Samwise Service App

**Project Plan**

# **Introduction**

The project plan for the Samwise Service App provides a comprehensive roadmap for the development of our web-based service platform. It outlines the key components, practices, and measurements that will guide the project to success. The plan encompasses the project organization, practices, and metrics used, along with milestones and objectives for each iteration. The project is built on an iterative development approach, emphasizing continuous improvement and the delivery of incremental updates. This plan also addresses the deployment strategy, support and maintenance, security measures, and user communication. Finally, it addresses lessons learned, which will be gathered during retrospectives at the end of each iteration, ensuring the project's adaptability and continuous enhancement.

# **Project organization**

The project team consists of four members: Elif, Betül, Annie and Özde. Annie will work as Project Manager. Özde will work as a Business Analyst. Elif will work as Software Developer and Betül will work as Software Architect and Tester.

# **Project practices and measurements**

The project will use an iterative development approach which permits incremental updates and improvements. In order to track progress, we will have iteration assessments to provide regular reviews and retrospectives at the end of each iteration. Additionally, we will track progress by collecting the velocity per iteration (completed work item points/iteration). Continuous integration will also be implemented to ensure that changes to code are integrated and tested regularly. The metrics that will track continuous integration includes code coverage, build success/failure rates, and test pass rates. Independent Testing will be employed to ensure software quality and will be measured using defect reports and test case pass rates. Regarding source code version control, GitHub will be utilized to manage source code changes and tracked using commit history and branching statistics.

# **Project milestones and objectives**

| **Iteration** | **Primary objectives** | **Scheduled start or milestone** | **Target velocity** |
| --- | --- | --- | --- |
| I1 | Phase 1: Project Initiation and Planning   1. Establish project team and roles 2. Define project scope, objectives, success criteria 3. Develop a detailed project plan and timeline 4. Define necessary resources and budget | 3.10.2023 - 16.10.2023 | 30 |
| I2 | Phase 2: System Design and Architecture   1. Create the system architecture and database schema 2. Define the user interface (UI) design and user experience (UX) principles 3. Choose the technology stack, programming languages, and development tools | 17.10.2023 - 3.11.2023 | 32 |
| l3 | Phase 3: Development Phase 1   1. Develop additional features, including payment processing, chatbot support, and search functionalities 2. Implement support for premium subscriptions and payment features 3. Conduct regular code reviews and quality assurance testing | 4.11.2023-3.01.2024 | 25 |
| l4 | Phase 4: Development Phase 2   1. Continue development of additional features focusing on user interactions and service improvements 2. Implement features related to service request history and user accounts 3. Ensure the application is user-friendly and performant on browsers | 4.01.2024-11.02.2024 | 20 |
| l5 | Phase 5: Testing and Quality Assurance   1. Conduct extensive testing, including functional usability, security, and performance testing 2. Identify and address any bugs and issues 3. Develop and test backup and recovery procedures | 12.02.2024-27.02.2024 | 55 |
| l6 | Phase 6: User Acceptance Testing, Feedback and Iterations   1. Invite a group of beta testers to evaluate the app 2. Gather feedback and make necessary improvements based on user input | 28.02.2024-01.04.2024 | 12 |
| l7 | Phase 7: Launch and Initial User Onboarding   1. Officially launch the Samwise Service App 2. Onboard initial users and service providers | 02.04.2024-3.10.2024 | 12 |

# **Deployment**

The Samwise Service App will be deployed as an internet-based application accessible through standard web browsers. The deployment strategy is designed to ensure compatibility, customization, scalability, and long-term support for a seamless user experience.

**Continuous Deployment:**

Continuous Integration and Deployment (CI/CD): CI/CD practices will be employed to automate the build, test, and deployment processes, ensuring rapid and error-free deployment of new features and updates.

**Compatibility and Customization:**

The application will support major web browsers (such as Mozilla Firefox, Google Chrome, Safari, and Microsoft Edge).

Regular compatibility testing and updates will be conducted to ensure optimal performance across supported browsers and devices.

Users and administrators will have the ability to configure and customize the app based on their preferences, enabling a personalized user experience.

**Support and Maintenance:**

The Samwise Service App will have a defined level of support and maintenance.

Regular updates, including feature enhancements and bug fixes, will be released to improve the app's functionality and user experience.

Robust security precautions and data protection mechanisms will be in place to safeguard user data.

The system will commit to providing support for a minimum of three years from the initial release.

Support response times for critical issues will be within 24 hours to ensure prompt resolution of urgent user concerns.

**Security Measures:**

Robust security protocols, including encryption, secure socket layer (SSL) certificates, and firewalls, will be implemented to protect user data and application integrity.

Regular security audits and vulnerability assessments will be conducted to identify and mitigate potential security risks, ensuring a secure environment for user data and maintaining user trust.

**Scalability and Performance Optimization:**

The system will be designed to scale horizontally to accommodate increased demand.

A scalability requirement mandates the ability to handle a 50% increase in user data within six months without performance degradation.

Performance optimization measures will be implemented to maintain responsive user interactions and efficient data processing.

**User Communication**

Clear communication channels will be established to inform users and administrators about system updates, maintenance schedules, and new features.

User guides and notifications within the app will provide assistance and information to users regarding customization options and app functionalities.

# **Lessons learned**

At the end of each iteration, retrospectives and regular reviews will be performed to evaluate project progress, address issues, and identify improvement strategies. We will track progress by collecting the velocity per iteration (completed work item points/iteration). The lessons learned from the retrospective will be recorded in this section as the iterations progress. The topics for lessons learned may include evaluating the build and deployment times, availability of reliable test data, communication among the team, sprint planning efficiency, availability of documentation, and clarity of the scope for each iteration.