The AgeWise system handles sensitive user data, including personal information, health reminders, and authentication credentials. The following security measures ensure data privacy, protection against cyber threats, and system integrity.

Authentication & Access Control
S1: The system shall implement OAuth 2.0 authentication (Google, OTP verification) for secure user access.
S2: The system shall enforce Role-Based Access Control (RBAC) to prevent unauthorized data access.
S3: All passwords shall be stored using crypt hashing with a minimum of 12 salt rounds.
Data Privacy & Encryption
S4: All personal data shall be encrypted using AES-256 encryption in storage and TLS 1.3 for transmission.
S5: The system shall comply with GDPR and the Indian IT Act to protect user privacy.
S6: User data shall be deleted permanently upon account deactivation as per compliance regulations.
Fraud Prevention & Secure Transactions
S7: Any payment transactions (if applicable) shall be processed through PCI-DSS compliant payment gateways.
S8: The system shall detect and block suspicious activity such as multiple failed login attempts (account lockout after 5 failed tries).
S9: System logs shall record all access attempts and actions, maintaining a 90-day audit history for security reviews.
System Resilience & Disaster Recovery
S10: The system shall automatically back up user data every 12 hours to prevent data loss.
S11: A failover mechanism shall redirect users to a backup server in case of a primary server failure.
S12: If a data breach is detected, administrators shall receive immediate security alerts, and affected users shall be notified within 24 hours.
By enforcing multi-layered security policies, AgeWise ensures safe digital interactions for senior citizens.

4.3 Software Quality Attributes

4.3.1 Reliability

The system must ensure 99.9% uptime, ensuring that senior citizens always have access to critical services (pension verification, health reminders).
Automated error detection and recovery mechanisms will monitor the system for unexpected failures and restart failed processes automatically.
Redundant database backups and failover servers will prevent data loss in case of hardware failures.

4.3.2 Usability
The UI must support WCAG 2.1 accessibility standards, ensuring high contrast, large fonts, and screen reader compatibility.
The system shall provide voice-guided navigation and AI chatbot assistance to help non-tech-savvy users.
Tutorials must be simple, step-by-step, and available in multiple languages to enhance user learning.

4.3.3 Maintainability
The system shall follow a modular architecture using the COMET design method, making it easy to update and extend features.
Automated logging and error reporting will help developers quickly identify and fix issues.
Code shall be well-documented to ensure future teams can modify and maintain the system efficiently.

4.3.4 Scalability
The system must support a growing number of users (500+ concurrent users at launch, expandable to 5000 users over time).
Load balancing techniques shall distribute traffic across multiple servers to prevent overload.
Database sharding and indexing shall optimize performance as data volume increases.

5 Other Requirements

*<This section is **Optional**. Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

Appendix A – Data Dictionary

Variable Name	*Type*	*Description*	*Possible Values*	*Related Operations*	*Functional Requirement*
User ID	*String*	*Unique identifier for each user*	*Auto-generated UID*	*Create, Retrieve, Update, Delete*	*F1, F10*
user Role	*String*	*Defines user role (Senior Citizen, Guardian, Admin)*	*"SeniorCitizen", "Guardian", "Admin"*	*Role-based Access Control (RBAC)*	*F22*
language Preference	*String*	*Stores user's preferred language*	`"English"`, `"Hindi"`, `"Telugu"`, etc.	*Multi-language Support*	*F19*
govServiceID	*String*	*Unique identifier for government services*	Auto-generated	*Retrieve, Redirect*	*F4, F5*
reminderID	*String*	*Unique ID for health reminders*	Auto-generated	*Create, Update, Notify*	*F7, F8*
reminderStatus	*String*	*Status of a health reminder*	`"Pending"`, `"Completed"`, `"Missed"`	*Update status, Notify guardian*	*F9*
notificationType	*String*	*Mode of sending notifications*	`"SMS"`, `"Email"`, `"Push"`	*Notification Dispatch*	*F16, F17*
aiChatbotQuery	*String*	*Stores user queries for AI chatbot*	Free text	*Process, Respond*	*F10, F11*
authMethod	*String*	*Authentication type used*	`"OAuth"`, `"OTP"`, `"Password"`	*User Authentication*	*S1, S2*
timestamp	*DateTime*	*Records the date and time of an action*	*YYYY-MM-DD HH:MM: SS*	*Logging, Audit Trail*	*S9, S10*

Appendix B - Group Log