s+ softpositive

Need assistance defining the requirements for your product or solution? Contact:

Michael Tsapenko
CEO & Founder @ SoftPositive
m.tsapenko@softpositive.com

www | meet | linkedin

Software Requirements Specification (SRS) Template for MVP

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to detail the functional and non-functional requirements for the Minimum Viable Product (MVP) of [Product Name]. This document will serve as a foundational guide for the development team, stakeholders, and testers, ensuring a shared understanding of the system's capabilities and limitations.

1.2 Document Conventions

- **[Bracketed Text]:** Placeholder text that needs to be replaced with specific information.
- **Bold:** Key terms or headings.
- **Italics:** Emphasis or examples.

1.3 Intended Audience and Reading Suggestions

This document is intended for:

- **Product Owners/Stakeholders:** To validate that the requirements align with the business goals.
- **Development Team:** To understand what needs to be built.
- **Quality Assurance Team:** To develop test plans and cases.
- **Project Managers:** To plan and track project progress.

Readers are advised to first review the "Product Scope" to grasp the overall vision before delving into specific requirements.

1.4 Product Scope (for MVP)

The [Product Name] MVP aims to [briefly state the core problem it solves and the core value proposition]. It will focus on delivering [list 2-3 essential, core functionalities] to [target user group]. The goal is to validate [key assumption or hypothesis] and gather early user feedback to inform future development.

- **In Scope for MVP:** [List core features/functionalities that *will* be included in the MVP]
  - *Example: User registration and login.*
  - *Example: Basic profile management.*
  - *Example: Core feature A (e.g., creating a post, adding an item to cart).*
- **Out of Scope for MVP:** [List features/functionalities that are explicitly *not* included in the MVP but might be considered for future iterations]
  - *Example: Advanced search filters.*
  - *Example: Social sharing integrations.*
  - *Example: Offline mode.*

1.5 References

- [Link to any related business requirement documents, user stories, wireframes, or market research.]
- [Any relevant industry standards or regulations.]

2. Overall Description

2.1 Product Perspective

The [Product Name] MVP will be a [e.g., web application, mobile application (iOS/Android), desktop application, hybrid application]. It will operate as a [standalone system / part of a larger system (if applicable)].

- **Relationship to other systems (if any):** [Describe how this MVP interacts with existing systems or services.]
- **Dependencies on other systems/components:** [e.g., third-party APIs, external databases.]

2.2 Product Functions (High-Level Summary)

The MVP will provide the following high-level functions:

- [Function 1: e.g., Allow users to create and manage their accounts.]
- [Function 2: e.g., Enable users to perform the primary action (e.g., list an item, send a message).]
- [Function 3: e.g., Display relevant information to the user.]

2.3 User Characteristics

- **User Type 1: [e.g., End User / Customer]**
  - **Description:** [Brief description of this user type, their goals, and typical tasks.]
  - **Skill Level:** [e.g., Novice, Intermediate, Expert]
  - **Technical Proficiency:** [e.g., Low, Medium, High]
- **User Type 2: [e.g., Administrator / Content Creator]**
  - **Description:** [Brief description of this user type, their goals, and typical tasks.]
  - **Skill Level:** [e.g., Novice, Intermediate, Expert]
  - **Technical Proficiency:** [e.g., Low, Medium, High]

2.4 Constraints

- **Operational Constraints:**
  - **Operating Environment:** [e.g., Web browser compatibility (Chrome, Firefox, Safari, Edge), specific mobile OS versions (iOS 15+, Android 10+).]
  - **Deployment Environment:** [e.g., Cloud platform (AWS, GCP, Azure), specific server requirements.]
- **Technical Constraints:**
  - **Technology Stack:** [e.g., Frontend: React, Backend: Node.js, Database: PostgreSQL.]
  - **Integration Requirements:** [e.g., Must integrate with Stripe for payments, must use OAuth2 for authentication.]
  - **Performance Limits:** [e.g., Must handle X concurrent users, response time for critical operations must be under Y seconds.]
- **Legal/Regulatory Constraints:**
  - [e.g., GDPR compliance, HIPAA compliance, specific industry regulations.]
- **Resource Constraints:**
  - **Timeframe:** [e.g., MVP must be launched by [Date].]
  - **Budget:** [e.g., Development budget limited to [Amount].]

2.5 Assumptions and Dependencies

- **Assumptions:**
  - [Assumption 1: e.g., Users will have a stable internet connection.]

- [Assumption 2: e.g., The third-party API (e.g., payment gateway) will be available and perform as expected.]
- [Assumption 3: e.g., Specific data formats will be provided by external systems.]

- **Dependencies:**
  - [Dependency 1: e.g., Availability of a specific third-party library or service.]
  - [Dependency 2: e.g., Completion of a related backend service by another team.]

3. External Interface Requirements

3.1 User Interfaces

- **Look and Feel:** [Brief description of the desired aesthetic, e.g., "Clean, modern, and intuitive UI following Material Design principles."]
- **Screen Layouts:** [Refer to wireframes, mockups, or design prototypes if available.]
  - *Example: Login Page (refer to Wireframe_Login.png)*
  - *Example: Dashboard (refer to Mockup_Dashboard.figma)*
- **Navigation:** [Describe general navigation patterns, e.g., "Standard top navigation bar, persistent sidebar for key sections."]
- **Accessibility:** [e.g., "Must conform to WCAG 2.1 AA standards for accessibility."]
- **Responsiveness:** [e.g., "UI must be fully responsive and optimized for desktop, tablet, and mobile devices."]

3.2 Hardware Interfaces

- [e.g., "None for web application." or "Requires a microphone for voice input on mobile devices."]

3.3 Software Interfaces

- **Operating Systems:** [e.g., "Windows, macOS, Linux (for web browser access); iOS, Android (for mobile app)."]
- **Databases:** [e.g., "PostgreSQL, MongoDB, Firestore."]
- **APIs:**
  - **Internal APIs:** [e.g., "RESTful API for frontend-backend communication."]
  - **External APIs:** [e.g., "Stripe API for payment processing, Google Maps API for location services."]
- **Communication Protocols:** [e.g., "HTTPS, WebSockets."]

3.4 Communications Interfaces

- **Network Protocols:** [e.g., "TCP/IP, HTTP/HTTPS."]

- **Email/SMS Integration:** [e.g., "System must send transactional emails for account verification and password resets."]
- **Push Notifications:** [e.g., "Mobile app should support push notifications for critical alerts."]

4. System Features (Functional Requirements)

This section details the specific functional requirements of the MVP. Each feature should be described clearly, often using a "User Story" format.

4.1 Feature 1: User Authentication

- **Description:** This feature allows users to create an account and securely log in to the application.
- **Functional Requirements:**
  - **FR-1.1:** As a new user, I can register for an account using my email address and a password.
    - *Details: Email must be unique. Password must meet complexity requirements (e.g., min 8 chars, 1 uppercase, 1 number).*
  - **FR-1.2:** As a new user, I receive an email verification link after registration.
  - **FR-1.3:** As a registered user, I can log in using my verified email and password.
  - **FR-1.4:** As a registered user, I can request a password reset if I forget my password.
  - **FR-1.5:** As a registered user, I can log out of my account.
  - **FR-1.6:** The system must encrypt user passwords at rest.

4.2 Feature 2: [Core Feature Name]

- **Description:** [Briefly describe the purpose of this core feature.]
- **Functional Requirements:**
  - **FR-2.1:** As a [User Type], I can [Action] [Object] in order to [Benefit].
    - *Details: [Specific conditions, inputs, outputs, data validations, error handling scenarios related to this requirement].*
  - **FR-2.2:** As a [User Type], I can [Action] [Object] in order to [Benefit].
  - **FR-2.3:** [Add more specific requirements for this feature.]

4.3 Feature 3: [Another Core Feature Name]

- **Description:** [Briefly describe the purpose of this core feature.]
- **Functional Requirements:**
  - **FR-3.1:** As a [User Type], I can [Action] [Object] in order to [Benefit].
  - **FR-3.2:** [Add more specific requirements for this feature.]

5. Non-functional Requirements

5.1 Performance Requirements

- **Response Time:**
  - **NFR-5.1.1:** The system shall load the [e.g., Dashboard] within [e.g., 3 seconds] for [e.g., 90%] of users under normal load.
  - **NFR-5.1.2:** Critical API calls (e.g., [e.g., submitting a form]) shall respond within [e.g., 1 second] [e.g., 95%] of the time.
- **Scalability:**
  - **NFR-5.1.3:** The system shall support up to [e.g., 1,000 concurrent users] without significant degradation in performance.
  - **NFR-5.1.4:** The system should be able to scale to [e.g., 10,000 users] within [e.g., 6 months] with minimal architectural changes.
- **Throughput:**
  - **NFR-5.1.5:** The system shall process [e.g., 100 transactions] per second.

5.2 Security Requirements

- **Authentication:**
  - **NFR-5.2.1:** User authentication shall be secure, using industry-standard protocols (e.g., OAuth2, JWT).
- **Authorization:**
  - **NFR-5.2.2:** The system shall enforce role-based access control, ensuring users can only access authorized features and data.
- **Data Protection:**
  - **NFR-5.2.3:** All sensitive data (e.g., passwords, personal identifiable information) shall be encrypted at rest and in transit.
- **Vulnerability:**
  - **NFR-5.2.4:** The system shall be protected against common web vulnerabilities (e.g., OWASP Top 10).
- **Auditing:**
  - **NFR-5.2.5:** Critical user actions (e.g., data modification, login attempts) shall be logged for auditing purposes.

5.3 Reliability Requirements

- **Availability:**
  - **NFR-5.3.1:** The system shall be available [e.g., 99.5%] of the time (excluding scheduled

maintenance).

- **Recoverability:**
  - **NFR-5.3.2:** In the event of a system failure, the system shall recover to an operational state within [e.g., 30 minutes] with minimal data loss.
- **Error Handling:**
  - **NFR-5.3.3:** The system shall provide informative and user-friendly error messages, without exposing sensitive system details.

5.4 Maintainability Requirements

- **Modularity:**
  - **NFR-5.4.1:** The system architecture shall be modular to allow for independent development and deployment of components.
- **Testability:**
  - **NFR-5.4.2:** The codebase shall be designed to facilitate automated unit, integration, and end-to-end testing.
- **Documentation:**
  - **NFR-5.4.3:** Code shall be well-commented, and API endpoints shall be documented (e.g., using OpenAPI/Swagger).

5.5 Portability Requirements

- **NFR-5.5.1:** The application shall be deployable on different cloud providers (e.g., AWS, GCP) with minimal configuration changes. (If applicable)
- **NFR-5.5.2:** The mobile application shall be compatible with both iOS and Android platforms. (If applicable)

5.6 Usability Requirements

- **Ease of Use:**
  - **NFR-5.6.1:** The user interface shall be intuitive and easy to navigate for target users, requiring minimal training.
- **Learnability:**
  - **NFR-5.6.2:** New users should be able to complete core tasks within [e.g., 5 minutes] of first use without external assistance.
- **Efficiency:**
  - **NFR-5.6.3:** Experienced users should be able to perform common tasks quickly and efficiently.

6. Other Requirements

6.1 Legal and Compliance Requirements

- **NFR-6.1.1:** The system shall comply with all relevant data privacy regulations (e.g., GDPR, CCPA).
- **NFR-6.1.2:** Any third-party libraries or services used shall have appropriate licenses.

6.2 Operational Requirements

- **NFR-6.2.1:** The system shall provide logging and monitoring capabilities for operational health and troubleshooting.
- **NFR-6.2.2:** Automated backups of the database shall be performed daily.

7. Glossary

- **MVP:** Minimum Viable Product – A version of a new product which allows a team to collect the maximum amount of validated learning about customers with the least effort.
- **SRS:** Software Requirements Specification – A document that describes the intended purpose and environment for software.
- **UI:** User Interface – The means by which the user and a computer system interact.
- **API:** Application Programming Interface – A set of defined methods of communication between various software components.
- **[Add other specific terms relevant to your project]**