CMP307 – SOFTWARE ENGINEERING PRACTISE

REPORT

EWAN STEWART

1900598

Contents

[Design 4](#_Toc89709024)

[Requirements 4](#_Toc89709025)

[Functional 4](#_Toc89709026)

[Non-Functional 4](#_Toc89709027)

[UML 4](#_Toc89709028)

[Use Case Diagrams 4](#_Toc89709029)

[Class Diagrams 4](#_Toc89709030)

[Sequence Diagrams 4](#_Toc89709031)

[Additional Diagrams 4](#_Toc89709032)

[Implementation 5](#_Toc89709033)

[Development Methodology 5](#_Toc89709034)

[Implementation Patterns 5](#_Toc89709035)

[Testing 6](#_Toc89709036)

[Unit Testing 6](#_Toc89709037)

[Hardware & Software Asset Display 6](#_Toc89709038)

[Hardware Asset Create / Update / Delete 6](#_Toc89709039)

[Software Asset Create / Update / Delete 6](#_Toc89709040)

[Vulnerability API Call 6](#_Toc89709041)

[Database Backup 6](#_Toc89709042)

[Usability Testing 6](#_Toc89709043)

[Tester 1 6](#_Toc89709044)

[Tester 2 6](#_Toc89709045)

[Tester 3 6](#_Toc89709046)

[Reliability & Security 7](#_Toc89709047)

[Fault Avoidance 7](#_Toc89709048)

[Fault Tolerance 7](#_Toc89709049)

[Security Measures 7](#_Toc89709050)

[Account Credentials 7](#_Toc89709051)

[Full Access 7](#_Toc89709052)

[Report Access 7](#_Toc89709053)

[Software Guide 8](#_Toc89709054)

[Login 8](#_Toc89709055)

[Control Panel 8](#_Toc89709056)

[Hardware 8](#_Toc89709057)

[Create Hardware Asset 8](#_Toc89709058)

[Update Hardware Asset 8](#_Toc89709059)

[Delete Hardware Asset 8](#_Toc89709060)

[Software 8](#_Toc89709061)

[Create Software Asset 8](#_Toc89709062)

[Update Software Asset 8](#_Toc89709063)

[Delete Software Asset 8](#_Toc89709064)

[Display Assets 8](#_Toc89709065)

[Assets 8](#_Toc89709066)

[Vulnerability API Call 8](#_Toc89709067)

[Backup Database 8](#_Toc89709068)

[Finding Backup 8](#_Toc89709069)

[Logout 8](#_Toc89709070)

[Evaluation 9](#_Toc89709071)

# Design

## Requirements

### Functional

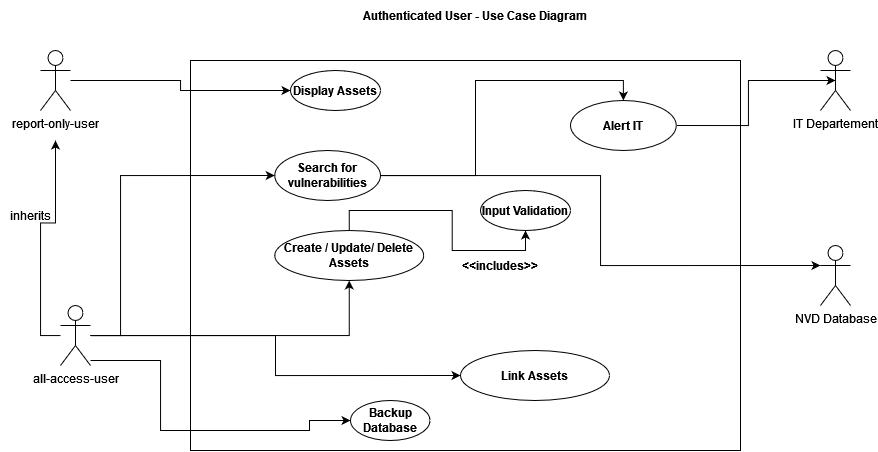
1. The system must allow the user to create / update / delete both a software and hardware asset.
2. The system must allow the user to link a hardware asset to a software asset.
3. The system must allow the asset data to be filtered by any field.
4. The system must prevent invalid data entry at the point of entry.
5. The system must have a clear instruction available for how to correctly enter data.
6. The system must run on Windows 10.
7. The system must authenticate users before they allowed access to the system.
8. The system must allow functionality for a report only account where this user can only access the asset listing feature.
9. The system must be able to provide local backups for the asset data.
10. The system must allow vulnerability reporting on the assets.

### Non-Functional

1. The system must make data entry for the user easy.
2. The system must be reliable and secure.
3. The system must make forwarding vulnerability data to the IT department easy.

## UML

### Use Case Diagram



### Use Case

Search for vulnerabilities

Main Success Scenario:

1. The user is authenticated by the system.
2. The user selects the search for vulnerabilities button
3. The user is presented with a list of all known vulnerabilities related to the data stored in the database.

Extensions:

3a. There is no vulnerabilities found.

1. The user is prompted that there is no known vulnerabilities found.

Display Assets

Main Success Scenario:

1. The user is authenticated by the system.
2. The user selects the display assets button.
3. The user is presented with tables of software and hardware assets.

Extensions:

3a. There is no data in the software and hardware asset tables.

1. The user is prompted that there is no data stored on software and hardware assets.

3b. There is data in only one of the asset tables. **(e.g., there is data in the hardware table but not software)**

1. The user is shown only the table with data.

### 

Delete Asset

Main Success Scenario:

1. The user is authenticated by the system.
2. The user selects the delete asset button
3. The user selects either software or hardware from a drop-down box.
4. The user enters an asset ID into either the hardware or software form
5. The user selects delete asset

Extensions:

4a. The user enters an invalid asset ID

1. The user is prompted the asset ID is invalid and no data is shown on the screen.

4b. The user enters an invalid character in an input box

1. The last character entered is removed

4c. The user’s input is too long

1. The last character entered is removed

Create Asset

Main Success Scenario:

1. The user is authenticated by the system.
2. The user selects the create asset button
3. The user selects either software or hardware from a drop-down box.
4. The user enters data in either the hardware or software form
5. The user selects create asset

Extensions:

4a. The user enters an invalid character in an input box

1. The last character entered is removed

4b. The user’s input is too long

1. The last character entered is removed

5a. The user enters an invalid MAC address

1. The user is prompted that their MAC is invalid and the create request is not processed.

5b. The user enters an invalid IP address

1. The user is prompted that their IP is invalid and the create request is not processed.

Update Asset

Main Success Scenario:

1. The user is authenticated by the system.
2. The user selects the update asset button
3. The user selects either software or hardware from a drop-down box.
4. The user enters an asset ID into either the hardware or software form
5. The relevant data to the asset ID is displayed
6. The user can now modify the displayed data
7. The user selects update asset

Extensions:

4a. The user enters an invalid asset ID

1. The user is prompted the asset ID is invalid and no data is shown on the screen.

4b, 6a. The user enters an invalid character in an input box

1. The last character entered is removed

6b. The user’s input is too long

1. The last character entered is removed

7a. The user enters an invalid MAC address

1. The user is prompted that their MAC is invalid and the create request is not processed.

7b. The user enters an invalid IP address

1. The user is prompted that their IP is invalid and the create request is not processed.

Link Asset

Main Success Scenario:

1. The user is authenticated by the system.
2. The user selects the link assets button.
3. The user enters the asset IDs to link.
4. The user selects link assets.

Extensions:

3a. The user enters an invalid asset ID.

1. The user is prompted that they entered an invalid ID.

3b. The user enters an invalid character in an input box.

1. The last character entered is removed

4a. The assets are already linked.

1. The user is prompted that the assets are already linked.

Backup

Main Success Scenario:

1. The user is authenticated by the system.
2. The user selects the backup asset button.
3. The user is prompted that the backup was successful.

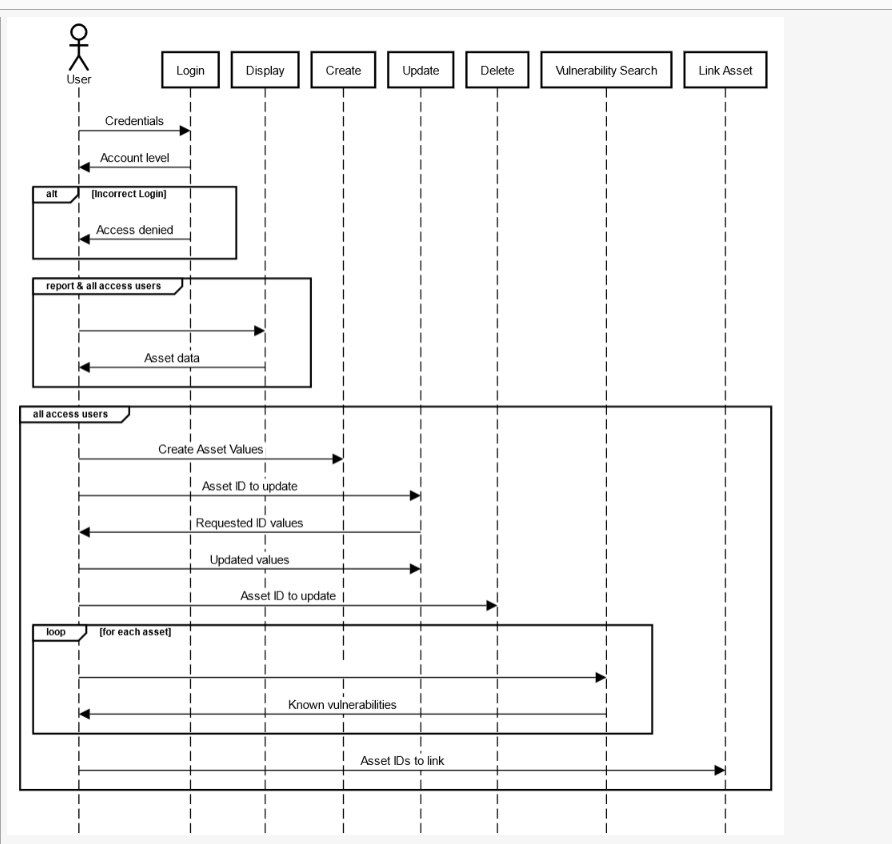
Extensions:

3a. The backup failed.

1. The user is prompted that the backup failed

## Class Diagrams

## Sequence Diagrams



## Additional Diagrams

# Implementation

## Development Methodology

## Implementation Patterns

# Testing

## Unit Testing

### Hardware & Software Asset Display

### Hardware Asset Create / Update / Delete

### Software Asset Create / Update / Delete

### Vulnerability API Call

### Database Backup

## Usability Testing

### Tester 1

### Tester 2

### Tester 3

# Reliability & Security

### Fault Avoidance

### Fault Tolerance

### Security Measures

## Account Credentials

### Full Access

### Report Access

# Software Guide

## Login

## Control Panel

## Hardware

### Create Hardware Asset

### Update Hardware Asset

### Delete Hardware Asset

## Software

### Create Software Asset

### Update Software Asset

### Delete Software Asset

## Display Assets

### Assets

### Vulnerability API Call

## Backup Database

### Finding Backup

## Logout

# Evaluation