Samwise Service App

Project Plan

1. 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.

2. Project Organization

2.1. Project Team Roles and Responsibilities

The project team consists of four members: Elif, Betül, Annie and Özde. Here are the roles and responsibilities of the team members:

Annie (Project Manager):
Responsibilities:

- Overall project planning, execution, and delivery within the defined scope, timeline, and budget.
- Communication and collaboration with stakeholders, ensuring their requirements are understood and addressed.
- Risk management and mitigation planning.
- Team coordination and task assignment, ensuring each team member's workload aligns with project goals.
- Monitoring project progress.

Özde (Business Analyst):
Responsibilities:

- Elicitation and documentation of business requirements from stakeholders.
- Analyzing and documenting current business processes and proposing improvements.
- Collaboration with stakeholders to ensure requirements align with business goals.
- Bridging communication gaps between technical and non-technical team members.
- Preparation of use cases, user stories, and other documentation for the development team.

Elif (Software Developer):
Responsibilities:

- Writing high-quality, efficient, and maintainable code.
- Collaborating with the software architect to understand the system architecture and design.
- Participating in code reviews and providing constructive feedback.
- Debugging and resolving software defects.
- Adhering to coding standards and contributing to the development of best practices.

Betül (Software Architect and Tester):
Responsibilities:

- Designing the overall software architecture and ensuring it aligns with project requirements.
- Collaborating with the development team to guide the implementation of the software architecture.
- Defining and implementing testing strategies, including unit testing and integration testing.
- Identifying and addressing software design issues.
- Ensuring the software is thoroughly tested and meets quality standards before release.

2.2. Communication Channels

- Whatsapp for communication

- Google Meet for meetings
- GitHub for sharing the documentation and codes

3. Project Practices and Measurements

The project will use an iterative agile 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 include 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.

4. Project Milestones and Objectives

4.1. Project Milestones and Objectives for Release 1

Iteration	Primary Objectives (risks and use case scenarios)	Scheduled Start or Milestone	Target Velocity (Story Points)
I1	**Objective:** Project Initiation and Planning 1. Project Initiations 2. Define Project Plan, Risk List and Work Item List Documents 3. Define Use Cases and their Scenarios 4. System-wide Requirements Specification Documents **Risks:** 1. Mitigate Risk 1	23.10.2023 - 4.11.2023	38
I2	**Objective:** Technical Research and Implementation of Use Case 5 - 9 1. System Design and Architecture, UI and Database for Use Case 5 - 9 2. Implement Use-Case 5: Manage User Account 3. Test Use-Case 5: Manage User Account 4. Implement 9: Manage Service 5. Test Use Case 9: Manage Service **Risks:** 1. Mitigate Risk 1 2. Mitigate Risk 6 3. Mitigate Risk 8 4. Mitigate Risk 7	6.11.2023 - 25.11.2023	53
I3	**Objective:** Implementation of Use Case 1-2 1. System Design and Architecture, UI and Database for Use Case 1 - 2 2. Implement Use-Case 2: Search Service 3. Test Use-Case 2: Search Service 4. Implement Use-Case 1: Schedule Appointment 5. Test Use-Case 1: Schedule Appointment **Risks:** 1. Mitigate Risk 1 2. Mitigate Risk 6	28.11.2023-15.12.2023	32

I4	**Objective:** Implementation of Use Case 4-6 1. System Design and Architecture, UI and Database for Use Case 4 - 6 2. Implement Use-Case 4: Manage Time Slot 3. Test Use-Case 4: Manage Time Slot 4. Implement Use-Case 6: Ask Help 5. Test Use-Case 6: Ask Help **Risks:** 1. Mitigate Risk 1 2. Mitigate Risk 6	16.12.2023-02.01.2024	34

4.2. Project Milestones and Objectives for Release 2

Iteration	Primary Objectives (risks and use case scenarios)	Scheduled Start or Milestone	Target Velocity (Story Points)
I1	**Objective:** Implementation of Use Case 8 1. System Design and Architecture, UI and Database for Use Case 8 2. Implement Use-Case 8: Make Proposal 3. Test Use-Case 8: Make Proposal **Risks:** 1. Mitigate Risk 1	3.01.2024-20.01.2024	32
I2	**Objective:** Implementation of Use Case 10 1. System Design and Architecture, UI and Database for Use Case 10 2. Implement Use-Case 10: Receive Payment 3. Test Use-Case 10: Receive Payment **Risks:** 1. Mitigate Risk 1 2. Mitigate Risk 6 3. Mitigate Risk 10	21.01.2024-15.02.2024	34
I3	**Objective:** Implementation of Use Case 3 1. System Design and Architecture, UI and Database for Use Case 3 2. Implement Use-Case 3: Make Payment 3. Test Use-Case 3: Make Payment **Risks:** 1. Mitigate Risk 1 2. Mitigate Risk 6 3. Mitigate Risk 10	16.02.2024-29.02.2024	20
I4	**Objective:** Implementation of Use Case X.X - 7 1. System Design and Architecture, UI and Database for Use Case 7 2. Implement Use-Case 7: Write Review 3. Test Use-Case 7: Write Review **Risks:** 1. Mitigate Risk 1 2. Mitigate Risk 6	1.03.2024-17.03.2024	32

5. Deployment

Our deployment strategy for this project adopts an agile and iterative methodology, allowing for flexibility and responsiveness to evolving requirements. The development stack comprises JavaScript and Node.js, providing a robust foundation for building dynamic and scalable applications. In the initial release, we will create a standalone application utilizing MySQL for database object definition and creation, ensuring data integrity and efficiency. This standalone version will undergo continuous development and testing, aligned with agile principles, to swiftly deliver a functional product. For continuous integration and development, GitHub will serve as the primary platform for updating and managing the source codes.

In the subsequent release, our focus will shift towards cloud deployment. Leveraging cloud services, we will enhance the application's availability and scalability while minimizing initial investment. The chosen cloud service providers will align with the frameworks employed, and we will adopt a microservices architecture, allowing for independent development and deployment of application components. This phased approach, transitioning from a standalone application to a cloud-hosted solution, embodies our commitment to delivering a dynamic and adaptable software product, meeting the evolving needs of our users.

6. 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.