



SOFTWARE DESIGN DOCUMENT

for

Software Bill of Materials Tracker

Release 1.0

Version 1.0 approved

Prepared by SBOM Team



Contents

List of Figures	4
List of Tables	5
1 INTRODUCTION	7
1.1 Purpose	7
1.2 Scope	7
1.3 Overview	7
1.4 References	8
1.5 Definitions and Acronyms	8
2 SYSTEM OVERVIEW	9
3 SYSTEM ARCHITECTURE	10
3.1 Architectural Design	10
3.2 Decomposition Description	11
3.3 Design Rationale	13
4 DATA DESIGN	14
4.1 Data Description	14
4.2 Data Dictionary	14
5 COMPONENT DESIGN	15
5.1 User Interface (UI) Component	15
5.1.1 Description	15
5.1.2 PDL/Pseudocode	15
5.1.3 UML Diagram Design	16
5.2 SBOM Management Component	16
5.2.1 Description	16

5.2.2	PDL/Pseudocode	16
5.2.3	UML Diagram Design	17
5.3	Database Component	18
5.3.1	Description	18
5.3.2	PDL/Pseudocode	18
5.3.3	UML Diagram Design	18
5.4	Authentication and User Management Component	19
5.4.1	Description	19
5.4.2	PDL/Pseudocode	19
5.4.3	UML Diagram Design	19
5.5	Export Component	20
5.5.1	Description	20
5.5.2	PDL/Pseudocode	20
5.5.3	UML Diagram Design	20
5.6	Real-Time Collaboration Component	21
5.6.1	Description	21
5.6.2	PDL/Pseudocode	21
6	HUMAN INTERFACE DESIGN	22
6.1	Overview of User Interface	22
6.2	Screen Designs	24
7	REQUIREMENTS MATRIX	31

List of Figures

6.1	SBOM Manager, Screen Design for Login	25
6.2	SBOM Manager, Screen Design for Registration Form	26
6.3	SBOM Manager, Screen Design for SBOM Library	27
6.4	SBOM Manager, Screen Design for SBOM Details	28
6.5	SBOM Manager, Screen Design for Change-Log	29
6.6	SBOM Manager, Screen Design for Create, Edit, and Delete Forms	30

List of Tables

4.1	Data dictionary Table	14
7.1	Requirements Matrix	31

Revision History

Date	Description	Revised by
03/11/22	Initial draft	Team name

1 INTRODUCTION

1.1 Purpose

This software design document (SDD) describes the architecture and system design of a Software Bill of Materials tracker. Starting at a general overview and decomposing down to the component and data levels.

1.2 Scope

A multi-user software that allows a user to create, edit, or remove software bills of materials for projects they are apart of. SBOMs must be exportable as a comma separated values (csv) file. The software must have features that support a user login system; a system to store different projects a user is apart of; features that support the creation, editing, and deletion of SBOMs; and be able to export SBOMs in a csv file format.

1.3 Overview

This software design document (SDD) contains the information necessary to understand the basis of the Software Bill of Materials (SBOM) tracker and its components. The design information provided is broken up into overarching sections. These sections start at higher overview/system architecture levels of abstraction and then work down to the lower component/user interface levels of abstraction. This progress is to be used by readers to get a general grasp of the SBOM tracker before learning about its internal components, data, and data structures. Also, the requirements matrix is to be

used to understand the ties between components/data and their underlying functional and/or nonfunctional requirements.

1.4 References

1.5 Definitions and Acronyms

Acronym	Description
SBOM	Software Bill of Materials
CSV	Comma Separated Values
UML	Unified Modeling Language

2 SYSTEM OVERVIEW

The Software Bill of Materials (SBOM) Tracking System is designed to provide transparency and accountability within software supply chains. SBOM will help organizations to understand, manage and secure their applications by allowing a detailed inventory, allowing different teams within an organization to collaborate effectively by providing them with a unified view of all components associated with any given project. The system will allow the users to create multiple records as well as edit, and delete them simultaneously or individually. User will also be able to edit the same SBOM records together. System will let users export SBOM data into CSV format for the ease of accessing, printing and sending SBOM records. Users will also be able to view a log of changes made to the records.

3 SYSTEM ARCHITECTURE

3.1 Architectural Design

The SBOM tracker system can be divided into five high-level subsystems. These subsystems are:

1. User Management Subsystem
2. SBOM Management Subsystem
3. Export Subsystem
4. History Log Subsystem
5. Database Subsystem

User Management Subsystem

The user management subsystem handles secure access for the SBOM system by ensuring that only authorized users can access or modify SBOMs in the system.

SBOM Management Subsystem

The SBOM management subsystem allows the authorized users to create, edit, and delete SBOMs.

Export Subsystem

The export subsystem allows for the exportation of SBOMs in the CSV(Comma-Separated Values) format.

History Log Subsystem

The history log subsystem maintains a log which contains data about modifications to the SBOMs.

Database Subsystem

The database subsystem stores the data for the SBOMs, user information, and modification logs.

Interactions between different subsystems

1. The user management subsystem authenticates the user and forwards them to the SBOM management subsystem, allowing authenticated users to create, delete, or edit SBOMs
2. When an authenticated user modifies anything on the SBOM system, this change will be reflected on the modification log maintained by the history log subsystem. This change will also be updated to the existing data by the database subsystem.
3. Finally, when an authorized user wants to export a SBOM from the system, the database subsystem sends the data to the export subsystem for conversion to a CSV(Comma-Separated Values) format.

3.2 Decomposition Description

The subsystems of the SBOM tracker system can be further decomposed into modules.

User Management Subsystem

The user management subsystem consists of two modules:

1. Verification Module: This module verifies the login and password of the user.

2. Authorization Module: This module is responsible for controlling access levels and permissions required to modify SBOMs on the system.

SBOM Management Subsystem

The SBOM management subsystem consists of three modules:

1. SBOM Creation Module: This module allows for the creation of new SBOMs within the system.
2. SBOM Deletion Module: This module allows deletion of existing SBOMs in the system.
3. SBOM Modification Module: This module allows users to edit the existing SBOMs in the system.

Export Subsystem

The export subsystem consists of two modules:

1. CSV conversion Module: This module converts the SBOM data into a CSV format.
2. Export Module: This module exports the CSV format of SBOM data.

History Log Subsystem

The history log subsystem consists of two modules:

1. Logging Module: This module tracks the changes made to the SBOMs.
2. Display Module: This module displays the changes made to the user.

Database Subsystem

The database subsystem consists of one module:

1. Storage Module: This module saves and retrieves data of SBOMs.

3.3 Design Rationale

This SBOM tracker system was structured in a modular approach to distinguish clear responsibilities of each subsystems, and the nature of their interactions with each other. In addition, this modularization allows for better scalability, flexibility, and ease of maintenance.

An alternative approach was to design the system as a monolithic block. However, this design would be more difficult for scalability, as it is more difficult to modify individual functionalities without causing unintended changes to the whole system.

4 DATA DESIGN

4.1 Data Description

Data from the users, products, and orders will all be passed through encrypted channels to secure the integrity and security of the users, which will then be stored in a database. This database cannot be accessed unless through hashed passwords to ensure no false users can access or alter the data

4.2 Data Dictionary

Data will be listed in the system by entity name, the type of data, and a description of what that data represents. an example of this is in the following table below.

Entity name	Type	Description
User	Object	Represents a user of the system
Name	String	The name of the user
Email	String	The email address of the user
Order	Object	Represents an order placed by a user
OrderID	Integer	Unique identifier for the order
OrderDate	String	The Date the order was placed

Table 4.1: Data dictionary Table

5 COMPONENT DESIGN

The SBOM Tracker system consists of several key components, each responsible for specific functionality. This chapter provides a detailed description of each component's role, its functional behavior, and how the components interact within the system.

5.1 User Interface (UI) Component

5.1.1 Description

The User Interface (UI) Component allows users to interact with the system through a web-based interface built using Webtoolkit "Witty". It facilitates user interaction for creating, editing, deleting SBOMs, and exporting them in CSV format.

5.1.2 PDL/Pseudocode

Function displayHomePage():

- Render SBOM list

- Provide options: Create, Edit, Delete, Export SBOM

Function createSBOM():

- Get user input for SBOM details

- Send request to SBOM Management Component to create a new SBOM

Function editSBOM(s bomID):

- Get SBOM details from SBOM Management Component

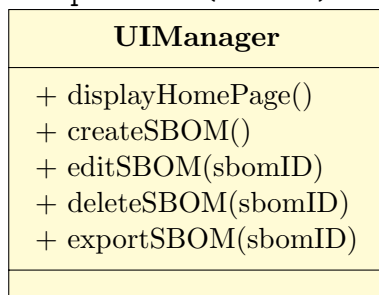
- Update the fields and send update request

```
Function deleteSBOM(s bomID):
    Confirm deletion with the user
    Send delete request to SBOM Management Component
```

```
Function exportSBOM(s bomID):
    Request SBOM data from SBOM Management Component
    Convert data to CSV format
    Provide CSV file for download
```

5.1.3 UML Diagram Design

In the UML class diagram below, the `UIManager` class includes methods like `displayHomePage()`, `createSBOM()`, `editSBOM(s bomID)`, `deleteSBOM(s bomID)`, and `exportSBOM(s bomID)`.



5.2 SBOM Management Component

5.2.1 Description

The SBOM Management Component handles the creation, updating, deletion, and retrieval of SBOM records. It interacts with the database to ensure the proper handling of SBOM data and maintains consistency.

5.2.2 PDL/Pseudocode

```
Function createSBOM(data):
```



```
Validate SBOM data
Insert SBOM data into the database
Return success/failure
```

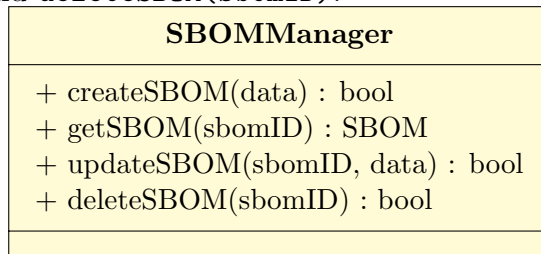
```
Function getSbom(sbomID):
    Query database for SBOM with sbomID
    Return SBOM data
```

```
Function updateSBOM(sbomID, data):
    Validate new SBOM data
    Update SBOM record in the database
    Return success/failure
```

```
Function deleteSBOM(sbomID):
    Validate sbomID
    Delete SBOM record from the database
    Return success/failure
```

5.2.3 UML Diagram Design

In the UML class diagram below, the `SBOMManager` class includes methods like `createSBOM(data)`, `getSBOM(sbomID)`, `updateSBOM(sbomID, data)`, and `deleteSBOM(sbomID)`.



5.3 Database Component

5.3.1 Description

The Database Component handles the storage of SBOMs and user data. It provides methods for creating, retrieving, updating, and deleting records from the database.

5.3.2 PDL/Pseudocode

```
Function insertRecord(table, data):  
    Generate SQL query for inserting data into 'table'  
    Execute query  
    Return success/failure  
  
Function retrieveRecord(table, criteria):  
    Generate SQL SELECT query with 'criteria'  
    Execute query and return data  
  
Function updateRecord(table, data, criteria):  
    Generate SQL UPDATE query with 'data' and 'criteria'  
    Execute query and return success/failure  
  
Function deleteRecord(table, criteria):  
    Generate SQL DELETE query with 'criteria'  
    Execute query and return success/failure
```

5.3.3 UML Diagram Design

In the UML class diagram below, the Database class includes methods for database operations such as `insertRecord()`, `retrieveRecord()`, `updateRecord()`, and `deleteRecord()`.

Database
+ insertRecord(table, data) : bool + retrieveRecord(table, criteria) : Record + updateRecord(table, data, criteria) : bool + deleteRecord(table, criteria) : bool

5.4 Authentication and User Management Component

5.4.1 Description

The Authentication and User Management Component authenticates users and manages their access rights, ensuring secure operations.

5.4.2 PDL/Pseudocode

Function authenticateUser(username, password):

 Hash the password

 Check database for username and hashed password

 Return authentication success/failure

Function checkUserPermissions(userID, action):

 Query database for user role and permissions

 Return true/false for permission to perform action

5.4.3 UML Diagram Design

In the UML class diagram below, the AuthManager class includes methods like authenticateUser() and checkUserPermissions().

AuthManager
+ authenticateUser(username, password) : bool + checkUserPermissions(userID, action) : bool

5.5 Export Component

5.5.1 Description

The Export Component handles exporting SBOM data to CSV format for external use.

5.5.2 PDL/Pseudocode

```
Function exportSBOMToCSV(s bomID):
    Get SBOM data using SBOM Management Component
    Format data into CSV structure
    Create a CSV file
    Provide the file for download
```

5.5.3 UML Diagram Design

In the UML class diagram below, the **ExportManager** class includes the method `exportSBOMToCSV(s bomID)`.

ExportManager
+ exportSBOMToCSV(s bomID) : CSVFile

5.6 Real-Time Collaboration Component

5.6.1 Description

This component allows multiple users to collaborate on SBOMs in real-time, synchronizing changes across sessions.

5.6.2 PDL/Pseudocode

```
Function synchronizeChanges(sbomID, userAction):  
    Notify all active users working on the same SBOM  
    Apply changes to the shared SBOM data  
    Reflect updates in the UI of all users
```

6 HUMAN INTERFACE DESIGN

6.1 Overview of User Interface

The SBOM Tracking System provides users with a way to track and manage the components, dependencies, and metadata associated with a given software application. The provided user interfaces help users maintain visibility into software supply chains by facilitating the creation and maintenance of SBOMs for various projects with the following interfaces:

1. **Login:** The login page serves as a gateway for users to access the system.
 - **Email/Username Field:** A text field where users input a registered email.
 - **Password Field:** A text field where users input a registered password. The contents of this field are masked with asterisks for an added layer of security.
 - **Login Button:** A button that submits the login form and authenticates the user.
 - **Sign-up Option:** A link that opens a new window for new users to register a new email and password.
2. **Registration Form:** The registration form allows users to create a new account for the SBOM Tracking System.
 - **Email Address Field:** A required text field for users to input a valid email address.

- **Password Creation:** Two required text fields. The first field is for inputting a password while the second field confirms it.
 - **Sign-Up Button:** Button to submit the registration form and create user account.
3. **SBOM Library:** A repository for all SBOMs under a given project.
- **Table View:** List of all SBOMs with columns that include SBOM name and date of last edit. A link will be embedded in each table entry to open the details page for that particular SBOM. Format and layout is consistent with the table view from the Project Library.
 - **Actions Menu:** Quick access for managing SBOMs: provides options to create, rename, delete, and export SBOMs to a CSV file. Menu layout and format should be consistent with the actions menu from the Project Library.
 - **Create SBOM Form:** Text field for the name of the SBOM. Input for this field is required. Users will submit the form using a "Create SBOM" button located at the bottom of the form, as well as a "Cancel" button that exits the form without applying changes.
 - **Rename SBOM Form:** Text field for the name of the SBOM. When the form opens, the field will have the current name inputted as a placeholder. Users will submit the form using a "Confirm" button, as well as a "Cancel" button that exits the form without applying changes.
 - **Delete SBOM Form:** A checkbox that prompts the user to confirm the deletion of the selected SBOM(s). Users will submit the form using a "Delete SBOM" button located at the bottom of the form, as well as a "Cancel" button that exists the form without applying changes.
 - **Export SBOM Button:** A button that will download the selected SBOM in CSV format under the same name.

4. **SBOM Details:** The itemized list of software dependencies and components for the SBOM selected from the SBOM Library interface.
- **Table View:** List of all software dependencies and their relevant information (type, name, vendor, license, version, etc) presented in table format
 - **Actions Menu:** Quick access for managing SBOM dependencies: provides options to add, edit, and delete entries. Format and layout is consistent with the actions menu from the Project and SBOM Library interfaces.
 - **Add Entry Button:** Text fields for the details of the component/dependency being added. Users may either submit the form using a "Confirm" button or exit the form without applying changes using a "Cancel" button.
 - **Edit Entry Form:** Text fields for the details of the component/dependency being changed. When the form opens, the fields will have the current information inputted as a placeholder. Users may either submit the form using a "confirm" button or exit the form without applying changes using a "Cancel" button.
 - **Delete Entry Form:** A checkbox that prompts the user to confirm the deletion of the selected entries. Users may either submit the form using a "Delete" button or exit the form without applying changes using a "Cancel" button.
 - **Change Log:** A history of the selected SBOM, including a list of changes made by collaborators to SBOM records, with the most recent history being displayed at the top of the interface. Change Log records will detail which entries were changed, when those changes were made, who made those changes, as well as a general description of what changes occurred.

6.2 Screen Designs

Sign In

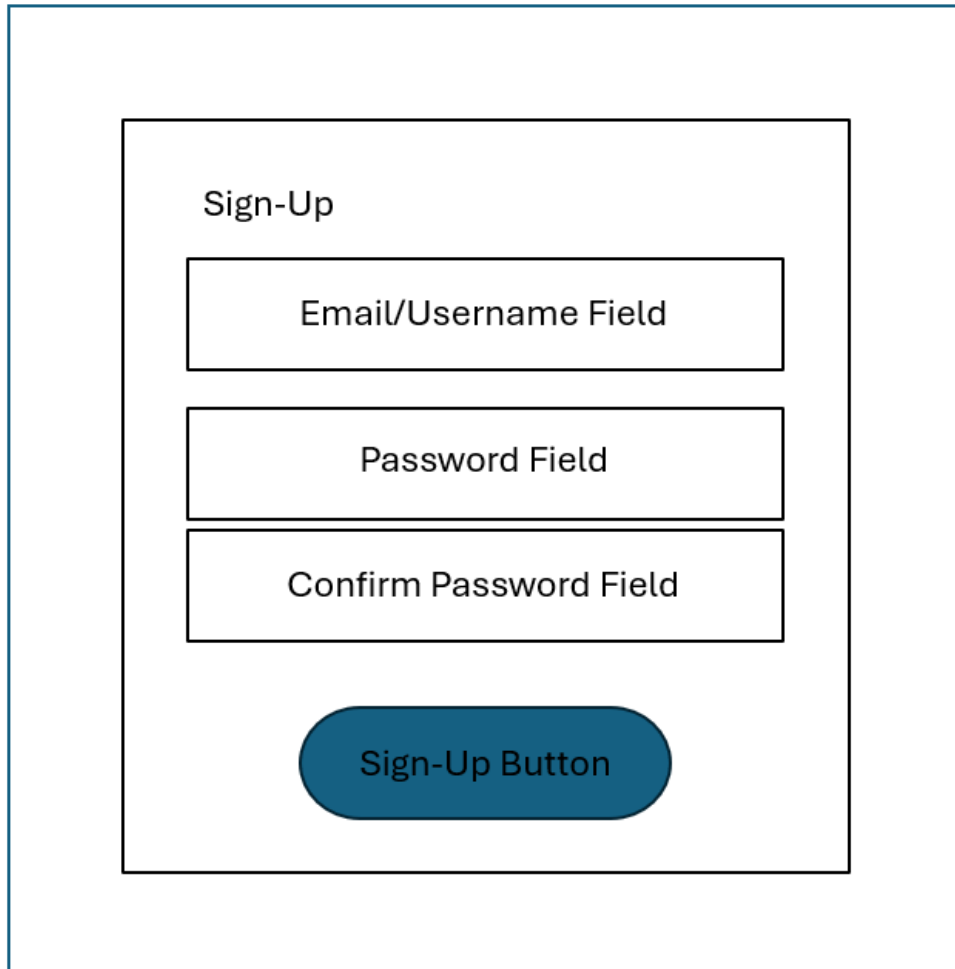
Email/Username Field

Password Field

Login Button

Sign-Up Option

Figure 6.1: SBOM Manager, Screen Design for Login



The diagram illustrates a registration form layout. It is contained within a light blue rectangular frame. Inside this frame is a white rectangular box with a thin black border. At the top left of this white box is the text "Sign-Up". Below this text are three stacked rectangular input fields, each with a thin black border. The first field is labeled "Email/Username Field", the second is labeled "Password Field", and the third is labeled "Confirm Password Field". Below these three fields is a dark blue, rounded rectangular button with the text "Sign-Up Button" in white.

Figure 6.2: SBOM Manager, Screen Design for Registration Form

[illegible]

Figure 6.4: SBOM Manager, Screen Design for SBOM Details

SBOM CHANGE-LOG

SBOM DETAILS
TAB

CHANGE LOG AND SBOM HISTORY

Collaborator	*Component and Description*	MM-DD-YYYY

TABLE VIEW

Figure 6.5: SBOM Manager, Screen Design for Change-Log

CREATE/EDIT FORMS

INFORMATION FIELDS

CANCEL BUTTON

Create/Edit Button

DELETE FORM

Obligatory Warning

Checkbox

I understand

Delete Button

CANCEL BUTTON

Figure 6.6: SBOM Manager, Screen Design for Create, Edit, and Delete Forms

7 REQUIREMENTS MATRIX

Table 7.1: Requirements Matrix

Requirement	Description	Design Component
Product Feature-1	SBOM Management: Create, edit, delete SBOM records	UIManager, SBOMManager
Product Feature-2	Multi-user support for simultaneous SBOM management	AuthManager
Product Feature-3	CSV Export: Export SBOMs in CSV format	ExportManager
Product Feature-4	Changelog Support: View log of changes	Database
Performance	Handle up to 50 users, 2s operation time	AuthManager
Security	Secure authentication and encryption	AuthManager
Usability	98% user satisfaction score in user surveys	UIManager
Scalability	Support 200 users/SBOMs	Database, AuthManager
<i>Continued on next page</i>		

Requirement	Description	Design Component
Maintainability	Modular-design, well-documented code	Code Documentation
Reliability	99.9% uptime, failure logging	Database
Compatibility	Compatibility: Chrome, Firefox, OpenBSD	UIManager