SOFTWARE ENGINEERING

CONTINUOUS ASSESSMENT

REPORT-

D15A: GROUP_5

OUR TEAM:

20.Madhura Jangale

28. Rujuta Medhi

38. Shravani Patil

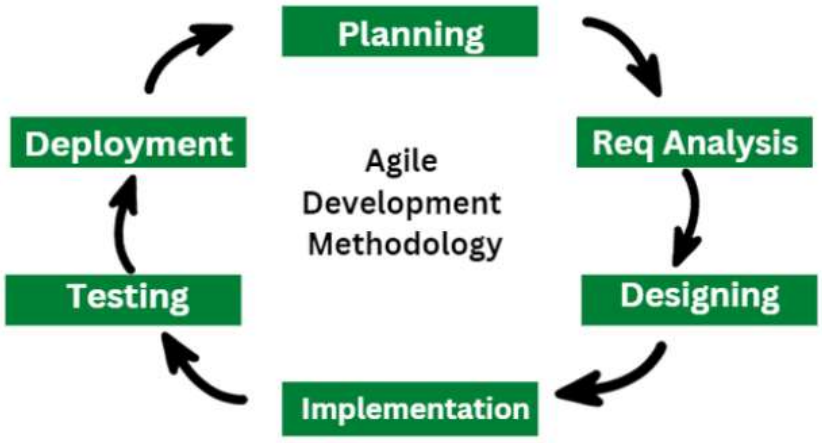
66.Vedang Wajge

Introduction:

E-yojana is a project designed to help people stay informed about upcoming government schemes and allow them to apply directly from the website. For this project, we have chosen the Agile SDLC model due to its flexibility, iterative nature, collaboration, and customer feedback. This report outlines the reasons for choosing the Agile model and details the activities planned for each phase of the project.

Why Agile?

1. **Flexibility**: Agile allows us to adapt to changes in requirements and feedback from users, which is crucial for a project like E-yojana where user needs might evolve.
2. **Iterative Development**: We can deliver incremental updates, ensuring that core functionalities are available early and the product can be improved over time.



An outline of the sprints for our E-Yojana project under the **Scrum Model,** aligning with the Agile methodology.
Each sprint focuses on delivering a working increment of the platform and involves regular customer feedback and iterative improvements.

Sprint 1: Project Initialization & Planning
Duration: 2 weeks
Goals:
Finalize project scope and objectives.
Set up the development environment (frontend, backend, and database).
Choose tech stack (React,Node,Expess + PostgreSQL).
Draft user stories and create the product backlog.
Initial wireframes and mockups for the website.
Deliverables:
Product backlog with prioritized features.
Initial UI designs for key sections (e.g., Home, Login, Categories).

Sprint 2: Frontend and Backend Setup
Duration: 2 weeks
Goals:
Create a basic layout and structure for the website.
Set up the backend server with PostgreSQL.
Implement the user authentication system (Login, Sign Up).
Basic design of the homepage, navbar, and category page.
Deliverables:
Functional homepage, login, sign-up, and navbar.
Backend setup and database connectivity.

Sprint 3: Core Features Development - Categories & Schemes
Duration: 2 weeks
Goals:
Develop the Category page to display the list of government schemes.
Implement the feature to allow users to apply for schemes.
Create filter mechanism (age category, state).
Implement the backend API for scheme data handling.
Deliverables:
Fully functional Category page.
Working filters and apply functionality.
Backend APIs for scheme retrieval and application submission.

Sprint 4: Admin Portal & Document Verification
Duration: 2 weeks
Goals:
Develop the admin portal for managing scheme applications.
Implement document verification process (status: verified, pending, rejected).

Notification system to inform users of their application status.
Admin login and authentication.
Deliverables:
Admin portal with document verification.
Notification feature for application status.

Sprint 5: Notifications & User Dashboard
Duration: 2 weeks
Goals:
Build the user dashboard to view submitted applications and status updates.
Implement an email notification system for scheme updates and application status.
Refine the user interface based on feedback.
Deliverables:
Functional user dashboard.
Email notification feature.
Enhanced UI for better user experience.

Sprint 6: Testing & Feedback
Duration: 2 weeks
Goals:
Conduct end-to-end testing (frontend and backend).
Collect user and admin feedback for improvements.
Perform security checks and validation for user inputs.
Fix any bugs or issues found during testing.
Deliverables:
Bug-free, tested platform.
Implemented feedback for minor improvements.
Completed security validation.

Sprint 7: Deployment & Final Review
Duration: 1 week
Goals:
Deploy the platform on a live server.
Final review of all features and performance.
Prepare project documentation (user manuals, developer notes).
Deliver the project to stakeholders.
Deliverables:
Live deployment of the E-Yojana platform.
Complete project documentation.
Final stakeholder approval.
These sprints provide a structured yet flexible framework for delivering a functional and user-centric E-Yojana platform within the Agile Scrum process.

Conclusion:

By adopting the Agile SDLC model for E-yojana, we aim to create a flexible and user-centric platform that effectively informs users about government schemes and allows them to apply seamlessly.

Software Requirements Specification for

E-yojana

Version 1.0 approved

Prepared by:

Shravani Patil , Rujuta Medhi , Madhura Jangale , Vedang Wajge

21, August 2024

Table of Contents

1. Introduction

1.1 Purpose

The purpose of E-yojana is to streamline the application process by enabling users to submit their forms and documents online for government provided schemes. By providing a centralized and efficient system, we aim to reduce the time and effort required for both applicants and administrators. This ensures that applications are processed quickly, accurately, and securely, enhancing the overall experience for all parties involved.

1.2 Document Conventions

The SRS document follows a hierarchical structure with clear headings and subheadings for different sections.

Each section is numbered (e.g., 1.0, 1.1, 1.1.1) for easy reference.

2. Font and Formatting:

Font Style: The standard font used throughout the document is **Times New Roman**.

Font Size:

Headings: 19pt.

Subheadings: 14pt.

Body Text: 12 pt, regular.

Bold: Represents headings or important terms.

Italics: Used for emphasis on key terms or phrases.

1.3 Intended Audience and Reading Suggestions

This document is intended for the following audiences:

Developers:

Purpose: Understand functional and non-functional requirements for effective system design and implementation.

Focus: Functional Requirements, Detailed Use Cases, System Architecture, Non-functional Requirements.

Project Managers:

Purpose: Manage project scope, requirements, and constraints to ensure timely and budget-friendly delivery.

Focus: Introduction, Overall Description, Prioritization of Requirements, Risk Management.

Testers:

Purpose: Ensure the system meets all requirements through testing.

Focus: Functional Requirements, Detailed Use Cases, Non-functional Requirements, Acceptance Criteria.

1.4 Product Scope

Online Application Submission:

Users can submit application forms and supporting documents electronically for government schemes.

Centralized System:

Provides a unified platform for managing applications, reducing manual handling and paperwork.

Processing and Verification:

Streamlines the verification and processing tasks for administrators, ensuring quicker and more accurate handling.

User Experience:

Enhances user experience through a user-friendly interface, making it easier to track and manage applications.

1.5 References

https://www.india.gov.in/my-government/schemes-0

https://ieeexplore.ieee.org/document/10085527

2. Overall Description

2.1 Product Perspective

E-yojana is a user-friendly app designed to list ongoing government and private schemes, allowing users to apply directly by uploading required documents. The app features a document verification system that checks submissions for errors, providing feedback to users for corrections if needed. Once documents are verified as accurate, the app automatically fills out the original application form, ensuring a smooth and efficient process for users to access and benefit from various schemes.

2.2 Product Functions

The *E-yojana* application must perform the following major functions:

- **Scheme Listing**: Display a comprehensive list of all ongoing schemes, with details and eligibility criteria.
- **Application Submission**: Allow users to apply for schemes by filling out forms and uploading required documents.
- **Document Verification**: Check uploaded documents for errors or missing information and provide feedback to users.
- **Feedback and Correction**: Notify users of any issues with their submissions and allow them to resubmit corrected documents.
- **User Notifications**: Send alerts and updates to users regarding the status of their applications and any required actions.
- **Admin Dashboard**: Provide administrators with tools to manage schemes, review submissions, and monitor the verification process.

2.3 User Classes and Characteristics

Applicants (End Users):

Characteristics: General public, including individuals or families looking to apply for government or private schemes.

System Administrators:

Characteristics: IT professionals responsible for maintaining the app, managing user accounts, and ensuring system security and performance.

Scheme Administrators:

Characteristics: Government or private sector employees responsible for managing schemes and verifying documents.

2.4 Operating Environment

Hardware Platform:

- **Client Devices**: Accessible on smartphones, tablets, and desktop computers.
- **Server Infrastructure**: Hosted on scalable cloud servers with high availability and redundancy.

Operating System: Compatible with Android (7.0+), iOS (12+), Windows (10+), macOS (10.14+), and Linux.

Seamless integration with existing applications and systems like email clients, web browsers, and government portals.

2.5 Design and Implementation Constraints

Security Compliance:The app must adhere to strict security protocols, including data encryption, secure authentication, and compliance with regulations like GDPR, to protect user data and documents.

Scalability:The app should be designed to scale efficiently as the number of users and the volume of document processing increases. This requires careful planning of the server infrastructure and database management.

2.6 User Documentation

The following user documentation will be provided:

- **User Guide:**

Apply for schemes, submit documents, and track your application status via the app.

- **Admin Guide**

Manage schemes, verify documents, and oversee user applications through the admin dashboard.

2.7 Assumptions and Dependencies

Assumptions:

- Users have access to compatible devices (smartphones, tablets, desktops) and internet connectivity.
- Users can navigate the app with basic digital literacy.

Dependencies:

- **Compliance Regulations:** Adherence to data protection laws (e.g., GDPR) and industry standards.
- **System Integration:** Compatibility with existing systems and databases for scheme information and updates.

3. External Interface Requirements

3.1 User Interfaces

The platform will feature a responsive web interface, ensuring accessibility across all devices, including desktops, tablets, and smartphones. The interface will be designed for ease of use, with a consistent layout and navigation.

UI Navigation flow

Login/Register → Home Page: Users land on the home page after logging in or registering, where they can explore communities or access profile settings.

Home Page → Explore Communities: Users browse communities, filter by tags, or use the search bar to find specific communities.

Home Page → Create Community: Logged-in users can create a new community via an easy-to-use form.

Home Page → Profile: Users access their profile to view their communities and manage account settings.

Profile → Admin Dashboard: If the user is an administrator, they can navigate to the admin dashboard from their profile to manage community-related tasks.

3.2 Hardware Interfaces

No specific hardware interfaces are required. The platform will be accessible from any device with internet connectivity.

3.3 Software Interfaces

- Database:PostgreSQL (stores user profiles, documents uploaded).
- Operating System: Windows 11
- Backend:Node.js 18 with Express.js (handles server-side logic and API requests).
- Frontend: React 18 (manages UI).

- Email Notifications: Integration with email services for sending application updates and feedback.

3.4 Communications Interfaces

- Secure Communication: All communication between the client and server will be encrypted using HTTPS.
- Notifications: The platform will use email and push notifications to inform users about the status of the application that they have applied for.

4. System Features

4.1 System Feature 1: Accurate and Updated Information on Government Schemes

Description: Eyojana allows users to access the latest government schemes by specifying their age, state, caste, category, and location. Users can apply directly for these schemes through the platform and receive feedback from admins for any necessary corrections to their applications.

Functional Requirements:

- Users must be able to create and edit their profile details relevant to scheme eligibility.
- Admins can invite users to specific schemes and assign appropriate roles or categories.
- Privacy settings related to personal information should be configurable by the user.

4.2 System Feature 2: Filtered Schemes According to User Needs

Description: Eyojana provides a feature that filters and lists government schemes based on users' specific needs, such as age, state, caste, category, and location. This ensures that users are only shown schemes for which they are eligible, making the application process more efficient.

Functional Requirements:

- Users must be able to filter schemes based on their personal information and preferences.
- The system should dynamically update the list of schemes as users refine their filter criteria.
- Users should be able to save filtered searches for quick access in the future.

4.3 System Feature 3: Errors and Feedback from Admins

Description: Eyojana enables users to receive feedback from admins regarding any errors or issues with their scheme applications. This feedback helps users correct mistakes and improve their chances of successfully applying for government schemes.

Functional Requirements:

- Admins must be able to review applications and provide detailed feedback on errors or issues.
- Users should be notified promptly of any feedback or required corrections.
- The system should allow users to resubmit applications after addressing the feedback.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- E-yojana must handle at least 10,000 concurrent users without slowing down.
- Page Load Time: Average page load time should be under 3 seconds during normal usage.

5.2 Safety Requirements

- Data Protection: Implement robust data protection to prevent unauthorized access and breaches.
- Backups: Perform regular backups to maintain data integrity and ensure availability.

5.3 Security Requirements

- Encryption: Encrypt all sensitive user data in transit and at rest to protect against unauthorized access.
- Authentication: Use multi-factor authentication to enhance security for user accounts.
- Access Control: Implement role-based access control to safeguard sensitive features and data.

5.4 Software Quality Attributes

Usability: Aim for a user satisfaction rate of at least 85% by providing an intuitive experience.

- Reliability: Maintain a system uptime of 99.9% to ensure constant availability.

- Maintainability: Resolve issues within an average of 24 hours to ensure smooth operation.

5.5 Business Rules

- Regular Users: Create and manage profiles, access scheme information, and submit applications efficiently.
- Community Admins: Oversee community settings, manage events, and facilitate user interactions.
- Platform Admins: Supervise platform operations, handle application management, and ensure data moderation.