SOFTWARE REQUIREMENTS SPECIFICATION

for

<Project>

Version 1.0 approved

Prepared by <author>

<Organization>

June 24, 2024

Contents

1 Overview

Preface

The Software Requirements Specification (SRS) provides a comprehensive outline of what the developers need to create for the system. It encompasses both user requirements and a thorough description of system specifications. This document starts with an overview of user needs, transitioning into detailed system requirements. As an external contractor tasked with developing the system, Efficiency Monitor, we recognize the importance of this SRS in defining the technical and non-functional requirements of the software. It ensures that we don't overlook system-wide requirements while concentrating on functional aspects. In essence, this SRS document will primarily delineate user requirements along with overarching non-functional system requirements.

Readership

This requirements document is designed for a broad audience, including senior management, software engineers, and everyone involved in the project. Its primary audience includes customers, system analysts, project managers, designers, developers, and testers. Essentially, anyone connected to the development and maintenance of the software will refer to this document.

The SRS will be reviewed by our customers to confirm that it accurately reflects the requirements gathered from user interviews. Designers will use it as a fundamental resource for the design phase. Developers and testers will rely on this document to understand their tasks and ensure they meet the specified requirements. In summary, this document serves as a critical reference point for all stakeholders involved in the software's lifecycle.

Version History

Since this is the introductory version of the "Odyssey Travels" application, there is no previous history or version to reference. In version 1.0 of Odyssey Travels, our goal is to meet all identified user requirements gathered through extensive study and customer interviews (as detailed in Chapter 4 of this document). Should users encounter any limitations or shortcomings while using our product and provide feedback or request new features, we will release updated versions of Odyssey Travels. These updates will address any reported issues, incorporate requested features, or introduce enhancements based on our own assessments.

2 Introduction

2.1 Purpose

The purpose of this document is to provide a comprehensive description of the web-based project named "Odyssey Travels," developed using Next.js. It aims to outline the system's objectives, functionalities, user interfaces, operational constraints, and how it handles external interactions. This document serves as a detailed guide for stakeholders and developers involved in the project, ensuring a clear understanding of the system's scope and requirements. It will define how "Odyssey Travels" enhances the online travel booking experience through innovative features and responsive web interfaces.

2.2 Scope

This web-based application, "Odyssey Travels," is designed to enhance user efficiency and streamline travel planning processes. It aims to empower users by providing intuitive tools to manage and prioritize travel itineraries and bookings seamlessly. Users will no longer need to rely on traditional methods like spreadsheets or multiple websites to organize their travel plans. "Odyssey Travels" will enable users to categorize and prioritize travel activities effortlessly, ensuring optimal utilization of their time and resources. By offering clear insights into itinerary management and suggesting efficient scheduling of activities, the application enhances user productivity while maintaining ease of use. Specifically, the system will guide users in making informed decisions about their travel plans, suggesting ideal times for activities to maximize efficiency throughout their journey. It will help users save valuable time by centralizing booking processes and providing real-time updates on travel arrangements. Additionally, the application will assist users in optimizing their itineraries by recommending adjustments or identifying unnecessary tasks, ensuring that their travel experiences are both productive and fulfilling. Overall, "Odyssey Travels" aims to revolutionize the travel planning experience by offering a user-friendly interface that simplifies task management and enhances productivity, thereby meeting the diverse needs of modern travelers.

2.3 Organization

This document consists of 7 chapters, each serving a distinct purpose in detailing the "Odyssey Travels" project. The following chapter, the glossary section, provides definitions for technical terms used throughout this document, aimed at enhancing readability and understanding for all stakeholders. The third section, titled User Requirement Definition, outlines the services offered to users and encompasses non-functional system requirements. This section utilizes natural language, diagrams, and other notations that are accessible and comprehensible to our customers. Chapter 4, System Architecture, offers a high-level overview of the planned system architecture, illustrating how functions are distributed across various modules. It also highlights components that are reused within the architecture. The subsequent chapter, System Requirement Specification, delves deeper into both functional and non-functional requirements, offering detailed descriptions and specifications. Graphical system models depicting relationships between system components and their environment are included in Chapter 6, titled System Models. Chapter 7, System Evolution, outlines the core assumptions underpinning the system's design and anticipates future changes influenced by hardware advancements and evolving user needs. Appendices containing detailed, application-specific information relevant to "Odyssey Travels" are appended after the 7th section, providing supplementary details for stakeholders and developers.

2.4 References

IEEE: IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications, IEEE Computer Society, 1998.

3 Glossary

3.1 Web Framework

Next.js – A JavaScript framework designed for building server-side rendered (SSR) React applications, developed by Vercel.

3.2 Supported Devices

Next.js application – Any web application built using Next.js. In this document, synonymous with web application developed with Next.js

3.3 DFD (Data Flow Diagram)

A graphical representation that refers the system using functions performed in the system and data produced by these functions.

3.4 Interface

A specification of the attributes and operations associated with a software component. The interface is used as the means of accessing the component's functionality.

3.5 Maintenance prediction

The process in which project managers try to predict what system changes might be proposed and what parts of the system are likely to be the most difficult to maintain.

3.6 Object Model

A model of a software system that is structured and organized as a set of object classes and the relationships between these classes.

3.7 Object-Oriented (OO) Approach

An approach to software development where the fundamental abstractions in the system are independent objects.

3.8 Reliabity

The ability of a system to deliver services as specified. Reliability can be specified quantitatively as a probability of failure on demand or as the rate of occurrence of failure.

3.9 Requirement, functional

A statement of some function or feature that should be implemented in a system.

3.10 Requirement, non-functional

A statement of a constraint or expected behavior that applies to a system. This constraint may refer to the emergent properties of the software that is being developed or to the development process.

3.11 Sequence Diagram

A diagram that shows the sequence of interactions required to complete some operation. In the UML, sequence diagrams may be associated with use cases.

3.12 Software Requirements Specification

A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.

3.13 Software Architecture

A model of the fundamental structure and organization of a software System.

3.14 Stakeholder

Any person with an interest in the project who is not a developer.

3.15 User

Someone who interacts with the Web Browser.

3.16 User Interface

The way in which information produced by the system is displayed and the way in which system users can access system functionality.

3.17 Unified Modeling Language (UML)

A graphical language used in object-oriented development that includes several types of system models that provide different views of a system.

3.18 Use Cases

Use cases are a requirements discovery technique that identifies the actors involved in an interaction and names the type of interaction.

4 User Requirements Definition

This section encompasses all functional and quality requirements of the system, including non-functional requirements. It provides a comprehensive description of the system and its complete feature set. To ensure clarity and avoid misunderstandings, user requirements are articulated in natural language. This version clarifies the purpose of the section while maintaining its original intent and focus.

4.1 External Interface Requirements

4.1.1 User Interface

- **Home**:
  - Serves as the main landing page.
  - Provides an overview of offerings and services.
  - Designed for intuitive navigation and accessibility.
- **About**:
  - Provides comprehensive project information.
  - Includes objectives, mission statement, team members, and background details.
  - Builds trust and transparency.
- **Packages**:
  - Users explore various travel options and service offerings.
  - Includes detailed descriptions, pricing information, and possibly images or videos.
  - Supports informed decision-making.
- **Login**:
  - Registered users access their accounts securely.
  - Provides access to booking history, saved preferences, and account settings.
  - Enhances user engagement and personalized experiences.
- **Signup**:
  - New users create accounts easily.

  - Access to exclusive features and personalized recommendations.
  - Facilitates seamless booking experiences.

- **Contact Us**:
  - Provides essential contact information.
  - Includes email addresses, phone numbers, and possibly a contact form.
  - Facilitates direct communication with customer support or the project team.

Design Considerations

Each section is designed with:

- User-friendliness in mind.
- Intuitive navigation and accessibility across different devices and platforms.
- Home page as the focal point for engaging users.
- About section for building trust and transparency.
- Packages section for comprehensive showcasing and informed decision-making.
- Seamless Login and Signup process for enhanced engagement and personalized experiences.
- Contact Us section for easy access to customer support and inquiries.
- Promotion of customer satisfaction and support throughout user interaction.

4.1.1 Hardware Interfaces

Odyssey Travels, the hardware interface consists of Android-based smartphones. There are no specific hardware requirements for the Android devices used, except that the system must be compatible with any version of Android

4.2 Functional Requirements

This section outlines the fundamental actions required for the Odyssey Travels software system:

- The system must allow users to view available travel packages and details.
- The system must provide options to filter packages based on destination, duration, and price range.
- The system must display a progress bar indicating booking completion and show the percentage of bookings done.

- The system must offer a statistical view of popular travel destinations, package categories, and user preferences.
- The system must send notifications for important booking updates and travel reminders.
- The system must display average user ratings and reviews for each travel package.
- The system must allow users to upload images or documents (e.g., passport copies) securely for booking purposes.
- The system must include a calendar view for users to select travel dates and check availability.
- The system must provide a profile section for users to manage personal information, preferences, and past bookings.
- The system must feature interactive travel quizzes to engage users and provide insights into destinations.
- The system must maintain a history of user bookings and travel preferences for personalized recommendations.
- All functionalities can be accessed and executed using the touchscreen interface of a smartphone.

All functionalities can be accessed and executed using the touchscreen interface of a smartphone.

4.3 Non Functional Requirements

- System must provide a pleasant and user-friendly graphical interface.
- System must provide multilingual advantage, especially Bengali and English.
- System must respond within a few seconds.
- System must work with relatively long input lists.
- System mustn't alter the task list automatically.
- System must restart instantly in case of failure without any data loss.
- System must be usable to every class of users, even if they have no technical knowledge.
- System must work cooperatively with other key processes without causing any delay.

5 System Architecture

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with user and introduce the basic functionality of it. It will also describe what functionality is available for stakeholders. At last, the constraints and assumptions for the system will be presented.

5.1 Product perspective and functions

Odyssey Travels is a mobile application designed for managing travel plans and bookings efficiently. Users can add and prioritize daily tasks based on their travel itinerary, ensuring seamless organization and tracking of completed tasks. The application includes a "Priority Game" feature, offering a quiz game to entertain users during their travel downtime. Additional minor functions will be detailed later in this report. In this subsection, we will create a block diagram illustrating the main functionalities of our system. The block diagram, depicted in Figure 5.1, will provide a visual representation of the key features and interactions within Odyssey Travels.

6 System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

6.1 System Feature 1

<Don't really say "System Feature 1." State the feature name in just a few words.>

6.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

6.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

6.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1: REQ-2:

6.2 System Feature 2 (and so on)

7 Other Nonfunctional Requirements

7.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

7.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

7.3 Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

7.4 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

7.5 Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

8 Other Requirements

<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.>

8.1 Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

8.2 Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

8.3 Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>