
SOFTWARE REQUIREMENTS SPECIFICATION

for

Software Risk Tracking

Release 0.2

Prepared by Software Risk Tracking Team

Contents

List of Figures	4
List of Tables	5
1 INTRODUCTION	7
1.1 Purpose	7
1.2 Document Conventions	7
1.3 Acronyms	8
1.4 Project Scope and Product Features	8
2 OVERALL DESCRIPTION	9
2.1 Product Perspective	9
2.2 Product Features	9
2.2.1 Admin users - add/edit/delete users	9
2.2.2 Admins - create and maintain risk ranks	10
2.2.3 Admins - create and maintain status types	10
2.2.4 Admins - define ID format	10
2.2.5 All users - display/print risk reports	10
2.2.6 All users - display/print risk matrix	10
2.2.7 Admins - purge risks	10
2.3 User Classes and Characteristics	10
2.4 Operating Environment	15
2.5 Design and Implementation Constraints	15
3 NONFUNCTIONAL REQUIREMENTS	16
3.1 Performance Requirements	16
3.2 Safety Requirements	16
3.3 Security Requirements	16

3.4	Software Quality Attributes	17
4	STANDARDS AND REFERENCES	18
4.1	Applicable Standards	18
4.2	References	18
4.2.1	Links	18

List of Figures

2.1	Admin Use Case	12
2.2	Track Use Case	13
2.3	Audit Use Case	13
2.4	Manage Users Activity	14

List of Tables

1.1 Acronym List 8

Revision History

Date	Description	Revised by
2/21/24	Baseline	Software Risk Tracking Team
3/2/24	Version 0.2	Software Risk Tracking Team

1 INTRODUCTION

1.1 Purpose

The purpose of this document is to build a software risk-tracking system that will identify risks and help the business streamline its Software. It will keep track of the impact of risk and its status. Once completed it will help the team analyze the risks according to their ranks and solve accordingly.

1.2 Document Conventions

This SRS document conforms to the IEEE Standards Style Manual, ensuring consistency and clarity in the communication of requirements. The term "shall" is used to indicate mandatory requirements which must be strictly followed. The term "should" suggests a course of action among several possibilities, recommended for its suitability, though not exclusively mandated. "May" indicates a permissible course of action within the requirements' limits. The phrases "must" and "will" are deprecated for describing mandatory requirements in this document. For data entry fields, "text" or "printable text" refers to any printable alphanumeric character available on the US standard keyboard, excluding tabs, carriage returns, new lines, and other non-printable or control characters. When "numeric" is indicated for a data entry field, only digits are permitted, and characters such as dashes, commas, or periods are excluded unless specifically allowed.

1.3 Acronyms

CS	Computer Science
BSD	Berkeley Software Distribution
OS	Operating System
AMD	Advanced Micro Devices
STL	Standard Template Library
SQL	Structured Query Language
GB	Giga Byte
RAM	Random Access Memory
RMS	Risk Management System
ERP	Enterprise Resource Planning
CRM	Customer Relationship Management
REST	Representational State Transfer
API	Application Programming Interface
IEEE	the Institute of Electrical and Electronics Engineers

Table 1.1: Acronym List

1.4 Project Scope and Product Features

The purpose of the risk tracking system is to help maintain a tracked list of risks to aid in the management of said risks.

The application allows admin users to perform the following tasks: add/edit/delete users, assign a single or multiple roles to users, create and maintain risk ranks, create and maintain status types, define ID format, and purge risks.

The application allows all users to perform the following tasks: display/print risk reports and display/print risk matrices.

2 OVERALL DESCRIPTION

2.1 Product Perspective

The Risk Management System (RMS) is designed as a web-based application integral to an organization's IT ecosystem, aiming to streamline risk management processes for small to medium-sized businesses. Hosted on the UNA CS Server, it operates within a secure and stable OpenBSD environment, optimized for 64-bit Intel/AMD architectures.

Integration and Dependencies

This product relies on a MySQL database for data storage, utilizing C++ with STL and Boost Libraries for backend development to ensure robust performance and scalability, and the Wt ("Witty") library for frontend.

Operating Environment and Scalability

Refer to section 2.4

2.2 Product Features

2.2.1 Admin users - add/edit/delete users

Users of the application shall be assigned one or more of the following roles: Admin (allowed access to all privileges), Audit (allowed access to view only), and Track (allowed access to add, edit, and delete risks). An Admin user has the authority to change any user's assigned roles.

2.2.2 Admins - create and maintain risk ranks

Risks shall be ranked by their likelihood and impact which are both independent variables. Each risk shall include the following information: ID, short description, long description, likelihood rank, impact rank, owner, status, notes, open date, and close date.

2.2.3 Admins - create and maintain status types

Admins may create status types for risks based on their team's needs.

2.2.4 Admins - define ID format

Admins may specify how risk IDs are formatted in the system.

2.2.5 All users - display/print risk reports

Risks shall be displayed by their ranks, date, and owner. When a risk report is created, all fields are shown.

2.2.6 All users - display/print risk matrix

Risk matrices shall be displayed by their Risk ID. On the login screen, a link is included to a page for "forgot password".

2.2.7 Admins - purge risks

Admin users may purge risks by date or by ID.

2.3 User Classes and Characteristics

This section outlines stepped use cases and scenarios and any exceptions to those actions an actor may experience using this software.

Use cases for the system describe the steps that a user could take for viewing, adding, editing, or deleting risks. Users can be either an admin, audit, or track. Each role has differing functionality and permissions. Administrators have all privileges. Auditors have view only privileges. Trackers have privileges to add, edit, or delete risks.

The admin role shall have all privileges to every action associated with the risk tracking software.

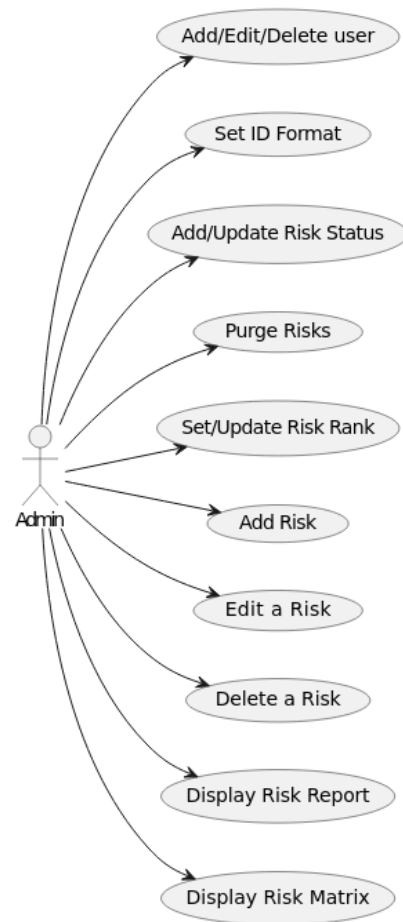


Figure 2.1: Admin Use Case

A subset of activities of the admin role are managing users, as outlined below.

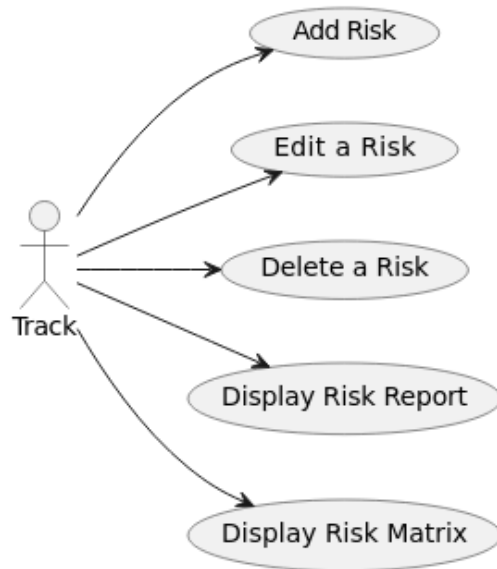


Figure 2.2: Track Use Case

The track role shall have limited privileges to make necessary risk management actions, such as add risks, print risks, etc.

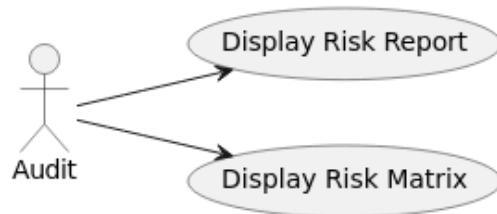


Figure 2.3: Audit Use Case

The audit role shall have the most limited privileges for the purpose of only being allowed to view risks in different manners.

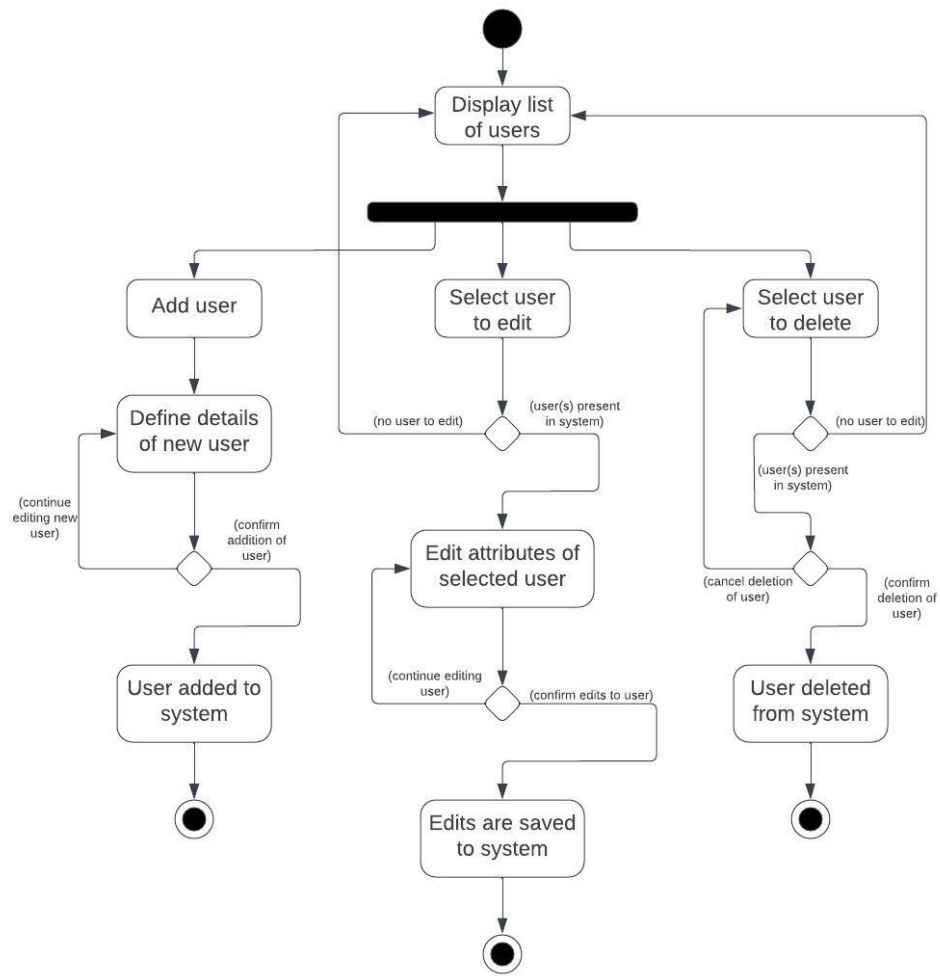


Figure 2.4: Manage Users Activity

2.4 Operating Environment

The operating environment will be on the UNA CS Server. The CS server runs an OpenBSD OS, 64-bit, Intel/AMD compatible architecture.

This software will be a self-contained C++ application that is accessed via http(s) in a browser (client). This will be done using the C++ STL and the Wt library, which will handle both frontend development and the server/client communication

The software shall utilize a MySQL database for storing user and risk information.

The system will involve many users logging in to the system to access/-modify risk information based on their privilege. The software is designed for a small business that includes less than 500 people.

The amount of disk space will depend on the amount of risks, a minimum amount would be 4GB, but may need to be scaled up. A more comfortable amount would be 16GB⁺. The disk space includes OpenBSD, MySQL, C++ STL, and Boost, along with extra space to be used to store risks.

The RAM required is 4GB at a minimum for MySQL and the running processes, but a more comfortable amount is 8GB⁺.

2.5 Design and Implementation Constraints

Refer to Section 2.4

3 NONFUNCTIONAL REQUIREMENTS

3.1 Performance Requirements

1. This software shall respond to all user-initial actions(submissions, button clicks, or others interactions) within 1 second. Achieving this response time ensures a smooth and responsive user experience.
2. Optimize to handle multiple users at once this means that the system has to have the capacity to handle requests from 16 different users at once without experiencing any lag or unresponsiveness.
3. Database Query and Stored Procedure Response Times shall be 10 milliseconds and no single query or stored procedure shall exceed 250 milliseconds in response time under typical load conditions.

3.2 Safety Requirements

This software will ensure user safety by avoiding flashing lights that could cause injury.

3.3 Security Requirements

The software will store user credentials using a MySQL database on the CS server. User credentials will be hashed when comparing the credentials the user submits.

3.4 Software Quality Attributes

The product shall make use of secure coding strategies and UI standards. It shall follow the requirements set forth by stakeholders. It shall be optimized for performance where possible to reduce wait times and failures. See the SQA Plan for more details on SQA.

4 STANDARDS AND REFERENCES

4.1 Applicable Standards

This SRS follows applicable industry and international standards to ensure the software development process's quality and interoperability. These include:

1. IEEE 830-1998: This standard provides guidelines for SRS content and format, ensuring comprehensive documentation of software requirements.
2. ISO/IEC 12207: This international standard specifies requirements for software lifecycle processes, guiding the planning, development, testing, and maintenance of software.
3. ISO/IEC 27001: It outlines best practices for an information security management system (ISMS), ensuring the confidentiality, integrity, and availability of data.
4. ANSI/NIST-ITL 1-2011: This standard specifies data formats for the interchange of information between criminal justice organizations.

4.2 References

4.2.1 Links

- <https://advocacy.sba.gov/wp-content/uploads/2023/03/Frequently-Asked-Questions-About-Small-Business-March-2023-508c.pdf>
- <https://www.boost.org/users/history/>

- <https://dev.mysql.com/doc/mysql-monitor/8.0/en/system-pre-reqs-reference.html>
- <https://www.openbsd.org/faq/faq4.html#:~:text=OpenBSD%20can%20be%20installed%20in,more%20disk%20space%20is%20recommended.>