Requirements Analysis Document

Configuration Management Information System (CMIS)

CSCI 4712 Senior Capstone Project

Fall 2018

Augusta University

Augusta, GA

Date: 11/18/2018

Version 3

Team Members

Evan Crane

Roy Njeru

Bilal Sellak

Abstract

This document contains the use cases for the Control Management Information System (CMIS). CMIS is a web application that will enable users to manage the status of several infrastructure components vital to the Information Technology network of a company. The users will be broken into three distinct categories which are Manager, Administrator (Admin), and Reader. Each user will be categorized by the system based on their login information.

This document describes the requirements, analysis and design of the CMIS system. The rest of this document is structured as follows. Chapter 1 contains the introduction. This chapter presents a brief description of the system. Chapter 2 outlines the use cases of the system.

Table of Contents

1 INTRODUCTION

1.1 Project Scope

The Configuration Management Information System(CMIS) plays a vital role in establishing and maintaining documentation of system information and system changes. CMIS contains the configuration of an information system(s) which represents the system's components, how each component is configured, and how the components are connected or arranged to implement the information system. System components consist of Servers, Databases, Server Instances, Applications, and Environments. Each with detailed information about the component, ownership, relationships, and sub-objects. As such, there is a need for an application that allows the creation of new entries and the management of pre-existing entries.

The CMIS application is a tool to manage the information detailed above from an administrative perspective. It is not a system that is connected or reading from any real-time devices or networks. The system is constrained to the main application, a database, and users. The actual servers, databases, server instances, etc. described in the system do not feed into CMIS nor does any information from CMIS reach the infrastructure.

1. 2 OVERVIEW OF DOCUMENT

The rest of the document is structured as follows. Chapter 2 outlines the use cases of the system. Along with the use case diagram, a list of use case descriptions accompanies each use case.

2 REQUIREMENTS OF SYSTEM

2.1 USE CASES

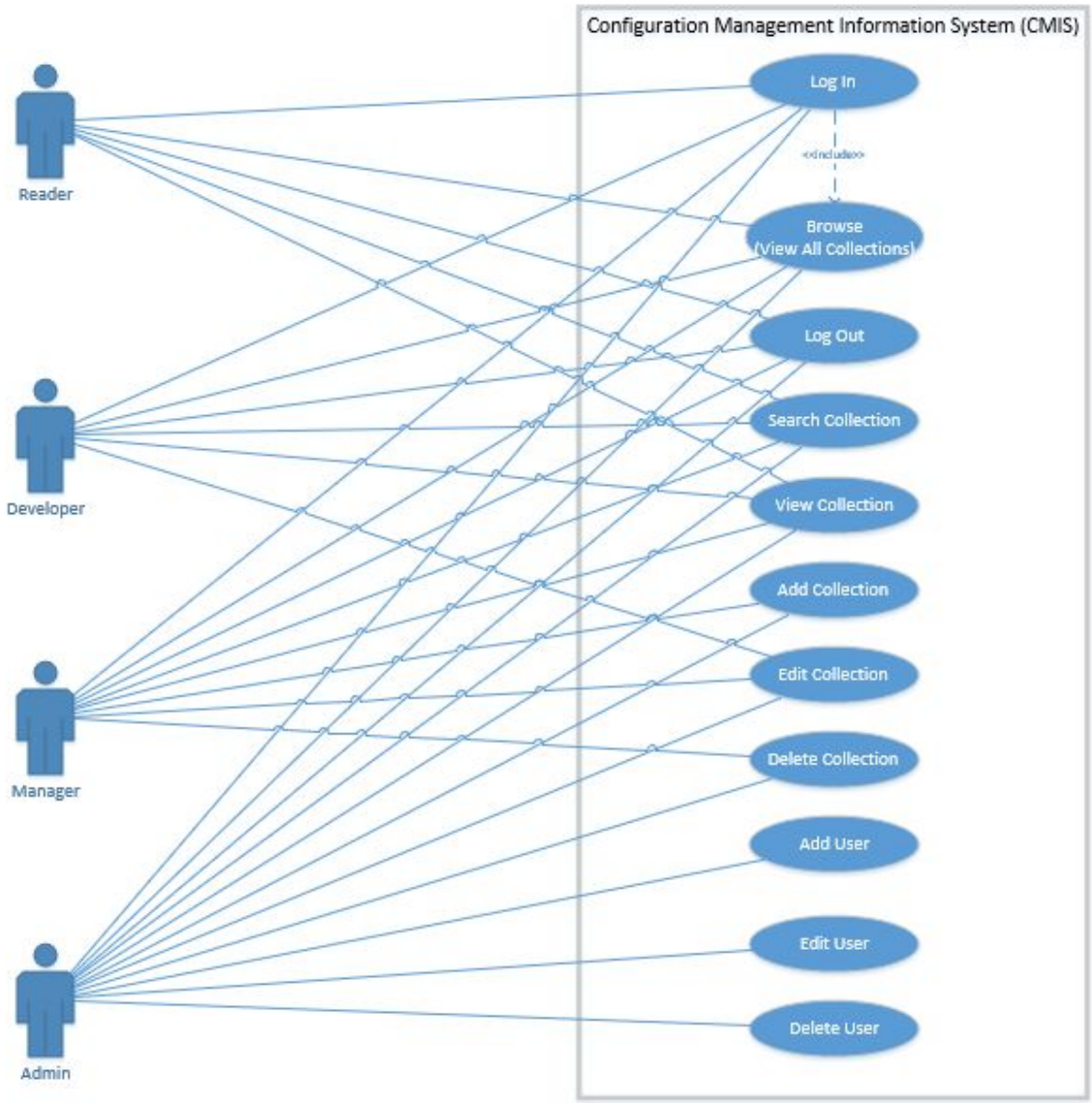

Figure 2.1: Use case diagram for CMIS

2.3 USE CASE DESCRIPTIONS

Use case name	LogIn
Participating actors	Initiated by Manager, Admin, Developer, and Reader
Flow of events	1. The user fills out the form by entering their credentials. Once completed, the user submits the form 2. **CMIS receives the form and verifies the credentials. The credentials are correct. System saves login history. System performs *BrowseCollections (View All Collections)* and presents the user with the home page.**
Entry condition	
Exit condition	<<includes>> BrowseCollection
Security requirements	Password should not be displayed as characters in the textbox during entry. User input is validated for appropriateness and prevention of SQL injection attacks.

Figure 2.2: LogIn: successful

Use case name	LogIn
Participating actors	Initiated by Manager, Admin, Developer, and Reader
Flow of events	1. The user fills out the form by entering their credentials. Once completed, the user submits the form **2. CMIS receives the form and verifies the credentials. The credentials are incorrect. The login screen displays a failed credentials message.**
Entry condition	
Exit condition	Login view is displayed with message to check log in information.
Security requirements	Password should not be displayed as characters in the textbox during entry. User input is validated for appropriateness and prevention of SQL injection attacks.

Figure 2.3: LogIn: unsuccessful

Use case name	BrowseCollections (View All Collections)
Participating actors	Initiated by Manager, Admin, Developer, and Reader
Flow of events	**1. CMIS presents all collections in a table with the collection attributes such as full name, organization, design agency organization, type and status listed. The default display will be 10 collections per page and listed in alphabetical order.**
Entry condition	Log in is performed.
Exit condition	
Security requirements	A valid user is logged in.

Figure 2.4: Browse Collection : Home Page

Use case name	BrowseCollections (View All Collections)
Participating actors	Initiated by Manager, Admin, Developer, and Reader
Flow of events	**2. CMIS presents all collections in a table with the collection attributes such as full name, organization, design agency organization, type and status listed. The default display will be 10 collections per page and listed in alphabetical order.** 3. The user manipulates the order of collections by toggling the alphabetical order of the collection categories. **4. CMIS changes the order of the collections in the table based on user input.**
Entry condition	Log in is performed.
Exit condition	
Security requirements	A valid user is logged in.

Figure 2.5: Browse Collection : Toggle collection categories.

Use case name	BrowseCollections (View All Collections)
Participating actors	Initiated by Manager, Admin, Developer, and Reader
Flow of events	**1. CMIS presents all collections in a table with the collection attributes such as full name, organization, design agency organization, type and status listed. The default display will be 10 collections per page and listed in alphabetical order.** 2. The user clicks on the next button, previous button, or table page number at the bottom of the collections table. **3. CMIS updates the collections shown in the table based on user input.**
Entry condition	Log in is performed.
Exit condition	
Security requirements	A valid user is logged in.

Figure 2.6: Browse Collection : Page by Page

Use case name	BrowseCollections (View All Collections)
Participating actors	Initiated by Manager, Admin, Developer, and Reader
Flow of events	**1. CMIS presents all collections in a table with the collection attributes such as full name, organization, design agency organization, type and status listed. The default display will be 10 collections per page and listed in alphabetical order.** 2. The user chooses to display either 10, 25, 50 or 100 collections at once. **3. CMIS updates the number of collections shown in the table based on user input.**
Entry condition	Log in is performed.
Exit condition	
Security requirements	A valid user is logged in.

Figure 2.7: Browse Collection : Show [#] Entries

Use case name	LogOut
Participating actors	Initiated by Admin, Manager, and Reader
Flow of events	1. This is function is activated when a user pushes the "Logout" button on the Home screen. 2. **CMIS presents the Logout Window popup with a prompt to confirm the wish to logout and a "Logout" button.** 3. The user selects the "Cancel" button. 4. **CMIS returns to the Home Screen that shows user still logged in.**
Entry condition	Valid user is logged in
Exit condition	Login screen is displayed.
Security requirements	

Figure 2.8: LogOut: cancel

Use case name	LogOut
Participating actors	Initiated by Admin, Manager, and Reader
Flow of events	1. This is function is activated when a user pushes the "Logout" button on the Home screen. **2. CMIS presents the Logout Window popup with a prompt to confirm the wish to logout and a "Logout" button.** 3. The user selects the "Logout" button. 4. **CMIS discards all non-saved information, saves logout data to database, and returns to the initial Login Form**
Entry condition	Valid user is logged in
Exit condition	Login screen is displayed.
Security requirements	

Figure 2.9: LogOut: successful

Use case name	SearchCollections
Participating actors	Initiated by Manager, Admin, Developer and Reader.
Flow of events	1. User inputs a search term in a search bar, located at the top right of the collections table on the home page. **2. CMIS responds by filtering the collections table and presenting a list of 1 or more search results that matched the search term as the user types in the search bar. Search collection is based on the attributes in the collection table.** 3. The user clicks on a collection name in the filtered collections table. **4. CMIS directs the user to the corresponding "ViewCollection" page for the particular user.**
Entry condition	Any user is logged in
Exit condition	"ViewCollection" use case is activated for the chosen collection.
Security requirements	Valid user is logged in.

Figure 2.10: Search collection: proceed to collection page

Use case name	SearchCollections
Participating actors	Initiated by Manager, Admin, Developer, and Reader
Flow of events	1. User inputs a search term in a search bar, located at the top right of the collections table. **2. CSIM responds by presenting an empty result table.**
Entry condition	Any user is logged in
Exit condition	A message indicating that no search results were found.
Security requirements	A valid user is logged in.

Figure 2.11: Search collection: none found

Use case name	ViewCollection
Participating actors	Initiated by Manager, Admin, Developer, or Reader.
Flow of events	1. The "ViewCollection" function is activated when a user clicks on a collection full name on the home screen. **2. CMIS responds by presenting the collection page for the specific collection clicked. The collection page will have all the attributes of the specified collection.**
Entry condition	User is logged in as a Manager, Admin, Developer or Reader.
Exit condition	User will be taken to the collection page.
Security requirements	

Figure 2.12: View Collection

Use case name	AddCollection
Participating actors	Initiated by Manager and Admin
Flow of events	1. The "AddCollection" function is activated when a Manager or Admin clicks on the *"Add Collection"* button. **2. CMIS responds by presenting the add collection form to the user. The add collection form contains the Collection Type, Acronym or Short Name, Full Name, Organization as well as other collection attributes.** 3. The user fills out the form by entering values for the collection attributes. Then presses the submit button to save the collection. **4. CMIS receives the form and verifies the collection. The collection is correct. The collection is displayed on the collection page.**
Entry condition	User is logged in as a Manager or Admin
Exit condition	User will be taken to the collection page.
Security requirements	All sql queries for this use case should be generated in a manner that prevents sql injection attacks.

Figure 2.13 Add collection: valid submission

Use case name	AddCollection
Participating actors	Initiated by Manager and Admin
Flow of events	1. The "AddCollection" function is activated when a Manager or Admin clicks on the *"Add Collection"* button. **2. CMIS responds by presenting the add collection form to the user. The add collection form contains the Collection Type, Acronym or Short Name, Full Name, Organization as well as other collection attributes.** 3. The user fills out the form by entering values for the collection attributes. Then presses the submit button to save the collection. **4. CMIS receives the form and verifies the collection. The collection is incorrect. The home page is displayed with an error message.**
Entry condition	User is logged in as a Manager of Admin
Exit condition	Add collection form is displayed with invalid collection attribute entries
Security requirements	All sql queries for this use case should be generated in a manner that prevents sql injection attacks.

Figure 2.14: Add Collection: invalid submission

Use case name	EditCollection
Participating actors	Initiated by Managers, Developers or Admins
Flow of events	1. This function is activated when a user selects the *"Edit Collection"* button next at the bottom right of the collection page. **2. CMIS responds by fetching the collection form that is populated with the attributes for the selected collection.** 3. The user fills out the form by entering new attributes or editing existing values for attributes. Once the form is complete, the user submits the form. **4. CMIS receives the form and verifies the attributes. The form is correct.**
Entry condition	ViewCollection use case has been performed.
Exit condition	Updated collection is displayed.
Security requirements	All sql queries for this use case should be generated in a manner that prevents sql injection attacks.

Figure 2.15: Edit collection: valid submission

Use case name	EditCollection
Participating actors	Initiated by Managers, Developers or Admins
Flow of events	1. This function is activated when a user selects the *"Edit Collection"* button next at the bottom right of the collection page. **2. CMIS responds by fetching the collection form that is populated with the attributes for the selected collection.** 3. The user fills out the form by entering new attributes or editing existing values for attributes. Once the form is complete, the user submits the form. **4. CMIS receives the form and verifies the attributes. The form is incorrect**
Entry condition	User is logged in as a Manager, Developer, or Admin
Exit condition	Edit collection form is displayed with incorrect inputs
Security requirements	All sql queries for this use case should be generated in a manner that prevents sql injection attacks.

Figure 2.16: Edit collection: invalid submission

Use case name	DeleteCollection
Participating actors	Initiated by Manager and Admin
Flow of events	1. The user selects the Delete button on a Collection page **2. CMIS responds by presenting a Confirm Delete prompt to the user.** 3. The user selects the "Delete" button. **4. CMIS removes Collection data from database, and returns to the initial Home Screen.**
Entry condition	"View Collection" is performed.
Exit condition	
Security requirements	

Figure 2.17: Delete collection -confirm

Use case name	DeleteCollection
Participating actors	Initiated by Manager and Admin
Flow of events	1. The user selects the Delete button on a Collection page **2. CMIS responds by presenting a Confirm Delete prompt to the user.** 3. The user selects the "Cancel" button. **4. CMIS closes the confirm modal and does not remove the collection from the Database.**
Entry condition	"View Collection" is performed.
Exit condition	Search page displayed to user.
Security requirements	

Figure 2.18: Delete collection - cancel

Use case name	AddUser
Participating actors	Initiated by Admin
Flow of events	1. An admin clicks on the "Add User" button on the "Manage Users" page. **2. CMIS presents a the "Add User" modal with options for the user name, organization, access level, and password.** 3. The Admin inputs the values for the new user. **4. CMIS accepts the user input, closes the modal and refreshes the Manage Users page with the new user added to the list of CMIS users.**
Entry condition	User is logged in as an Admin
Exit condition	Updated list of users is presented to the Admin.
Security requirements	All sql queries for this use case should be generated in a manner that prevents sql injection attacks.

Figure 2.19: AddUser - successful

Use case name	AddUser
Participating actors	Initiated by Admin
Flow of events	1. An admin clicks on the "Add User" button on the "Manage Users" page. **2. CMIS presents a the "Add User" modal with options for the user name, organization, access level, and password.** 3. The Admin inputs the values for the new user. **4. CMIS rejects the user input and tells the user to review their input.**
Entry condition	User is logged in as an Admin.
Exit condition	
Security requirements	All sql queries for this use case should be generated in a manner that prevents sql injection attacks.

Figure 2.20: AddUser - unsuccessful

Use case name	AddUser
Participating actors	Initiated by Admin
Flow of events	1. An admin clicks on the “Add User” button on the “Manage Users” page. **2. CMIS presents a the “Add User” modal with options for the user name, organization, access level, and password.** 3. The Admin clicks the “cancel” button before submitting the modal. **4. CMIS does not attempt to save the information in the modal.**
Entry condition	User is logged in as an Admin.
Exit condition	
Security requirements	All sql queries for this use case should be generated in a manner that prevents sql injection attacks.

Figure 2.21: AddUser - cancel

Use case name	EditUser
Participating actors	Initiated by Admin
Flow of events	1. An admin clicks on a user on the "Manage Users" page. **2. CMIS presents a the "Edit User" modal with the values for the user name, organization, and access level filled in for the user they chose.** 3. The Admin edits the user information they desire and submits the information. **4. CMIS accepts the user input, closes the modal and refreshes the Manage Users page with the new user added to the list of CMIS users.**
Entry condition	User is logged in as an Admin.
Exit condition	
Security requirements	All sql queries for this use case should be generated in a manner that prevents sql injection attacks.

Figure 2.22: EditUser - successful

Use case name	EditUser
Participating actors	Initiated by Admin
Flow of events	1. An admin clicks on a user on the "Manage Users" page. **2. CMIS presents a the "Edit User" modal with the values for the user name, organization, and access level filled in for the user they chose.** 3. The Edits the user information they desire and submits the information. **4. CMIS rejects the user input and tells the user to review their input.**
Entry condition	User is logged in as an Admin.
Exit condition	
Security requirements	All sql queries for this use case should be generated in a manner that prevents sql injection attacks.

Figure 2.23: EditUser - unsuccessful

Use case name	EditUser
Participating actors	Initiated by Admin
Flow of events	1. An admin clicks on a user on the "Manage Users" page. **2. CMIS presents a the "Edit User" modal with the values for the user name, organization, and access level filled in for the user they chose.** 3. The user closes the modal by clicking the "close" button before submitting the modal. **4. CMIS does not update the user information for the selected user in the modal.**
Entry condition	User is logged in as an Admin.
Exit condition	
Security requirements	All sql queries for this use case should be generated in a manner that prevents sql injection attacks.

Figure 2.24: EditUser - cancel

Use case name	DeleteUser
Participating actors	Initiated by Admin
Flow of events	1. The Admin clicks on the “Delete” button next to a user’s name in the “Manage User” page. **2. CMIS presents the Admin with a modal asking them to confirm if they want to delete the user.** 3. The user confirms they want to delete the user. **4. CMIS proceeds to delete the user from the database.**
Entry condition	The user is logged in as an admin.
Exit condition	
Security requirements	All sql queries for this use case should be generated in a manner that prevents sql injection attacks.

Figure 2.25: DeleteUser - successful

Use case name	DeleteUser
Participating actors	Initiated by Admin
Flow of events	1. The Admin clicks on the “Delete” button next to a user’s name in the “Manage User” page. **2. CMIS presents the Admin with a modal asking them to confirm if they want to delete the user.** 3. The user cancels the process by clicking the “cancel” button on the modal. **4. CMIS proceeds to close the modal and no changes are made to the database.**
Entry condition	A user is logged in as an Admin.
Exit condition	
Security requirements	All sql queries for this use case should be generated in a manner that prevents sql injection attacks.

Figure 2.26: DeleteUser - cancel