

**Department of Computer Science and Engineering**

**SRS REPORT**

**Assignment No.: 01**

**Assignment on: Software Requirements Specification (SRS)**

**Submitted By :**

**Mmk Mahin**

**ID:221-15-5580**

**Section:61-U**

CSE236 | SOFTWARE PROJECT - 2

**Submitted to:** Md**.** Golam Rabbany

**Dept. of CSE, Daffodil International University**

**wertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnm**

Niloy Sarker

Managing Director

Go Travel

Dhaka 1204

Subject: Application for agreement of Travel Management System Project.

Dear Mr sarker

I hope this letter finds you in good health and high spirits. I am writing to propose the development of a comprehensive Travel Management System that I believe will significantly enhance the efficiency and quality of your Travel planning services. I am the member of SoftTech company. I have identified the whole system of your Travel company and decided to create a unique Travel management system.

According to your business requirement we design GO TRAVEL Website and Web Application System for you, in this agreement paper we right each and every part and all possible feature which covered our application. So please read carefully and if it satisfies you then sign and confirm the agreement.

Sincerely

MMK Mahin

Head

Application development Brach

SoftTech

**Table of Contents**

[Software Requirements Specification (SRS) 5](#_Toc150402964)

[1.Introduction 5](#_Toc150402965)

[1.1 Purpose 5](#_Toc150402966)

[1.2 Scope 5](#_Toc150402967)

[1.3 Definitions, Acronyms, and Abbreviations 5](#_Toc150402968)

[2. Project Goals 5](#_Toc150402969)

[2.1 Primary Goals 5](#_Toc150402970)

[2.2 Secondary Goals 5](#_Toc150402971)

[3. Features 5](#_Toc150402972)

[3.1 User Registration and Authentication 5](#_Toc150402973)

[3.2 Trip Planning 6](#_Toc150402974)

[3.3 Destination Information 6](#_Toc150402975)

[3.4 Booking and Reservations 6](#_Toc150402976)

[3.5 User Profile Management 6](#_Toc150402977)

[3.6 Reviews and Recommendations 6](#_Toc150402978)

[3.7 Notifications 6](#_Toc150402979)

[3.8 Search and Filters 6](#_Toc150402980)

[4.Functionality 6](#_Toc150402981)

[4.1 User Roles and Permissions 6](#_Toc150402982)

[4.2 User Interface 6](#_Toc150402983)

[4.3 Database Management 7](#_Toc150402984)

[4.4 Integration with External Services 7](#_Toc150402985)

[4.5 Security Measures 7](#_Toc150402986)

[4.6 Performance and Scalability 7](#_Toc150402987)

[5. Milestone & Reporting 7](#_Toc150402988)

[6. Software Development Life Cycle (SDLC) 8](#_Toc150402989)

[6.1 Purpose 8](#_Toc150402990)

[6.2 Phases 8](#_Toc150402991)

[6.3 Activities 8](#_Toc150402992)

[6.4 Documentation 8](#_Toc150402993)

[6.5 Roles and Responsibilities 8](#_Toc150402994)

[6.6 Iterations and Feedback 8](#_Toc150402995)

[6.7 Testing 8](#_Toc150402996)

[6.8 Change Management 8](#_Toc150402997)

[6.9 Risk Management 8](#_Toc150402998)

[6.10 Quality Assurance 9](#_Toc150402999)

[6.11 Deliverables 9](#_Toc150403000)

[6.12 Team Roles 9](#_Toc150403001)

[7.Types of Software Development Methodologies: 9](#_Toc150403002)

[7.1 Waterfall Model: 9](#_Toc150403003)

[7.2 Agile Methodology (Scrum and Kanban): 9](#_Toc150403004)

[7.3 Iterative Model: 9](#_Toc150403005)

[7.4 Spiral Model: 9](#_Toc150403006)

[7.5 V-Model (Validation and Verification Model): 10](#_Toc150403007)

[7.6 Rapid Application Development (RAD): 10](#_Toc150403008)

[7.7 Incremental Model: 10](#_Toc150403009)

[7.8 DevOps: 10](#_Toc150403010)

[8.The methodology I would prefere to develop my application and why? 10](#_Toc150403011)

[9. Testing 11](#_Toc150403012)

[10. Payment Terms & condition 11](#_Toc150403013)

[11. Responsibility 12](#_Toc150403014)

[12. Contact Us 12](#_Toc150403015)

[13. Agreement Signed By: 12](#_Toc150403016)

# Software Requirements Specification (SRS)

## 1.Introduction

This document includes specifications and features of the software in detail. It helps understand the target audience and user classes accordingly and internal and external interface requirements.

### 1.1 Purpose

The Software Requirements Specification (SRS) outlines the goals, features, functionality, and constraints of the "Go Travel" project. This document serves as a guide for the development team, stakeholders, and project managers to ensure a clear understanding of the project's requirements.

### 1.2 Scope

"Go Travel" is a web-based travel planning and booking platform that aims to simplify the travel experience for users by providing access to destination information, trip planning tools, and the ability to book various travel services.

### 1.3 Definitions, Acronyms, and Abbreviations

SRS: Software Requirements Specification

UI: User Interface

API: Application Programming Interface

## 2. Project Goals

### 2.1 Primary Goals

* To provide users with a user-friendly platform for planning and booking trips.
* To offer comprehensive destination information and travel recommendations.
* To facilitate secure and convenient booking of flights, accommodations, and activities.
* To build a community where users can share travel experiences and reviews.

### 2.2 Secondary Goals

* To implement a responsive and aesthetically pleasing user interface.
* To support multiple languages and currencies.
* To integrate with external services for real-time pricing and availability.

## 3. Features

### 3.1 User Registration and Authentication

* User registration with email or social media accounts.
* User login and authentication mechanisms.
* Password reset functionality.

### 3.2 Trip Planning

* Create, edit, and save trip itineraries.
* Add and remove destinations and activities to itineraries.
* Calendar view for trip planning.

### 3.3 Destination Information

* Detailed information about destinations, including attractions, weather, and local tips.
* User-generated reviews and ratings for destinations.

### 3.4 Booking and Reservations

* Flight booking and reservation system.
* Hotel and accommodation booking.
* Activity and tour reservations.
* Payment processing and invoice generation.

### 3.5 User Profile Management

* User profile creation and editing.
* Profile picture upload.
* History of past trips and bookings.

### 3.6 Reviews and Recommendations

* User-generated reviews and ratings for accommodations, activities, and destinations.
* Personalized travel recommendations based on user preferences and history.

### 3.7 Notifications

Email and in-app notifications for booking confirmations, updates, and reminders.

### 3.8 Search and Filters

* Advanced search functionality for destinations, accommodations, and activities.
* Filters for price range, location, and user ratings.

## 4.Functionality

### 4.1 User Roles and Permissions

User roles: guest, registered user, admin.

Permissions based on roles.

### 4.2 User Interface

* Intuitive and responsive design for web and mobile platforms.
* Consistent navigation and layout.
* Interactive trip planning interface.

### 4.3 Database Management

* Database to store user data, trip itineraries, bookings, and reviews.
* Regular backups and data security measures.

### 4.4 Integration with External Services

* Integration with payment gateways for secure transactions.
* Integration with third-party APIs for real-time data (flights, accommodations, activities).

### 4.5 Security Measures

* Secure data transmission using HTTPS.
* Encryption of sensitive user data.
* User authentication and authorization.

### 4.6 Performance and Scalability

* Efficient coding practices for optimal performance.
* Scalability to accommodate a growing user base.

## 5. Milestone & Reporting

## 

|  |  |  |  |
| --- | --- | --- | --- |
| **Milestone** | **Task** | **Reporting** | **Required Time** |
| Analyzing Project | Submit Idea & Design | Submit The Design | 3 Days |
| Requirements Collection | Gathering Data & submit |  | 5 Days |
| Development | Work with the project from the root | Review The Work | 21 Days |
| Testing | Testing the entire application system |  | 10 Days |
| Deployment | Fining and Review Final project | Review Final Work | 7 Days |
| Delivery Project | Available to Online platform | Live On Server | 5 Days |

## 

## 6. Software Development Life Cycle (SDLC)

Software Development Life Cycle (SDLC) methodologies are systematic processes and frameworks used by software development teams to plan, design, build, test, deploy, and maintain software applications. These methodologies provide a structured approach to managing the entire software development process. Each SDLC methodology follows a set of phases, activities, and best practices to ensure that software projects are completed efficiently and with high quality.

### 6.1 **Purpose**

The primary purpose of a software development methodology is to provide a set of best practices and guidelines to help development teams plan, execute, and complete software projects successfully. It aims to ensure that software is developed efficiently, meets quality standards, and satisfies customer requirements.

### 6.2 **Phases**

SDLC methodologies are typically divided into phases, which are the stages of development. Common phases include requirements analysis, design, coding, testing, deployment, and maintenance. The order and number of phases can vary between methodologies.

### **6.3 Activities**

Each phase involves a set of specific activities that must be performed. **For example**, the design phase may include activities such as architectural design, database design, and user interface design.

## 6.4 **Documentation**

SDLC methodologies emphasize the importance of documentation. Detailed documentation is created at each phase to capture requirements, design decisions, coding guidelines, and testing plans.

### **6.5 Roles and Responsibilities**

SDLC methodologies define roles and responsibilities for team members, including developers, testers, project managers, and stakeholders. This ensures that everyone understands their roles in the development process.

### 6.6 **Iterations and Feedback**

Some SDLC methodologies, such as Agile, emphasize iterative development with frequent feedback and collaboration between the development team and stakeholders.

### 6.7 **Testing**

Testing is a critical component of SDLC methodologies, ensuring that the software is reliable and free of defects. Different methodologies may have varying approaches to testing.

### 6.8 **Change Management**

SDLC methodologies often include change management processes to handle modifications to project requirements or scope.

### 6.9 **Risk Management**

Many methodologies incorporate risk assessment and mitigation strategies to identify and address potential project risks.

### **6.10 Quality Assurance**

Ensuring the quality of the software is a central concern in SDLC methodologies. Various methodologies may have different quality assurance processes.

### **6.11 Deliverables**

Each phase of an SDLC produces specific deliverables, such as design documents, source code, test plans, and user manuals. These deliverables help track progress and ensure that project goals are met.

### **6.12 Team Roles**

Many methodologies define specific roles within a development team, such as product owners, scrum masters, and developers. These roles have specific responsibilities and contribute to the successful execution of the methodology.

## 7.Types of Software Development Methodologies:

There are several types of software development methodologies, each offering a distinct approach to managing and executing software projects. Here are some of the most commonly used software development methodologies:

### 7.1 Waterfall Model:

The Waterfall model is a linear and sequential approach to software development. It consists of distinct phases, such as requirements, design, implementation, testing, deployment, and maintenance. Each phase must be completed before moving on to the next. It's suitable for projects with well-defined and stable requirements.

### 7.2 Agile Methodology (Scrum and Kanban):

Agile methodologies are iterative and flexible, promoting collaboration and adaptability. They emphasize breaking the project into small increments and delivering working software in short cycles. Continuous feedback and user involvement are key principles. Scrum and Kanban are specific Agile frameworks with regular, time-boxed iterations (sprints).

### 7.3 Iterative Model:

The Iterative model involves developing a partial system or version and then refining it through repeated iterations. Each iteration can add new features or improve existing ones. It's suitable for projects where requirements may evolve or where prototyping is valuable.

### 7.4 Spiral Model:

The Spiral model combines elements of the Waterfall and Iterative models. It emphasizes risk analysis, with each cycle consisting of planning, risk analysis, engineering, and evaluation. It's beneficial for projects with high uncertainty and significant risks.

### 7.5 V-Model (Validation and Verification Model):

The V-Model is an extension of the Waterfall model that focuses on validation and verification. It emphasizes testing at each stage of development, corresponding to a verification phase and a validation phase. It's suitable for projects with strong testing requirements.

### 7.6 Rapid Application Development (RAD):

RAD emphasizes rapid prototyping and quick development. It involves building prototypes and iteratively refining them based on user feedback. It's ideal for projects where speed and user involvement are critical.

### 7.7 Incremental Model:

The Incremental model divides the software into smaller, manageable parts, and each part is developed separately. New increments build upon the previous ones. It's suitable for projects that can be divided into distinct and independently deliverable components.

### 7.8 DevOps:

DevOps isn't a traditional SDLC model but rather a cultural and technical approach that emphasizes collaboration between development and operations teams. It focuses on automation, integration, and continuous delivery (CI/CD). DevOps is beneficial for projects requiring rapid development, testing, and deployment.

## 8.The methodology I would prefere to develop my application and why?

In my case, developing a phonebook application, it's likely thatanAgile methodologylike Scrum or Kanban would be a good choice. These methodologies provide the flexibility to quick changes, gather user feedback, and incrementally build and refine theapplication. Also**,** the specific choice will depend on project size, and other contextual factors**.**

**Agile Methodology (Scrum):**

**Flexibility for Evolving Requirements:**

The project's primary and secondary goals include features that may evolve over time, such as user registration, trip planning, destination information, and booking. Agile allows for flexibility in accommodating changes and additions to requirements.

**Regular Collaboration with Stakeholders:**

Agile methodologies, including Scrum, emphasize close collaboration with stakeholders. This aligns well with your goal of building a community where users can share travel experiences and reviews. Frequent interactions and feedback are essential for this feature.

**Iterative Development**:

Agile promotes iterative development with short development cycles. Your project's milestone and reporting section includes tasks for development, testing, and reviewing work, which can fit well within Scrum's iterative approach.

**Continuous Improvement:**

Agile methodologies encourage continuous improvement and adaptation. For features like user-generated reviews and recommendations, this iterative approach can be valuable to refine algorithms and personalization.

**Quick Deployment:**

The project includes deployment as a milestone, indicating a need for timely releases. Agile, particularly Scrum, focuses on delivering potentially shippable increments at the end of each sprint.

**Transparency:**

Agile provides transparency into the project's progress through frequent meetings and status updates, ensuring that all stakeholders are informed and aligned.

## 

**Dynamic Scaling:**

Agile can handle dynamic scaling as your user base grows. The user roles and permissions, user profiles, and the community aspect of your application can benefit from Agile's adaptability.

**User-Centric Design:**

The project goals include providing a user-friendly platform and responsive user interface. Agile emphasizes user-centric design and incorporates user feedback into the development process.

## 9. Testing

Following features will be used for testing

* This Application will be tested with Agile model
* Application will be tested by PHPUnit.
* Application also tested By Codeception.

## 10. Payment Terms & condition

15% payment will be accepted for the Project proposal and design Submission.

45% payment will be accepted for the Application Development

70% payment will be accepted after application review and Testing

100% payment will be accepted after handover the fully completed Application

## 11. Responsibility

The entire Application has been done by MMK Mahin and all the responsibility including terms and condition will goes to him.

## 12. Contact Us

You can get in touch with us in any of the below ways:

Golam Rabbany

By Phone: +8801787774996

By Email

grabbany1234@gmail.com

## 13. Agreement Signed By:

|  |  |  |
| --- | --- | --- |
| ……………………………    Client Signature  Niloy sarker  Managing Director | …………………………..    Order Provider Signature  MMK Mahin  Officer  SoftTech | ……………………………    Authority Signature  Golam Rabbany  Managing Director (MD)  SoftTech |