**Software Requirements Specification (SRS)**

**For Web up**

**Version:** 1.0

**Prepared by:**

Adjaoud ilias zoheir

Tertag noureddine

Nemdil amira

Aguenarous abdelghani

Arab lydia marya

**Date:** 22/October/2024

**Table of Contents**

1. Introduction  
   1.1 [Purpose](#11-purpose)  
   1.2 [Document Conventions](#12-document-conventions)  
   1.3 [Intended Audience and Reading Suggestions](#13-intended-audience-and-reading-sugges)  
   1.4 [Product Scope](#14-product-scope)  
   1.5 [References](#15-references)
2. [Overall Description](#2-overall-description)  
   2.1 [Product Perspective](#21-product-perspective)  
   2.2 [Product Functions](#22-product-functions)  
   2.3 [User Classes and Characteristics](#23-user-classes-and-characteristics)  
   2.4 [Operating Environment](#24-operating-environment)  
   2.5 [Design and Implementation Constraints](#25-design-and-implementation-constraint)  
   2.6 [User Documentation](#26-user-documentation)  
   2.7 [Assumptions and Dependencies](#27-assumptions-and-dependencies)
3. [External Interface Requirements](#3-external-interface-requirements)  
   3.1 [User Interfaces](#31-user-interfaces)  
   3.2 [Hardware Interfaces](#32-hardware-interfaces)  
   3.3 [Software Interfaces](#33-software-interfaces)  
   3.4 [Communications Interfaces](#34-communications-interfaces)
4. [System Features](#4-system-features)  
   4.1 **Project Setup with Next.js 14**  
   4.2 **Local Database with Prisma**  
   4.3 **Authentication in Next.js 14**  
   4.4 **Freelance Community & Landing Page**  
   4.5 **Subdomains Architecture**  
   4.6 **Prisma Integration in Next.js**  
   4.7 **Agency and Forms**  
   4.8 **Image Upload**  
   4.9 **Sidebar and Global State Management**  
   4.10 **Sub Account and Subdomain**  
   4.11 **Media Storage and Project Management System**  
   4.12 **Stripe**  
   4.13 **Funnels and Website Builder**
5. [Other Nonfunctional Requirements](#5-other-nonfunctional-requirements)  
   5.1 [Performance Requirements](#51-performance-requirements)  
   5.2 [Safety Requirements](#52-safety-requirements)  
   5.3 [Security Requirements](#53-security-requirements)  
   5.4 [Software Quality Attributes](#54-software-quality-attributes)  
   5.5 [Business Rules](#55-business-rules)
6. [Other Requirements](#6-other-requirements)

Appendices  
A. [Glossary](#appendix-a-glossary)  
B. [Analysis Models](#appendix-b-analysis-models)  
C. [To Be Determined List](#appendix-c-to-be-determined-list)

**1. Introduction**

**1.1 Purpose**

The purpose of this SRS is to define the requirements for the development of the **Web up** platform. This system is designed to provide agencies, businesses, and individual users with a scalable solution for creating, customizing, and managing websites and sales funnels. It integrates multivendor support, role-based access control, project management tools, media storage, and payment solutions powered by **Stripe**.

**1.2 Document Conventions**

* **High Priority Requirements**: Indicated with **(High)**.
* **Medium Priority Requirements**: Indicated with **(Medium)**.
* **Low Priority Requirements**: Indicated with **(Low)**.
* **Functional Requirements**: Numbered as **REQ-1**, **REQ-2**, etc.
* **To Be Determined (TBD)**: Features under discussion are labeled as TBD.

**1.3 Intended Audience and Reading Suggestions**

* **Developers**: Review Sections 3 and 4 for technical requirements and implementation details.
* **Project Managers**: Focus on Sections 1, 2, and 5 for project scope, dependencies, and key requirements.
* **Testers**: Refer to Section 4 and 5 for testable system features and performance requirements.
* **Business Analysts**: Focus on Section 2 to understand the user base and Section 4 for system features.

**1.4 Product Scope**

The **Web up** allows users to create websites, sales funnels, and project management systems with integrated multivendor functionality. The platform will support multiple user roles (agencies, vendors, sub-accounts), subdomain management, media storage, and Stripe for payment processing and subscription management.

**1.5 References**

* **Stripe API Documentation**: <https://docs.stripe.com/payments/link>
* **PCI DSS Compliance Standards**: <https://docs.adyen.com/development-resources/pci-dss-compliance-guide/>
* **GDPR Guidelines**: <https://gdpr-info.eu>
* **Next.js Documentation**: <https://nextjs.org/docs>
* **Prisma Documentation**: <https://www.prisma.io/docs>

**2. Overall Description**

**2.1 Product Perspective**

This SaaS solution extends beyond traditional website builders by incorporating multivendor capabilities, project management tools, and advanced analytics. Built on **Next.js 14**, with **Prisma** for database management and **Stripe** for payment processing, the platform is designed to scale efficiently and serve a diverse user base (agencies, vendors, freelancers, and end-users).

**2.2 Product Functions**

Key functionalities of the platform include:

* **Multivendor SaaS**: Allow multiple vendors to register and sell products/services.
* **Agency and Sub-account Management**: Agencies can create sub-accounts and assign role-based access.
* **Unlimited Funnel Hosting**: Users can create and host an unlimited number of sales funnels.
* **Website & Funnel Builder**: Drag-and-drop builder for creating websites and funnels.
* **Stripe Integration**: Subscription plans, add-on products, Stripe Connect for vendors, and custom Stripe forms.
* **Media Storage**: Secure storage for images and media files.
* **Project Management**: Tools like Kanban boards for managing tasks and projects.
* **Performance Metrics**: Dashboards and analytics for agencies and sub-accounts.
* **User Interface Themes**: Support for Light and Dark mode.

**2.3 User Classes and Characteristics**

* **Agencies**: Manage multiple client projects, create sub-accounts, and oversee billing and analytics.
* **Sub-account Users**: Access features based on roles assigned by agencies.
* **Vendors**: Sell products/services, manage storefronts, and handle transactions.
* **End-users**: Build websites and funnels with pre-built templates and tools.
* **Administrators**: Oversee platform operations, settings, and ensure regulatory compliance.

**2.4 Operating Environment**

* **Frontend**: Built with **Next.js 14**.
* **Backend**: Prisma will handle database operations. Cloud hosting on **AWS** or **Azure**.
* **Supported Browsers**: Chrome, Firefox, Safari, and Edge (latest versions).
* **Devices**: Responsive design for desktop and mobile.

**2.5 Design and Implementation Constraints**

* **Technology Stack**: Must use Next.js 14, Prisma, and Stripe.
* **Security**: Must comply with **PCI DSS** and **GDPR** standards.
* **Scalability**: The platform must support up to 50,000 concurrent users.

**2.6 User Documentation**

* **User Manuals**: Comprehensive documentation covering all features.
* **Online Help Center**: FAQs, tutorials, and support articles.
* **Video Tutorials**: Walkthroughs for key features like website builder and Stripe integration.

**2.7 Assumptions and Dependencies**

* **Stable Internet Access**: Assumes that users have stable internet connections.
* **Stripe Availability**: Assumes the Stripe payment gateway is available for all geographic regions.
* **Third-party APIs**: The platform relies on external APIs for payment processing, media storage, and project management.

**3. External Interface Requirements**

**3.1 User Interfaces**

* **Dashboard**: Customizable dashboard with widgets for viewing analytics and notifications.
* **Website & Funnel Builder**: A drag-and-drop interface for creating websites and sales funnels.
* **Project Management**: Kanban boards for tracking

 **Settings**:

* Profile management.
* Role and permission settings.

 **Notifications Panel**:

* Real-time updates.
* Actionable alerts.

**3.2 Hardware Interfaces**

* The platform requires no special hardware interfaces beyond standard input devices (keyboard, mouse, touchscreen).

**3.3 Software Interfaces**:

* **Prisma**: Interface with local databases.
* **Stripe**: Payment gateway and subscription handling.
* **Next.js**: Handles routing, middleware, and frontend/backend logic.

**3.4 Communications Interfaces**

* **HTTP/HTTPS**:
  + Secure communication between client and server.
* **WebSockets**:
  + Real-time features like notifications.
* **RESTful APIs**:
  + Integration with third-party services.

**4. System Features :**

#### **4.1 Project Setup with Next.js 14**

* **4.1.1 Description**: The project is built using Next.js 14 for both frontend and backend. The setup includes defining environments, configuration for routing, and initial deployment pipelines.
* **Functional Requirements**:
  + **REQ-1**: The system must be built on Next.js 14.
  + **REQ-2**: Implement project folder structures, middleware, and routing.

#### **4.2 Local Database with Prisma**

* **4.2.1 Description**: **Prisma** will be used to manage database models and interactions.
* **Functional Requirements**:
  + **REQ-1**: Use Prisma to set up the local database.
  + **REQ-2**: Implement CRUD operations for the website and user data.

#### **4.3 Authentication in Next.js 14**

* **4.3.1 Description**: Authentication is done using Next.js 14's built-in tools.
* **Functional Requirements**:
  + **REQ-1**: Use Next.js's built-in authentication middleware.
  + **REQ-2**: Implement secure authentication for agencies, subaccounts, and users.

#### **4.4 Freelance Community & Landing Page**

* **4.4.1 Description**: A landing page to showcase the product and allow users to sign up, possibly tying into the freelance community.
* **Functional Requirements**:
  + **REQ-1**: Create the landing page using Next.js.
  + **REQ-2**: Integrate form elements to capture leads and user sign-ups.

#### **4.5 Subdomains Architecture**

* **4.5.1 Description**: Each agency and subaccount will have its own subdomain for customization and content management.
* **Functional Requirements**:
  + **REQ-1**: Implement subdomain architecture in Next.js 14 using middleware.
  + **REQ-2**: Ensure that each agency and subaccount operates under separate subdomains.

#### **4.6 Prisma Integration in Next.js**

* **4.6.1 Description**: Use Prisma for database interaction in Next.js 14.
* **Functional Requirements**:
  + **REQ-1**: Implement Prisma to manage agency and subaccount data models.

#### **4.7 Agency and Forms**

* **4.7.1 Description**: Agency sections will allow for project and client management, including form creation and handling.
* **Functional Requirements**:
  + **REQ-1**: Build dynamic forms using Next.js 14.
  + **REQ-2**: Allow agencies to create custom forms for data collection.

#### **4.8 Image Upload**

* **4.8.1 Description**: Implement image upload functionality within Next.js 14.
* **Functional Requirements**:
  + **REQ-1**: Allow users to upload and manage images through media storage.

#### **4.9 Sidebar and Global State Management**

* **4.9.1 Description**: Implement a sidebar and global state management using **Zustand/Redux**.
* **Functional Requirements**:
  + **REQ-1**: Integrate a sidebar navigation system for the dashboard.
  + **REQ-2**: Manage global state with Zustand/Redux.

#### **4.10 Sub Account and Subdomain**

* **4.10.1 Description**: Manage sub-accounts and subdomains under agencies using dynamic routing and middleware in Next.js.
* **Functional Requirements**:
  + **REQ-1**: Implement sub-account features with role-based access control.
  + **REQ-2**: Enable subdomain routing for sub-accounts.

#### **4.11 Media Storage and Project Management System**

* **4.11.1 Description**: Allow users to store media and manage projects with a Kanban board.
* **Functional Requirements**:
  + **REQ-1**: Use media storage for uploaded files (images, videos).
  + **REQ-2**: Build project management tools including a Kanban board.

#### **4.12 Stripe Setup**

* **4.12.1 Description**: Integrate Stripe for payments, subscriptions, and product management.
* **Functional Requirements**:
  + **REQ-1**: Set up Stripe for payments.
  + **REQ-2**: Implement custom checkout forms and product synchronization with Stripe.

#### **4.13 Funnels and Website Builder**

* **4.13.1 Description**: Build the full website and funnel creation tool using Next.js 14.
* **Functional Requirements**:
  + **REQ-1**: Allow users to create and manage funnels and websites.
  + **REQ-2**: Implement real-time editing for website elements.

**5. Other Nonfunctional Requirements**

**5.1 Performance Requirements**

* **REQ-1**: Support at least 50,000 concurrent users without performance degradation.
* **REQ-2**: Average page load times should not exceed 2 seconds under normal load.
* **REQ-3**: System should process transactions within 2 seconds.

**5.2 Safety Requirements**

* **REQ-1**: Implement data backup and recovery procedures.
* **REQ-2**: Ensure system stability to prevent data loss.

**5.3 Security Requirements**

* **REQ-1**: All data transmission must use SSL/TLS encryption.
* **REQ-2**: Comply with GDPR and other data protection regulations.
* **REQ-3**: Implement role-based access control.
* **REQ-4**: Regularly update and patch the system to protect against vulnerabilities.

**5.4 Software Quality Attributes**

* **Usability**: Intuitive UI/UX design for all user classes.
* **Scalability**: System should scale horizontally to accommodate growth.
* **Maintainability**: Modular codebase with comprehensive documentation.
* **Reliability**: System uptime of 99.9% or higher.

**5.5 Business Rules**

* **BR-1**: Agencies can set custom pricing for their sub-accounts.
* **BR-2**: Application fees are charged per sale and on recurring sales via Stripe.
* **BR-3**: Vendors are responsible for their own compliance with selling regulations.

**6. Other Requirements**

* **Internationalization**: Support for multiple languages and regional settings.
* **Accessibility**: Compliance with WCAG 2.1 AA standards.
* **Legal Compliance**: Adherence to all applicable laws and regulations in operating regions.
* **Integration Capabilities**: Open APIs for third-party integrations.

**Appendix A: Glossary**

* **B2B2B**: Business-to-Business-to-Business.
* **SaaS**: Software as a Service.
* **Vendor**: A business or individual selling products/services on the platform.
* **Agency**: An organization managing multiple clients or projects.
* **Sub-account**: A user account under an agency with specific permissions.
* **Kanban Board**: A visual tool for project management and workflow.
* **Stripe Connect**: A service by Stripe for platforms to facilitate payments between third parties.
* **PCI DSS**: Payment Card Industry Data Security Standard.
* **GDPR**: General Data Protection Regulation.
* **WCAG**: Web Content Accessibility Guidelines.

**Appendix B: Analysis Models**

* **Use Case Diagrams**: Illustrate interactions between users and system features.
* **Activity Diagrams**: Show the flow of operations for key processes.
* **Entity-Relationship Diagrams**: Outline the database structure.

**Appendix C: To Be Determined List**

* **TBD-1**: Final selection of cloud service provider (AWS, Azure, etc.).
* **TBD-2**: Specific languages to support for internationalization.
* **TBD-3**: Additional third-party services for integration.
* **TBD-4**: Pricing models for agencies and vendors.