ReserveWell
Project Plan

Revision Table

Revision	Description	Date	Person(s)
1.0	The first version of the project plan is released.	04/11/2023	Umut GÜLER
1.1	Revisions are made from the other teams' and the instructor's feedback. In the "Project milestones and objectives part", the scoring aspect of the tasks are stated in more detail.	11/11/2023	Umut GÜLER, Ezgi Tunalı
1.2	Revisions are made according to the instructor's feedback. In the "Project milestones and objectives" part, the iterations are added, explained and linked to the iterations of the Unified Process (UP) more clearly. The system architecture is linked to the elaboration iteration more precisely.	25/11/2023	Umut GÜLER, Mustafa ZEMİN
1.3	Revisions are made according to our checks. The minor mistakes in the dates in the "Project milestones and objectives" are corrected. The "deployment" part is revised according to the decided deployment platform. The "lessons learned" part is updated according to the lessons learned from iterations 1 and 2.	30/11/2023	Umut GÜLER, Mustafa ZEMİN, Ezgi ERKAŞ
1.4	Revisions are made to part 6, lessons learned, after our acknowledgments to some of our mistakes and solutions in iteration 3.	23/12/2023	Umut GÜLER, Ezgi ERKAŞ

1 Introduction

This project plan serves as a comprehensive guide for the development and management of the Restaurant Reservation System. The application, ReserveWell, is basically an application that is designed to simplify the process of making and managing reservations at restaurants. It will initially be web-based, and the mobile application of it is to be made. The project plan explains the project's organizational structure, management practices, technical methodologies, measurement strategies, milestones, deployment approach, and strategies for continuous improvement. It provides a clear roadmap for the successful execution of the project and ensures that ReserveWell meets its objectives effectively and efficiently.

2 Project organization

The success of ReserveWell heavily depends on a well-structured project organization which consists of dedicated team members with defined roles and responsibilities. The project team is composed of the following key members:

Project Manager:

Role: The Project Manager, Ezgi Tunalı, is responsible for overseeing the project as a whole, ensuring that it is compliant with the defined scope, timeline, and resources of any kind. This role involves project planning, risk management, and progress monitoring.

Main Responsibilities:

- Project planning and scheduling
- Risk assessment and mitigation
- Team coordination and motivation

Business Analyst & Product Owner:

Role: The Business Analyst and Product Owner, Ezgi Erkaş, plays a role in the leading of the documentation and the general understanding of the business requirements and use cases. This role acts as a bridge between the technical responsibilities and the general business requirements.

Responsibilities:

- Requirement documentation and analysis
- Specification of the use cases and the documentation of them
- Feedback collection and analysis from the stakeholders

Software Engineer & Developer:

Role: The Software Engineer & Developer, Mustafa Zemin, is mainly responsible for designing the technical part of the project. This role is responsible for the main system architecture, determining the technology stack to be used, designing the database system and ensuring the general code quality of the project.

Responsibilities:

- System architecture and design
- General coding and development
- Design and manage the database system
- Code quality assurance
- Technical problem-solving

Web Developer & UI/UX Designer:

Role: The Web Developer & UI/UX Designer, Umut Güler, focuses on creating a user-friendly interface and developing the web code of it. The general front-end (for the user interface) and back-end (for the database communication) codes are developed by this role.

Responsibilities:

- User interface design
- Front-end development
- Back-end development
- Collaborating with the technical team

Work Areas:

The project is structured into distinct work areas, with each team member focusing on their respective domain:

Development: The Development work area is led by itself and is mainly responsible for coding, software development, and the technical aspects of the project.

Business Analysis: The Business Analysis work area, led by the Business Analyst, focuses on understanding, documenting, and managing the business requirements and the use cases.

User Interface and User Experience (UI/UX) Design: The UI/UX Design work area is responsible for creating visually appealing and easy-to-use user interfaces for the application.

Neighboring Projects:

Currently, there are no neighboring projects that could slow, interfere, intersect or impact the ReserveWell application. This situation allows the Reserwell team to give their full focus to the project itself.

Communication Channels:

The effective communication among the project team is of high importance to the success of the project. The following communication channels will be used for this purpose:

Weekly Progress Meetings: General face-to-face meetings will be held in order to discuss the weekly progress. These meetings will cover the assessment of the last week and what to be done in the following week. The assessments and the planning parts will be done in a more general manner.

Weekly Online Meetings: These online meetings will be held two times a week. In these meetings, the

team members will present what value they have created for the project in a detailed fashion. The feedback from the other team members will be gathered and discussed in detail. About the following days, the plans and the specific requirements will be examined and scheduled.

Daily Standup Meetings: In the code development phase of the project, daily online standup meetings will be held. In these meetings, the past and future developments will be discussed on a daily basis. These brief daily meetings ensure team synchronization, providing an opportunity to discuss progress, challenges, and coordinate efforts.

Iteration Meetings: After each iteration, a comprehensive meeting should be conducted by our team members. In these meetings, the previous iteration should be examined in a highly detailed fashion. All the mistakes, lessons learned, what has been done poorly or well, what should be avoided etc. should be discussed comprehensively. In addition to these, the next iteration is to be planned and revised (if needed) in these meetings.

3 Project practices and measurements

ReserveWell's success relies heavily on the adoption of effective management and technical practices. We have carefully defined our project practices and measurements to ensure a structured and productive development process:

Development Methodology:

Unified Process (UP) Methodology: The project will follow a UP approach. This methodology is considered to be a successful approach for a dynamic project environment, and it will enable us to adapt to changing requirements efficiently.

Management Practices:

Daily Standup Meetings: These daily meetings serve as a crucial point for our communication, collaboration and synchronization. Team members discuss their progress, address issues, and plan the future goals in detail. This practice allows us to monitor our progress and resolve any potential roadblocks.

Weekly Planning Meetings: We conduct weekly planning meetings to define and prioritize the work items for the upcoming iterations and phases. This helps us maintain a clear focus on the most important and immediate goals and deliverables. These meetings are planned to take charge of the duty of both the retrospective and the sprint meetings of the agile development approach.

Technical Practices:

Continuous Integration: Continuous integration is integral to our technical practices. We will ensure that new code integrates seamlessly and efficiently into the existing codebase, and the deployment process of the new code will be automated. This practice enhances code quality and reduces integration challenges.

Independent Unit and End-to-End Testing: Independent testing is a fundamental practice for verifying the quality of code and functionalities. Our testing team will rigorously evaluate the system, conducting unit (for the small parts of the application) and end-to-end (for the application as a whole) tests in order to ensure that it meets the defined requirements and standards.

Progress Tracking:

Iteration Assessments: We use iteration assessments to evaluate the success of each iteration. These assessments help us measure our progress and identify areas that need any kind of improvement. They planned to provide valuable insights to our iterative development process. These assessments are to be done in the iteration meetings.

Velocity Measurements: We collect velocity metrics to track the team's productivity. Velocity represents the number of completed work item points per iteration. This measurement is a valuable asset for assessing the team's efficiency, productivity and collaboration.

Automated Iteration Burndown Reports and Other Performance Monitoring Reports: In the code development part, we will use an automated issue and project tracking tool such as Jira. The iteration burndown charts, reports (which are used for visualizing the work done and remaining for each iteration) and other performance monitoring reports will be developed automatically in that tool. These reports serve as a reference for tracking our progress and making informed decisions.

Our chosen project practices and measurements are considered to align appropriately with our UP methodology and are designed to facilitate effective development, communication, synchronization and collaboration. These practices are aimed to help us maintain a high level of productivity, adapt to requirement changeability, and ensure the successful delivery of our Restaurant Reservation System, ReserveWell.

4 Project milestones and objectives

The Restaurant Reservation System project is organized into four iterations, each with a set of primary objectives, risks to mitigate, and key use case scenarios to develop. Note that the target velocities

depict the points given to the tasks which are to be completed in each iteration. Those are the divided tasks according to the Unified Process methodology. The whole project is going to be divided into manageable tasks where each of them can be estimated. For each task, our team will gather, and give points to the tasks according to their estimated efforts and times. The mentioned points can be found in the work items list document. However, several details that are not stated in the work items list are distributed iteration-wise. Therefore, the points in the two documents (work items list and the project plan) may differ. The risks are to be found in the risk list document. The project milestones and objectives are as follows:

Iteration	**Primary objectives** (risks and use case scenarios)	**Scheduled start or milestone**	**Target velocity**
I1 Inception Phase	Objectives 1. Define the project scope 2. Capture high-level requirements, such as the system's performance, security, and scalability. 3. Develop the use case for making restaurant reservations 4. Mitigate risk 1	22.10.2023 - 06.11.2023	29
I2 Elaboration Phase	Objectives 1. Mitigate risk 6 2. Refine detailed requirements and use cases 3. Develop the use case for managing reservations 4. Design the main system architecture 5. Begin construction of the core reservation system, implementing essential features.	06.11.2023 - 27.11.2023	23
I3 Elaboration Phase	1. Mitigate risks 2, 5, 7 2. Continue developing core reservation features, such as real-time reservation updates 3. Conduct the unit tests	27.11.2023 - 18.12.2023	32
I4 Construction Phase	1. Mitigate risks 9, 10 2. Construct and finalize the product for production deployment	18.12.2023 - 02.01.2024	50

	3. Conduct end-to-end tests		
I5 Transition Phase	1. Finalize the deployment phases 2. Release the end product	02.01.2024 - 23.01.2024	26

5 Deployment

The ReserveWell application will be deployed on the deployment platform which is known for its seamless and automated deployment processes for Next.js applications known as “Vercel”. The deployments are made automatically when the code is pushed on the GitHub repository. Version control will be managed through GitHub to track code changes and maintain a clear audit trail.

Key Deployment Aspects:

Automated Deployment: Vercel's automated deployment feature ensures that any code changes pushed to the GitHub repository trigger immediate deployment, which streamlines and automates the release process of the product.

Database Considerations: The deployment strategy will address database management to ensure data integrity and availability. Regular backups will be scheduled, and robust recovery procedures will be ready for the risks of any kind of data loss.

Monitoring Mechanisms: Real-time monitoring tools will track system performance, including the uptime of the server and the response times. This approach enables the capture of the issues with the system which maintains an optimal user experience.

Feedback and Support: A feedback mechanism will collect user input after deployment which guides the system improvements. Additionally, user training materials and support resources will facilitate a seamless transition for the diners and the restaurant staff.

The deployment strategy, combining Vercel's automation, GitHub version control, and robust monitoring mechanisms, ensures efficient, reliable, and secure software deployment for ReserveWell.

6 Lessons learned

Throughout the development of ReserveWell, our project team collaborated in several meetings to discuss the project's progress, identify areas for improvement, and capture valuable lessons from our

experiences. These meetings improved the continuous enhancement in various aspects of our project, including the development environment, iterative development, team collaboration, quality assurance.

Key Lessons and Actions Learned:

Development Environment: We learned the importance of maintaining a stable and consistent development environment. For this aspect, we will establish a comprehensive development environment checklist and ensure that the members of our team adhere to it. This will lead to a more streamlined and “semi-automated” development process.

Iterative Development and Change Management: Our experience highlighted the benefits of iterative development in addressing evolving project needs and challenges. We will continue to embrace this practice, focusing on regular assessments more effectively. Maintaining a continuous process with divided manageable tasks benefitted the change management considerably. With these manageable pieces of tasks, the changes in the requirements are to be handled much more efficiently.

Team Collaboration: Effective teamwork and collaboration were the key aspects to our project's success. We will strengthen these aspects by providing open communication between the team members, knowledge sharing, and regular cross-functional team meetings to maintain a shared understanding of our project goals.

Quality Assurance: We recognized the significance of rigorous testing and quality assurance. Not only for the code, but also for the documentation, the importance of unit and end-to-end tests are realized to be notable.

By applying these lessons learned, we aim to improve the development process, maintain efficient team collaboration, and consistently deliver a high-quality system in general. We plan to commit to the continuous improvement, adaptation and collaboration for getting the best for our project, ReserveWell.

After Iteration 1 and Iteration 2, the above lessons are learned more clearly. As a group, in order to increase our team collaboration for the productivity, we increased our number of meetings and scheduled them in more detail. For the change management, we shortened our code commits in order to see and comprehend the differences and the desired changes more meticulously.

After Iteration 3, the mentioned lessons are acknowledged even more in detail. We comprehended the importance of the planning parts, and we understood that the division of the work items, the separation of the tasks and the load of each iteration should be planned more carefully. The works that can be done in a single iteration while understanding the time and people constraints is learned to be planned and comprehended in a meticulous manner.