**CS330 Software Engineering**

**Software Requirements Specification (SRS) Template**

Items that are intended to stay in as part of your document are in **bold**; explanatory comments are in *italic* text. Plain text is used where you might insert wording about your project.

The document in this file is an annotated outline for specifying software requirements, adapted from the IEEE Guide to Software Requirements Specifications (Std 830-1993).

Tailor this to your needs, removing explanatory comments as you go along. Where you decide to omit a section, keep the header, but insert a comment saying why you omit the data.

Senior Project - USACE DSMC Customer Survey App

The Gremlins

Nate Blakely

blakely6@marshall.edu

Ember Adkins

adkins1487@marshall.edu

Gabriel Smith

smith2834@marshall.edu

**Software Requirements Specification**

**Document**

**Version: (n)** **Date: (01/25/2024)**

**Table of Contents**

**1. Introduction 5**

*1.1 Purpose*

*1.2 Scope*

*1.3 Definitions, Acronyms, and Abbreviations*

*1.4 References*

*1.5 Overview*

**2. The Overall Description**

*2.1 Product Perspective*

2.1.1 System Interfaces

2.1.2 Interfaces

2.1.3 Hardware Interfaces

2.1.4 Software Interfaces

2.1.5 Communications Interfaces

2.1.6 Memory Constraints

2.1.7 Operations

2.1.8 Site Adaptation Requirements

*2.2 Product Functions*

*2.3 User Characteristics*

*2.4 Constraints*

*2.5 Assumptions and Dependencies*

*2.6 Apportioning of Requirements*

**3. Specific Requirements**

*3.1 External interfaces*

*3.2 Functions*

*3.3 Performance Requirements*

*3.4 Logical Database Requirements*

*3.5 Design Constraints*

3.5.1 Standards Compliance

*3.6 Software System Attributes*

3.6.1 Reliability

3.6.2 Availability

3.6.3 Security

3.6.4 Maintainability

3.6.5 Portability

*3.7 Organizing the Specific Requirements*

3.7.1 System Mode

3.7.2 User Class

3.7.3 Objects

3.7.4 Feature

3.7.5 Stimulus

3.7.6 Response

3.7.7 Functional Hierarchy

*3.8 Additional Comments*

**4. Change Management Process**

**5. Document Approvals**

**6. Supporting Information**

# **1. Introduction**

Project Name

Dam Safety Survey Web App

1.2.2 Project Purpose

The purpose of the Dam Safety Survey Web App is to provide a centralized platform for the comprehensive assessment and monitoring of dam safety. The web application facilitates the collection of crucial data related to dam structures, safety parameters, and incidents, empowering dam owners, inspectors, and administrators with the tools needed for informed decision-making and risk mitigation.

1.2.3 Project Objectives

The key objectives of the project include:

* Enhancing the efficiency of dam information management.
* Facilitating risk assessment through structured surveys.
* Improving incident reporting and response mechanisms.
* Providing insightful data visualization for effective decision-making.
* Enabling custom reporting for in-depth analysis.
* Integrating seamlessly with external systems, like email notification services.

1.3 Target Audience

The Dam Safety Survey Web App caters to a diverse audience involved in dam management and safety. The primary stakeholders include dam owners responsible for registering and updating information, inspectors engaged in survey responses and incident reporting, and administrators overseeing system management and survey creation.

1.4 Document Scope

This Software Requirements Specification (SRS) document outlines the functional and non-functional requirements of the Dam Safety Survey Web App. It serves as a comprehensive guide for developers, testers, and stakeholders involved in the project, ensuring a common understanding of the system's features, capabilities, and constraints.

1.5 Document Structure

The document is organized into sections that cover different aspects of the Dam Safety Survey Web App, including an overview, functional requirements, external interfaces, user interface elements, and more. Each section provides detailed insights into specific aspects of the project to guide the development team in creating a robust and effective solution.

## **1.1 Purpose**

1. **Customer Satisfaction Measurement:**
   * Surveys help measure customer satisfaction levels by collecting feedback on various aspects of products, services, or the overall customer experience.
   * Companies can use satisfaction scores to identify areas of improvement or to acknowledge successful aspects of their business.
2. **Product and Service Improvement:**
   * Customer surveys provide valuable insights into how customers perceive products and services.
   * Feedback can be used to identify areas for improvement or to enhance existing features, ensuring that offerings meet or exceed customer expectations.
3. **Identifying Issues and Challenges:**
   * Surveys help in identifying any issues or challenges customers may be facing with products, services, or the overall customer journey.
   * Companies can address these issues promptly, preventing potential negative impacts on customer satisfaction and loyalty.

## **1.2 Scope**

The purpose of this project is to develop a customer survey app tailored for a USACE DSMC. The app will enable stakeholders, including dam safety engineers, environmental scientists, local communities, and other relevant parties, to provide valuable feedback on dam safety modifications and related activities.

The primary objectives of the customer survey app are as follows:

* Facilitate feedback collection from various stakeholders involved in dam safety projects.
* Gather insights on the environmental, engineering, and social aspects of dam safety modifications.
* Improve communication and collaboration between the dam safety company and stakeholders.
* Enhance overall customer satisfaction and engagement.

## **1.5 Overview**

1.1 Purpose

The Dam Safety Survey Web App aims to provide a comprehensive platform for assessing and monitoring the safety of dams. It facilitates data collection, analysis, and reporting related to dam structures, safety parameters, and incidents. The purpose is to enhance the overall safety management of dams by enabling dam owners, inspectors, and administrators to collaborate effectively in identifying and addressing potential risks.

1.2 Scope

The scope of the Dam Safety Survey Web App includes:

Dam Information Management:

Registration and maintenance of essential details about dams.

Tracking structural information, reservoir data, and monitoring instruments.

Survey Module:

Creation, distribution, and collection of dam safety surveys.

Dynamic survey capabilities to adapt to varying dam types and risk factors.

Incident Reporting:

1.3 Objectives

The key objectives of the Dam Safety Survey Web App are as follows:

Efficient Data Collection:

Streamline the process of gathering comprehensive information about dams and their safety parameters.

Risk Assessment:

Facilitate the assessment of dam safety risks through structured surveys and data analysis.

Custom Reporting:

Enable users to generate custom reports for in-depth analysis and decision-making.

Integration with External Systems:

Integrate seamlessly with email notification systems for timely alerts about survey requests.

1.4 Stakeholders

The primary stakeholders of the Dam Safety Survey Web App include:

Dam Owners:

Responsible for registering and updating information about their dams.

Inspectors:

Engage in survey responses and incident reporting, contributing to the overall dam safety assessment.

Administrators:

Manage survey creation, user roles, and overall system administration.

1.5 Assumptions and Constraints

Assumptions:

Users have access to a reliable internet connection for seamless web application usage.

Dam owners and inspectors will provide accurate and timely information.

Constraints:

The web app is constrained by the capabilities and limitations of external systems, such as the database’s storage and the mail notification service’s connectivity.

# **2. The Overall Description**

This SRS document displays the components of our project. Our goal is to develop a web based application that allows customers to freely evaluate USACE.

## **2.1 Product Perspective**

External Systems

The Dam Safety Survey Web App interacts with the following external systems:

2.1.2 Database Interface

The Dam Safety Survey Web App interfaces with a chosen Database Management System (DBMS) to store and retrieve data. The database contains information related to dams, survey responses, user profiles, and system configurations.

2.2 User Interfaces

2.2.1 Web Application Interface

The primary user interface is a web-based application accessible through standard web browsers. The interface includes components for user registration, login, navigation menus, forms for dam information and incident reporting, dashboards for data visualization, and modules for survey creation and response.

2.4 Software Interfaces

The web app relies on several software components, including:

Web Servers:

The application is hosted on web servers (e.g., Apache, Nginx) to handle user requests and deliver content.

Database Management System:

The chosen DBMS (e.g., MySQL, PostgreSQL) is used for data storage and retrieval.

2.5 Communication Interfaces

The Dam Safety Survey Web App communicates with external systems and services through secure communication protocols, including HTTPS for web-based interactions and industry-standard protocols for API integrations.

### **2.1.1 System Interfaces**

External Systems

### Notification Services

### Description: Integration with external notification services to provide real-time alerts for critical dam safety issues.

### Interactions:

### Transmission of incident alerts and notifications.

### Integration with notification APIs for seamless communication.

### External Authentication Providers

### Description: Integration with OAuth 2.0 for secure user authentication through third-party providers.

### Interactions:

### User authentication and authorization through external authentication providers.

### Secure retrieval of user profile information.

### Database Interface

### Database Management System (DBMS)

### Description: The Dam Safety Survey Web App interfaces with a chosen DBMS to store and retrieve data.

### Interactions:

### Storage of dam information, survey responses, incident reports, user profiles, and system configurations.

### Retrieval of data for presentation within the application.

### User Interfaces

### Web Application Interface

### Description: The primary user interface is a web-based application accessible through standard web browsers.

### Interactions:

### User registration and login.

### Navigation through menus and links.

### Data input through forms for dam information and incident reporting.

### Visualization of data through dashboards and charts.

### 3.4 Hardware Interfaces

### Description: The Dam Safety Survey Web App is designed to be platform-independent, requiring only a standard web browser to access its features.

### Interactions:

### Compatible with various computing devices, including desktops, laptops, smartphones, and tablets.

### No specific hardware dependencies.

### Software Interfaces

### Web Servers

### Description: The application is hosted on web servers (e.g., Apache, Nginx) to handle user requests and deliver content.

### Interactions:

### Processing and serving of web pages.

### Handling user requests and managing application logic.

### Database Management System

### Description: The chosen DBMS (e.g., MySQL, PostgreSQL) is used for data storage and retrieval.

### Interactions:

### Execution of SQL queries for data manipulation.

### Ensuring data integrity and security.

### Programming Languages and Frameworks

### Description: The web app is developed using [programming languages] and [frameworks] to ensure efficient functionality and maintainability.

### Interactions:

### Execution of application code to implement features.

### Integration with external libraries and frameworks.

### Communication Interfaces

### Description: The Dam Safety Survey Web App communicates with external systems and services through secure communication protocols.

### Interactions:

### Use of HTTPS for secure web-based interactions.

### Adherence to industry-standard protocols for API integrations.

### **2.1.2 Interfaces**

External Systems

### Notification Services

### Description: Integration with external notification services to provide real-time alerts for critical dam safety issues.

### Interactions:

### Transmission of incident alerts and notifications.

### Integration with notification APIs for seamless communication.

### External Authentication Providers

### Description: Integration with OAuth 2.0 for secure user authentication through third-party providers.

### Interactions:

### User authentication and authorization through external authentication providers.

### Secure retrieval of user profile information.

### Database Interface

### Database Management System (DBMS)

### Description: The Dam Safety Survey Web App interfaces with a chosen DBMS to store and retrieve data.

### Interactions:

### Storage of dam information, survey responses, incident reports, user profiles, and system configurations.

### Retrieval of data for presentation within the application.

### User Interfaces

### Web Application Interface

### Description: The primary user interface is a web-based application accessible through standard web browsers.

### Interactions:

### User registration and login.

### Navigation through menus and links.

### Data input through forms for dam information and incident reporting.

### Visualization of data through dashboards and charts.

### 3.4 Hardware Interfaces

### Description: The Dam Safety Survey Web App is designed to be platform-independent, requiring only a standard web browser to access its features.

### Interactions:

### Compatible with various computing devices, including desktops, laptops, smartphones, and tablets.

### No specific hardware dependencies.

### Software Interfaces

### Web Servers

### Description: The application is hosted on web servers (e.g., Apache, Nginx) to handle user requests and deliver content.

### Interactions:

### Processing and serving of web pages.

### Handling user requests and managing application logic.

### Database Management System

### Description: The chosen DBMS, mySQL is used for data storage and retrieval.

### Interactions:

### Execution of SQL queries for data manipulation.

### Ensuring data integrity and security.

### Programming Languages and Frameworks

### Description: The web app is developed using HTML, php and node.js to ensure efficient functionality and maintainability.

### Interactions:

### Execution of application code to implement features.

### Integration with external libraries and frameworks.

### Communication Interfaces

### Description: The Dam Safety Survey Web App communicates with external systems and services through secure communication protocols.

### Interactions:

### Use of HTTPS for secure web-based interactions.

### Adherence to industry-standard protocols for API integrations.

### **2.1.3 Hardware Interfaces**

### Desktop Computers

### Description: The web application should be compatible with standard desktop computers, including both Windows and macOS operating systems.

### Requirements:

### Modern web browser support (Google Chrome, Mozilla Firefox, Microsoft Edge, Safari).

### Sufficient processing power for smooth web application performance.

### Laptops

### Description: The web app should function seamlessly on various laptop devices running supported web browsers.

### Requirements:

### Compatibility with laptops running Windows, macOS, or Linux.

### Responsive design for different screen sizes.

### Mobile Devices

### Description: The web application employs a mobile responsive design to ensure optimal user experience on smartphones and tablets.

### Requirements:

### Compatibility with major mobile platforms (iOS, Android).

### Touch-friendly interactions for mobile users.

2.1.4 Software Interfaces

### Web Servers Interface

### Server Software

### Description: The Dam Safety Survey Web App is hosted on web servers, and it interacts with server software to handle user requests and deliver content.

### Requirements:

### Compatibility with standard web server software such as Apache, Nginx, or equivalent.

### Support for HTTPS to ensure secure communication.

### Server-Side Scripting

### Description: The web application utilizes server-side scripting for dynamic content generation and processing of user requests.

### Requirements:

### Support for server-side scripting languages like PHP, Node.js, or equivalent.

### Efficient handling of dynamic content generation.

### Database Management System Interface

### Database Software

### Description: The Dam Safety Survey Web App interfaces with a Database Management System (DBMS) to store and retrieve data.

### Requirements:

### Compatibility with the selected DBMS (e.g., MySQL, PostgreSQL).

### Support for structured data storage and retrieval.

### Structured Query Language (SQL)

### Description: The web application interacts with the DBMS using SQL queries for data manipulation and retrieval.

### Requirements:

### Support for standard SQL syntax.

### Efficient execution of complex queries for data analysis.

### Programming Languages and Frameworks Interface

### Front-End Frameworks

### Description: The user interface of the web app is developed using front-end frameworks to ensure a responsive and interactive user experience.

### Requirements:

### Compatibility with front-end frameworks such as React, Angular, or equivalent.

### Responsive design for various screen sizes.

### Back-End Frameworks

### Description: The back-end of the web app is developed using server-side frameworks to handle business logic and data processing.

### Requirements:

### Compatibility with back-end frameworks like Express, Django, or equivalent.

### Efficient handling of server-side logic and routing.

### Programming Languages

### Description: The application code is written in specific programming languages to implement features and functionality.

### Requirements:

### Compatibility with programming languages such as JavaScript, Python, or equivalent.

### Maintainability and scalability of the codebase.

### External APIs

### Notification API

### Description: Integration with external notification services is achieved through APIs to enable real-time alerts.

### Requirements:

### Compatibility with notification service API specifications.

### Secure transmission of incident alerts.

### 2.1.5 Communications Interfaces

**Web-Based Communication**

**HTTPS**

* Description: The Dam Safety Survey Web App employs HTTPS for secure communication between clients and servers.
* Requirements:
* Implementation of SSL/TLS encryption to secure data in transit.
* Configuration of web servers to support HTTPS.

**WebSocket**

* Description: WebSocket communication may be utilized for real-time interactions and notifications.
* Requirements:
* Implementation of WebSocket support in both client and server components.
* Handling real-time updates and notifications through WebSocket connections.

**API Communication**

**RESTful APIs**

* Description: RESTful APIs are employed for communication with external systems, including notification services and authentication providers.
* Requirements:
* Definition of clear and standardized RESTful API endpoints.
* Implementation of CRUD (Create, Read, Update, Delete) operations for data exchange.

**Authentication Protocols**

* Description: OAuth 2.0 is used for secure user authentication through third-party providers.
* Requirements:
* Implementation of OAuth 2.0 authentication flows.
* Secure retrieval of user profile information through authenticated API calls.

**Email Communication**

**SMTP**

* Description: The web app utilizes the Simple Mail Transfer Protocol (SMTP) for sending email notifications.
* Requirements:
* Configuration of email server details for outgoing emails.
* Secure transmission of email data.

**External System Communication**

**Data Format Standards**

* Description: Data exchanged with external systems, including email notification services, adheres to standardized formats.
* Requirements:
* Consistent use of JSON or XML for data serialization.
* Compliance with external system data format standards.

**API Key Authentication**

* Description: Authentication with external systems is achieved using API keys where applicable.
* Requirements:
* Secure management and transmission of API keys.
* Implementation of API key authentication mechanisms*.*

### **2.1.6 Memory Constraints**

**Client-Side Memory**

**Browser Memory**

* Description: The web application relies on client-side resources for rendering and executing functionality within web browsers.
* Constraints:
* The web application should be optimized for memory efficiency to ensure smooth performance on client devices.
* Consideration should be given to varying memory capacities of different client devices.

**Server-Side Memory**

**Web Server Memory**

* Description: The web server hosting the Dam Safety Survey Web App requires memory resources to handle concurrent user requests and process server-side logic.
* Constraints:
* Adequate server memory should be allocated to ensure efficient handling of web server processes.
* Monitoring and optimization mechanisms should be in place to prevent memory leaks.

**Database Server Memory**

* Description: The Database Management System (DBMS) used by the web app requires memory for data storage and retrieval.
* Constraints:
* The database server should have sufficient memory to accommodate the expected volume of data.
* Monitoring tools should be employed to track and optimize database memory usage.

**Network Bandwidth**

**Data Transfer**

* Description: The Dam Safety Survey Web App communicates with external systems and clients, requiring network bandwidth for data transfer.
* Constraints:
* Efficient data compression and transmission techniques should be employed to minimize bandwidth usage.
* Consideration for users with varying network speeds and limitations should be taken into account.

**System Logs**

**Logging**

* Description: The web app generates logs for tracking system events and errors.
* Constraints:
* Log files should be managed to prevent excessive growth and potential impact on system memory.
* Implement log rotation and archival mechanisms to ensure optimal use of storage.

**Caching**

**Data Caching**

* Description: Caching mechanisms are used to improve performance by storing frequently accessed data in memory.
* Constraints:
* Set cache size limits to prevent excessive memory usage.
* Implement cache expiration policies to maintain data accuracy.

### **2.1.7 Operations**

The Dam Safety Survey Web App offers a range of operations to facilitate efficient and effective survey management, incident reporting, and data analysis. Users interact with the application to perform the following key operations:

**Survey Management Operations**

**Create a New Survey**

* Description: Users can initiate the creation of a new survey to assess dam safety.
* Requirements:
* Access to the "Create Survey" interface.
* Input fields for survey details, such as title, description, and location.

**Edit Survey Details**

* Description: Users with appropriate permissions can modify details of an existing survey.
* Requirements:
* Access to the "Edit Survey" interface.
* Ability to update survey information, including title, description, and associated parameters.

**View Survey List**

* Description: Users can access a list of all surveys for quick reference and navigation.
* Requirements:
* Display of survey titles with links to individual survey details.

### **2.1.8 Site Adaptation Requirements**

The Dam Safety Survey Web App is designed to be adaptable to different environments and configurations. The following requirements detail the necessary adaptations for deploying, integrating, and customizing the application according to specific site needs.

**Deployment Requirements**

**Deployment Platforms**

* Requirement:
* The web application should be deployable on common web hosting platforms, including but not limited to Apache, Nginx, or other compatible servers.

**Supported Operating Systems**

* Requirement:
* The web application should be compatible with various operating systems, including Linux, Windows, and macOS.

**Database Management System Compatibility**

* Requirement:
* The application should support integration with different Database Management Systems (DBMS) such as MySQL, PostgreSQL, or SQLite.

## **2.2 Product Functions**

1. Sending surveys to the relevant users within a timely manner
2. Recording user answers and storing answers within a database
3. Presenting data obtained from surveys in a suitable manner.
4. Send notifications to users via outlook to tell them of a survey or an incomplete survey once 15 days have passed

## **2.3 User Characteristics**

**Application and Database Administration** – These users will be able to modify survey content, modify domain list content, create new surveys, define evaluation periods, notify respondents a survey has been requested

**Surveyors** – These users will be DSMC employees who are able to add respondents to be surveyed and select the mission/program/project and tasks that are to be evaluated

**Respondents** – These users will be able to link to and complete a requested survey

## **2.4 Constraints**

***Data Storage Requirements:***

The app needs to store survey data, including dam details, inspection results, and any other relevant information.

The volume of data collected per survey and the number of surveys conducted will impact the storage requirements.

Consideration should be given to efficient data storage methods and database optimization to manage data growth.

***Background Processes:***

**Data Synchronization**

Description

The web application may implement background processes to synchronize data between the application's database and external systems, such as databases.

**Requirements**:

Periodic synchronization to ensure that the application's data is consistent with external systems.

Logging and reporting mechanisms to track synchronization events and identify any discrepancies.*.*

User Interface Elements:

**General Layout and Navigation**

* **Header**

Description: The top section of each page providing navigation and key user actions.

Elements:

Logo linking to the homepage.

Navigation menu with links to key sections (Dashboard, Dam Information, Surveys, Incidents).

* **Sidebar**

Description: A collapsible sidebar for additional navigation options and quick access to features.

Elements:

User profile information.

Role-specific menu options (e.g., Dam Owner, Inspector, Administrator).

* **Footer**

Description: The bottom section of each page contains additional links and information.

Elements:

Copyright information.

Links to terms of service, privacy policy, and contact information.

* **Dam Registration Form**

Description: The form used by dam owners to input and update information about their dams.

Elements:

Fields for dam name, location, type, construction material, height, length, etc.

Save and Cancel buttons.

* **Dam Details View**

Description: The interface displaying a detailed view of a specific dam's information.

Elements:

Edit and Delete buttons for dam owners.

Navigation links to related surveys and incidents.

**Survey Module**

**Survey Creation Interface**

* Description: The interface used by administrators to create and customize dam safety surveys.
* Elements:
* Drag-and-drop question builder.
* Options for multiple-choice, text, and numeric responses.
* Save and Publish buttons.

**Survey Distribution Page**

Description: The page for distributing surveys to relevant dam owners and inspectors.

Elements:

List of available surveys.

Selectable recipients with role-based filtering.

Schedule and Send buttons.

**Survey Response Form**

* Description: The form for users to respond to surveys.
* Elements:
* Dynamic rendering of survey questions.
* Progress indicator.
* Submit and Save as Draft buttons.

**Incident Reporting**

* Incident Reporting Form
* Description: The form for users to report incidents related to dam safety.
* Elements:
* Incident details, date, time, severity.
* Option to attach media files.
* Submit and Cancel buttons.

**Incident Dashboard**

* Description: The dashboard summarizing reported incidents.
* Elements:
* Filter options based on incident parameters.
* Table or list displaying incident summaries.
* View details and download report buttons.

## **2.5 Assumptions and Dependencies**

**Assumptions**

The following assumptions have been made during the planning and development of the Dam Safety Survey Web App:

**User Internet Connectivity:**

Users are assumed to have reliable internet connectivity to access and interact with the web application.

**Browser Compatibility:**

The web application is designed to be compatible with modern web browsers, including Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.

**Data Accuracy:**

Assumption is made that the data provided by users during surveys and incident reporting is accurate and truthful.

**User Authentication Validity:**

External authentication providers, such as OAuth 2.0, are assumed to provide valid and secure user authentication.

**Hardware Independence:**

The web application is assumed to be platform-independent and compatible with a variety of computing devices.

**Dependencies**

The development and operation of the Dam Safety Survey Web App depend on the following external factors:

**External Authentication Providers:**

The functioning of user authentication through external providers (OAuth 2.0) depends on the stability and reliability of those services.

**Notification Services:**

Real-time alerting and notifications depend on the availability and proper functioning of external notification services.

**Database Management System:**

The web application relies on a chosen Database Management System (DBMS), and its performance depends on the efficiency and reliability of the selected DBMS.

**Web Servers:**

The hosting and deployment of the web application depend on the stability and proper configuration of web servers (e.g., Apache, Nginx).

**Network Infrastructure:**

The reliable operation of the web application depends on the stability and performance of the underlying network infrastructure.

**Third-Party Libraries and Frameworks:**

The use of third-party libraries and frameworks introduces dependencies on their maintenance, updates, and compatibility with the application.

## **2.6 Apportioning of Requirements.**

**Apportioning of Requirements**

The Dam Safety Survey Web App requirements will be apportioned across different releases or phases to ensure a structured and prioritized development approach. The apportioning is based on factors such as criticality, dependencies, and strategic objectives. The following outlines the planned distribution of requirements:

**Initial Release (Version 1.0)**

Core Functionality

* Requirements:
* Deploy basic survey creation and management functionalities.
* Implement incident reporting features.
* Establish user authentication and authorization mechanisms.

**Minimum Viable Product (MVP)**

* Requirements:
* Include essential survey data visualization and reporting capabilities.
* Integrate with a primary Database Management System (DBMS).
* Ensure basic customization options for users.

**Subsequent Releases**

**Advanced Reporting (Version 1.1)**

* Requirements:
* Enhance reporting capabilities with advanced filters and data analysis tools.

**User Customization (Version 1.2)**

* Requirements:
* Provide advanced customization options for survey forms and report generation.
* Implement theming options for user interface personalization.

**Integration Enhancements (Version 1.3)**

* Requirements:
* Expand integration capabilities with third-party services for notifications and monitoring.
* Introduce support for additional Database Management Systems.

# **3. Specific Requirements**

## The following specific requirements outline the detailed functionalities, features, and constraints of the Dam Safety Survey Web App. These requirements are organized into categories for clarity and comprehensive coverage.

## Survey Management

## Create New Survey

## Description:

## Users should be able to create a new dam safety survey.

## Requirements:

## The system shall provide a user interface for creating surveys.

## Survey creation should include fields for title, description, and location.

## Surveys must be associated with a responsible user or organization.

## 8.1.2 Edit Survey Details

## Description:

## Users with appropriate permissions should be able to modify survey details.

## Requirements:

## The system shall provide an interface for editing existing survey details.

## Authorized users should be able to update survey information, including title, description, and parameters.

## 

## **3.1 External Interfaces**

The Dam Safety Survey Web App interacts with various external entities to enable seamless data exchange, integration, and collaboration. This section outlines the key external interfaces that the system interfaces with:

**User Interfaces**

**Web Browser Interface**

* Description:

* The primary interface for users to interact with the Dam Safety Survey Web App is through web browsers.
* Requirements:
* The web interface shall be compatible with major web browsers, including but not limited to Google Chrome, Mozilla Firefox, and Microsoft Edge.
* The interface should provide a responsive design for optimal user experience across different devices.

**Third-Party Services API**

* Description:
* The system integrates with third-party services for additional functionalities such as notifications and monitoring.
* Requirements:
* The Dam Safety Survey Web App shall support integration with third-party APIs for services like email notifications and system monitoring.
* The APIs should be secure and adhere to industry standards for data exchange.

**Database Management System (DBMS)**

**Database Integration**

* Description:
* The Dam Safety Survey Web App interacts with a Database Management System (DBMS) for data storage and retrieval.
* Requirements:
* The system shall support integration with popular DBMS platforms, including MySQL, PostgreSQL, or SQLite.
* Database connections should be securely configured, and data integrity should be maintained.

**Authentication Providers**

**External Authentication Providers**

* Description:
* User authentication is facilitated through external authentication providers.
* Requirements:
* The system shall integrate with external authentication providers, such as OAuth 2.0.
* Authentication processes should be secure, and user credentials must be handled with care.

## **3.2 Functions**

* Web Application Interface

Description: The primary interface for users to interact with the Dam Safety Survey Web App.

Features:

User authentication and login.

Intuitive navigation for dam registration, survey responses, and incident reporting.

Role-specific dashboards for administrators, respondents, and surveyors.

* Survey API

Description: Facilitate communication between the web app and external systems for survey-related functionalities.

Features:

Endpoints for survey creation, distribution, and response collection.

Data validation for survey submissions.

Database Interface

* Database Management System (DBMS)

Description: Interface with the chosen database system to store and retrieve data.

Features:

Support for structured storage of dam information, survey responses, and incident reports.

Ensure data integrity and security.

Notification Systems

* Email Notification Service

Description: Enable the web app to send email notifications for account activation, survey invitations, and critical incident alerts.

Features:

Integration with a third-party email service.

Template-based email generation.

## **3.3 Performance Requirements**

Email notifications will be sent out within 15 minutes of when a survey starts along with sending reminders of unfinished surveys once 15 days have passed.

## **3.4 Logical Database Requirements**

The survey table houses the ID for the survey, the ID of the Admin user who created the survey, the survey’s title, and a description of the survey.

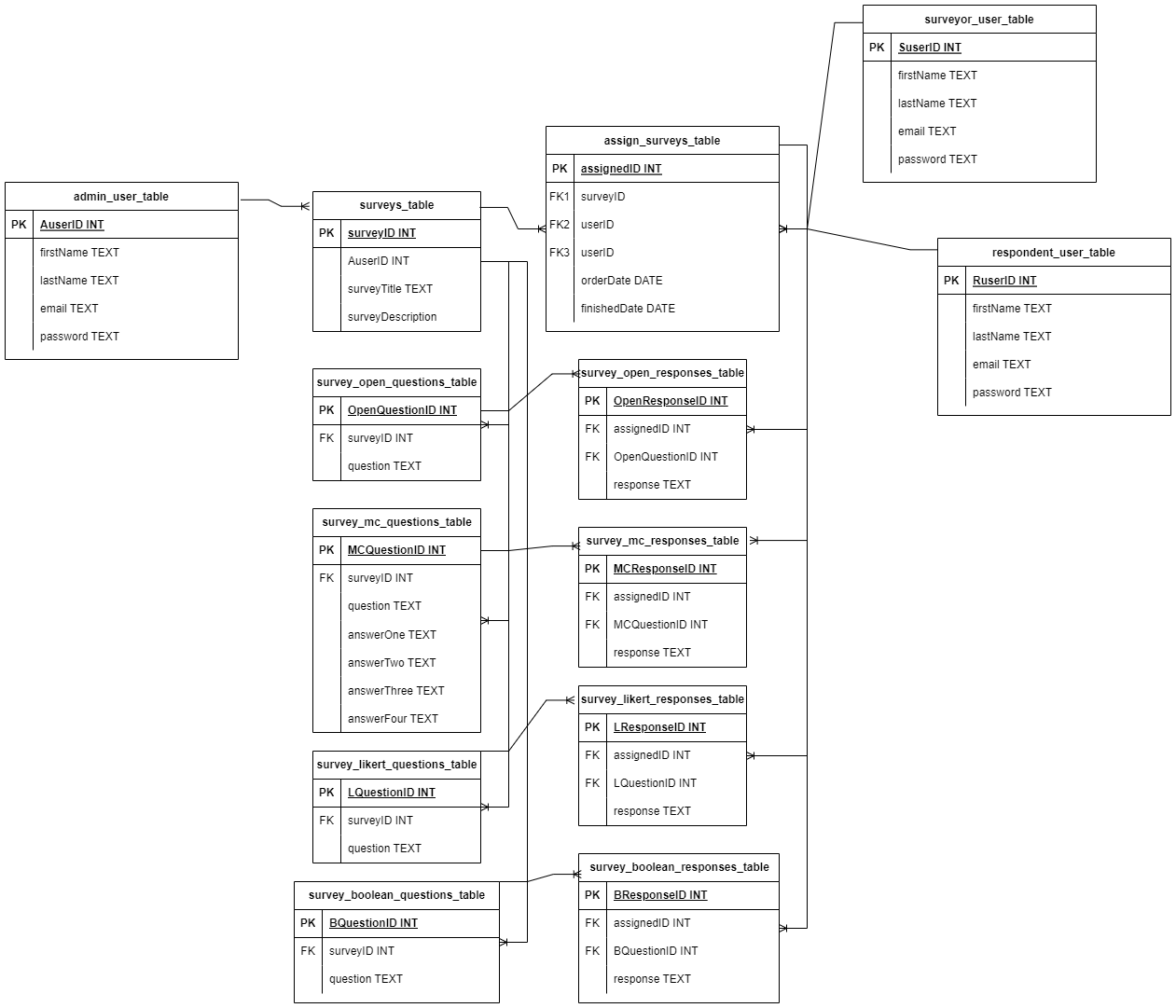
The assign surveys table contains the ID for the specific assignment, the surveyID of the survey being assigned, the userID of the surveyor user that assigned the survey, the userID for the respondent that has been assigned the survey, the date the survey was set for, and the date the respondent completed the survey.

The survey\_open\_questions\_table contains the ID for the question, the ID of the survey, and the text of the question. The survey\_open\_responses contains the id of the response, the assignedID of the survey instance being filled out, and the response.

The survey\_mc\_questions\_table contains the ID for the question, the ID of the survey, the text of the question, as well as the text for each of the answer choices. The survey\_mc\_responses\_table contains the ID of the response, the assignedID of the survey instance being filled out, the ID of the question, and the response.

The survey\_likert\_questions\_table contains the ID for each question, the ID of the survey it is associated with, and the question. The survey\_likert\_response\_table contains the ID of the response, the assignedID of the survey instance, the ID of the question being answered, and the response.

The survey\_boolean\_questions\_table contains the ID for each question, the ID of the survey it is associated with, and the question. The survey\_boolean\_response\_table contains the ID of the response, the assignedID of the survey instance, the ID of the question being answered, and the response.

******

## **3.6 Software System Attributes**

The Dam Safety Survey Web App is designed to exhibit specific attributes that contribute to its overall quality, performance, and user experience.

**Reliability**

**Availability**

* Requirement:
* The system shall strive for 99.9% availability, ensuring users can access the application consistently.

**Fault Tolerance**

* Requirement:
* The system shall be designed to tolerate faults and recover gracefully to minimize downtime.

**Performance**

**Response Tim**e

* Requirement:
* The web interfaces of the application shall have a response time of less than 2 seconds for standard operations.

**Scalability**

* Requirement:
* The application architecture shall support scalability to handle increased loads as the user base grows.

**Usability**

**User Interface Design**

* Requirement:
* The web interfaces shall follow best practices for user interface design, providing an intuitive and user-friendly experience.

**Accessibility**

* Requirement:
* The application shall be accessible to users with diverse needs, complying with Web Content Accessibility Guidelines (WCAG).

**Maintainability**

**Code Maintainability**

* Requirement:
* The source code shall be well-documented, following coding standards, to ensure ease of maintenance and future enhancements.

**Modularity**

* Requirement:
* The system architecture shall be modular, allowing for independent updates and changes to specific components.

**Compatibility**

**Browser Compatibility**

* Requirement:
* The web interfaces shall be compatible with major web browsers, including Google Chrome, Mozilla Firefox, and Microsoft Edge.

### **3.6.1 Reliability**

**Error Handling:**

Detail the expected behavior when errors occur, including error messages displayed to users, logging of errors for analysis, and any automated recovery processes.

**Data Integrity:**

Specify measures to ensure the integrity of data stored in the system, including error-checking mechanisms, validation rules, and backup procedures to prevent and recover from data corruption.

**Data Backup and Recovery:**

Describe the frequency and procedures for data backups, as well as the mechanisms for data recovery in case of system failures or data loss.

**Performance Monitoring:**

Outline the methods for monitoring the performance of the web app, including tools, metrics, and frequency of performance assessments. This ensures that the system operates within defined performance thresholds.

**Scalability**:

The server the database is housed in can be expanded to accommodate an increased storage need as needed.

### **3.6.3 Security**

* Not needed or specified.

### 

### **3.6.4 Maintainability**

***Bug Fixes:***

Regularly identify and address any software bugs or glitches that may arise during usage.

***Performance Monitoring and Optimization:***

Monitor the performance of the web app and optimize its code, database queries, and other components to ensure efficient operation, especially as the data and user load increase.

***Data Backup and Recovery:***

Implement regular data backup procedures to prevent data loss. Ensure that a robust recovery mechanism is in place in case of any unexpected issues.

***User Feedback and Improvement:***

Gather user feedback and incorporate necessary improvements to enhance user experience and address any usability issues.

### **3.6.5 Portability**

System runs on any machine with access to the internet and a browser

| **ID** | **Characteristic** | **H/M/L** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Correctness |  |  |  |  |  | x |  |  |  |  |  |  |  |
| 2 | Efficiency |  |  |  |  |  | x |  |  |  |  |  |  |  |
| 3 | Flexibility |  |  |  |  |  | x |  |  |  |  |  |  |  |
| 4 | Integrity/Security |  |  |  |  |  | x |  |  |  |  |  |  |  |
| 5 | Interoperability |  |  |  |  |  | x |  |  |  |  |  |  |  |
| 6 | Maintainability |  |  |  |  |  | x |  |  |  |  |  |  |  |
| 7 | Portability |  |  |  |  |  | x |  |  |  |  |  |  |  |
| 8 | Reliability |  |  |  |  |  | x |  |  |  |  |  |  |  |
| 9 | Reusability |  |  |  |  |  | x |  |  |  |  |  |  |  |
| 10 | Testability |  |  |  |  |  | x |  |  |  |  |  |  |  |
| 11 | Usability |  |  |  |  |  | x |  |  |  |  |  |  |  |
| 12 | Availability |  |  |  |  |  | x |  |  |  |  |  |  |  |

## **3.7 Organizing the Specific Requirements**

* Normalized relational database
* DB diagram documentation
* Modern web UI
* Efficient API
* API endpoints and models documentation
* API and/or DB access for generating reports

### **3.7.1 System Mode**

**Initialization Mode**

In this mode, the system initializes and sets up the necessary configurations. Users with administrative privileges can define survey parameters, inspection criteria, and user roles. This mode ensures that the application is ready for conducting dam safety surveys.

**Survey Creation Mode**

Authorized users can enter this mode to create new dam safety surveys. This involves specifying details such as dam location, type, dimensions, and any other relevant information. Survey templates may be utilized to streamline the process and ensure consistency across different surveys.

### **3.7.2 User Class**

Application and Database Administration – These users will be able to modify survey content, modify domain list content, create new surveys, define evaluation periods, notify respondents a survey has been requested

Surveyors – These users will be DSMC employees which are able to add respondents to be surveyed and select the mission/program/project and tasks that are to be evaluated

Respondents – These users will be able to link to and complete a requested survey

### 

### **3.7.3 Objects**

**Dam Safety Assessment:**

Provide a platform for conducting comprehensive dam safety assessments through a user-friendly interface.

**Data Collection and Management:**

Facilitate efficient collection, storage, and management of dam-related data, including structural, geotechnical, and environmental information.

**Risk Analysis and Mitigation:**

Enable users to perform risk analysis based on collected data, identifying potential hazards and suggesting appropriate mitigation measures.

**Regulatory Compliance:**

Ensure compliance with relevant dam safety regulations, standards, and guidelines by incorporating necessary features and reporting mechanisms.

**Collaboration and Communication:**

Support collaboration among dam safety professionals by providing tools for real-time communication, data sharing, and collaborative decision-making.

**User Authentication and Authorization:**

Implement secure user authentication and authorization mechanisms to control access to sensitive dam safety data and features.

### **3.7.4 Feature**

***1. User-Friendly Interface:***

*•* Ensure a simple and intuitive interface that allows users to navigate easily and submit reviews without confusion*.*

***2. Review Submission:***

*•* Provide a clear and accessible mechanism for users to submit reviews, including text feedback and ratings.

***3. Detailed Feedback Forms:***

*•* Include customizable forms or fields that allow users to provide detailed feedback on specific aspects of the company's

***System Triggers:***

Trigger: A survey has been left unanswered for 15 days

Response: An email notification is sent to the specific user who has not completed the survey

Trigger: A survey is made

Response: An email notification is sent to all relevant users that a new survey has been made.

**Survey Data Collection**

* **Dam Information**

**Dam Identification:**

The system shall collect and store information about the dam, including name, location, type, and purpose.

**Structural Details:**

The survey app shall include fields for capturing structural details of the dam, such as construction material, height, and length.

**Reservoir Information:**

Users should be able to input data related to the dam's reservoir, including capacity, water levels, and current storage.

### **3.7.7 Functional Hierarchy**

### 

**User Interface**

**Landing Page**

* Display general information about the survey app.
* Provide access to user authentication and registration.

**User Dashboard**

* Present a personalized dashboard for each user role (dam owner, inspector, administrator).
* Include widgets summarizing dam safety data and incident reports.

**Dam Details View**

* Provide a comprehensive view of all registered dams.
* Enable users to update dam information as needed.

**Survey Module**

**Survey Creation**

* Allow administrators to create and customize dam safety surveys.
* Define survey questions and response types.

**Survey Distribution**

* Facilitate the distribution of surveys to relevant dam owners and inspectors.
* Send notifications for pending surveys and reminders.

**Survey Response Collection**

* Capture and store survey responses securely.

Implement validation checks for response accuracy and completeness.

**Incident Dashboard**

* Display a dashboard summarizing reported incidents.
* Allow filtering and sorting based on incident parameters.

**Reporting and Analysis**

**Dashboard Visualization**

* Develop visualizations on the dashboard to represent dam safety status.
* Include charts, graphs, and maps for a comprehensive overview.

**Custom Report Generator**

* Enable users to generate custom reports based on specified criteria.
* Export reports in different formats (PDF, CSV).

**Integration**

* Notification Integration
* Integrate with external notification systems.
* Send real-time alerts for critical dam safety issues.

# **Change Management Process**

If there needs to be a change, email the team leads through their Marshall University emails. They will be brought up at the next weekly team meeting to be added.

# **Document Approvals**

*Paulus Wahjudi*

*Raymond (Butch) Rogers*

**Outline for SRS Section 3**

**Organized by mode: Version 1**

3. Specific Requirements

3.1 External interface requirements

1. User interfaces
2. Hardware interfaces
3. Software interfaces
4. Communications interfaces
5. Functional requirements

3.2.1 Mode 1

3.2.1.1 Functional requirement 1.1

.....

3.2.1.*n* Functional requirement 1.*n*

1. Mode 2

.....

3.2.*m* Mode *m*

3.2.*m*.1 Functional requirement *m*.1

.....

3.2.*m.n* Functional requirement *m.n*

3.3 Performance Requirements

3.4 Design Constraints

3.5 Software system attributes

3.6 Other requirements

**Outline for SRS Section 3**

**Organized by mode: Version 2**

3. Specific Requirements

3.1 Functional Requirements

1. Mode 1

3.1.1.1 External interfaces

3.1.1.1 User interfaces

3.1.1.2 Hardware interfaces

3.1.1.3 Software interfaces

3.1.1.4 Communications interfaces

3.1.1.2 Functional Requirement

3.1.1.2.1 Functional requirement 1

.....

3.1.1.2.*n* Functional requirement *n*

3.1.1.3 Performance

3.1.2 Mode 2

.....

3.1.*m* Mode *m*

1. Design constraints
2. Software system attributes
3. Other requirements

**Outline for SRS Section 3**

**Organized by user class (i.e. different types of users ->System Adminstrators, Managers, Clerks, etc.)**

3. Specific Requirements

3.1 External interface requirements

1. User interfaces
2. Hardware interfaces
3. Software interfaces
4. Communications interfaces
5. Functional requirements

3.2.1 User class 1

3.2.1.1 Functional requirement 1.1

.....

3.2.1.*n* Functional requirement 1.*n*

1. User class 2

.....

3.2.*m* User class *m*

3.2.*m*.1 Functional requirement *m*.1

.....

3.2.*m.n* Functional requirement *m.n*

3.3 Performance Requirements

3.4 Design Constraints

3.5 Software system attributes

3.6 Other requirements

**Outline for SRS Section 3**

**Organized by object (Good if you did an object-oriented analysis as part of your requirements)**

3 Specific Requirements

3.1 External interface requirements

1. User interfaces
2. Hardware interfaces
3. Software interfaces
4. Communications interfaces
5. Classes/Objects

3.2.1 Class/Object 1

3.2.1.1 Attributes (direct or inherited)

1. Attribute 1

.....

3.2.1.1.*n* Attribute *n*

1. Functions (services, methods, direct or inherited)

3.2.1.2.1 Functional requirement 1.1

.....

3.2.1.2.*m* Functional requirement 1.*m*

3.2.1.3 Messages (communications received or sent)

3.2.2 Class/Object 2

.....

3.2.*p* Class/Object *p*

3.3 Performance Requirements

3.4 Design Constraints

3.5 Software system attributes

3.6 Other requirements

**Outline for SRS Section 3**

**Organized by feature (Good when there are clearly delimited feature sets.**

3 Specific Requirements

3.1 External interface requirements

1. User interfaces
2. Hardware interfaces
3. Software interfaces
4. Communications interfaces
5. System features

3.2.1 System Feature 1

3.2.1.1 Introduction/Purpose of feature

3.2.1.2 Stimulus/Response sequence

3.2.1.3 Associated functional requirements

3.2.1.3.1 Functional requirement 1

.....

3.2.1.3.*n* Functional requirement *n*

3.2.2 System Feature 2

.....

3.2.*m* System Feature *m*

.....

3.3 Performance Requirements

3.4 Design Constraints

3.5 Software system attributes

3.6 Other requirements

**Outline for SRS Section 3**

**Organized by stimulus (Good for event driven systems where the events form logical groupings)**

3 Specific Requirements

3.1 External interface requirements

1. User interfaces
2. Hardware interfaces
3. Software interfaces
4. Communications interfaces
5. Functional requirements

3.2.1 Stimulus 1

3.2.1.1 Functional requirement 1.1

.....

3.2.1.*n* Functional requirement 1.*n*

3.2.2 Stimulus 2

.....

3.2.*m* Stimulus *m*

3.2.*m*.1 Functional requirement *m*.1

.....

3.2.*m.n* Functional requirement *m.n*

3.3 Performance Requirements

3.4 Design Constraints

3.5 Software system attributes

3.6 Other requirements

**Outline for SRS Section 3**

**Organized by response (Good for event driven systems where the responses form logical groupings)**

3 Specific Requirements

3.1 External interface requirements

1. User interfaces
2. Hardware interfaces
3. Software interfaces
4. Communications interfaces
5. Functional requirements

3.2.1 Response 1

3.2.1.1 Functional requirement 1.1

.....

3.2.1.*n* Functional requirement 1.*n*

3.2.2 Response 2

.....

3.2.*m* Response *m*

3.2.*m*.1 Functional requirement *m*.1

.....

3.2.*m.n* Functional requirement *m.n*

3.3 Performance Requirements

3.4 Design Constraints

3.5 Software system attributes

3.6 Other requirements

**Outline for SRS Section 3**

**Organized by functional hierarchy (Good if you have done structured analysis as part of your design.)**

3 Specific Requirements

3.1 External interface requirements

1. User interfaces
2. Hardware interfaces
3. Software interfaces
4. Communications interfaces
5. Functional requirements

3.2.1 Information flows

3.2.1.1 Data flow diagram 1

1. Data entities
2. Pertinent processes
3. Topology

3.2.1.2 Data flow diagram 2

1. Data entities
2. Pertinent processes
3. Topology

.....

3.2.1.*n* Data flow diagram *n*

3.2.1.*n*.1 Data entities

3.2.1.*n*.2 Pertinent processes

3.2.1.*n*.3 Topology

3.2.2 Process descriptions

1. Process 1
2. Input data entities
3. Algorithm or formula of process
4. Affected data entities

3.2.2.2 Process 2

3.2.2.2.1 Input data entities

3.2.2.2.2 Algorithm or formula of process

3.2.2.2.3 Affected data entities

.….

3.2.2.*m* Process *m*

3.2.2.*m*.1 Input data entities

3.2.2.*m*.2 Algorithm or formula of process

3.2.2.*m*.3 Affected data entities

3.2.3 Data construct specifications

3.2.3.1 Construct 1

3.2.3.1.1 Record type

3.2.3.1.2 Constituent fields

3.2.3.2 Construct 2

3.2.3.2.1 Record type

3.2.3.2.2 Constituent fields

…..

3.2.3.*p* Construct *p*

3.2.3.*p*.1 Record type

3.2.3.*p*.2 Constituent fields

3.2.4 Data dictionary

3.2.4.1 Data element 1

3.2.4.1.1 Name

3.2.4.1.2 Representation

3.2.4.1.3 Units/Format

3.2.4.1.4 Precision/Accuracy

3.2.4.1.5 Range

3.2.4.2 Data element 2

3.2.4.2.1 Name

3.2.4.2.2 Representation

3.2.4.2.3 Units/Format

3.2.4.2.4 Precision/Accuracy

3.2.4.2.5 Range

…..

3.2.4.*q* Data element *q*

3.2.4.*q*.1 Name

3.2.4.*q*.2 Representation

3.2.4.*q*.3 Units/Format

3.2.4.*q*.4 Precision/Accuracy

3.2.4.*q*.5 Range

3.3 Performance Requirements

3.4 Design Constraints

3.5 Software system attributes

3.6 Other requirements

**Outline for SRS Section 3**

**Showing multiple organizations (Can’t decide? Then glob it all together)**

3 Specific Requirements

3.1 External interface requirements

1. User interfaces
2. Hardware interfaces
3. Software interfaces
4. Communications interfaces
5. Functional requirements

3.2.1 User class 1

3.2.1.1 Feature 1.1

3.2.1.1.1 Introduction/Purpose of feature

3.2.1.1.2 Stimulus/Response sequence

3.2.1.1.3 Associated functional requirements

3.2.1.2 Feature 1.2

3.2.1.2.1 Introduction/Purpose of feature

3.2.1.2.2 Stimulus/Response sequence

3.2.1.2.3 Associated functional requirements

…..

3.2.1.*m* Feature 1.*m*

3.2.1.*m*.1 Introduction/Purpose of feature

3.2.1.*m*.2 Stimulus/Response sequence

3.2.1.*m*.3 Associated functional requirements

3.2.2 User class 2

.....

3.2.*n* User class *n*

.....

3.3 Performance Requirements

3.4 Design Constraints

3.5 Software system attributes

3.6 Other requirements

**Outline for SRS Section 3**

**Organized by Use Case (Good when following UML development)**

3. Specific Requirements

3.1 External Actor Descriptions

3.1.1 Human Actors

3.1.2 Hardware Actors

3.1.3 Software System Actors

3.2 Use Case Descriptions

3.2.1 Use Case 1

3.2.2 Use Case 2

3.2.n Use Case n

3.3 Performance Requirements

3.4 Design Constraints

3.5 Software system attributes

3.6 Other requirements