

VCC Standardized Patient File Management System

Project Requirements and Specifications

Washington State University Elson S. Floyd College of Medicine



Darya Simoneaux

Anise Casebolt

Issayas Yohannes

TABLE OF CONTENTS

I.	INTRODUCTION	3
II.	SYSTEM REQUIREMENTS SPECIFICATION	3
II.1.	USE CASES	3
II.2.	FUNCTIONAL REQUIREMENTS	6
II.2.1.	<i>Script Access</i>	6
II.2.2.	<i>Script Management</i>	7
II.2.3.	<i>Actor Management</i>	8
II.3.	NON-FUNCTIONAL REQUIREMENTS	8
III.	SYSTEM EVOLUTION	9
IV.	GLOSSARY	10
V.	REFERENCES	11

I. Introduction

The Elson S.Floyd College of Medicine Virtual Clinic Center (VCC) is a simulation-based training center, allowing medical students to practice with manikins as well as real people acting out example patient situations. This is called Standardized Patient Simulation. The interactions simulate experience for medical students without the high stakes of a real patient. The actors involved in these scenarios receive information on the patient they are portraying through documents known as scripts. Scripts contain a plethora of important information for the actors, the students, as well as the facilitators of the simulation. The actor learns what symptoms they should portray and how to talk about them, as well as how their character is feeling. The scripts also contain door notes, which the student will view before participating in the simulation. The door notes mimic, again, a real situation in which the doctor would be given some information on the patient and their background before entering the room to see the patient.

With the importance that the scripts play in the VCC, the file management system should be able to keep up with the fast pace necessary in medical education. The current system for creating, editing, finding, and reusing scripts is inefficient. The VCC is hoping for a file system that will provide easy access to scripts, allow searching and editing, generate information such as door notes from the scripts, and increase general efficiency. The objectives as outlined by the VCC are to create a standardized script form, be able to access information from the scripts, be able to easily update the scripts each year, and be able to search for the scripts.

II. System Requirements Specification

II.1. Use Cases

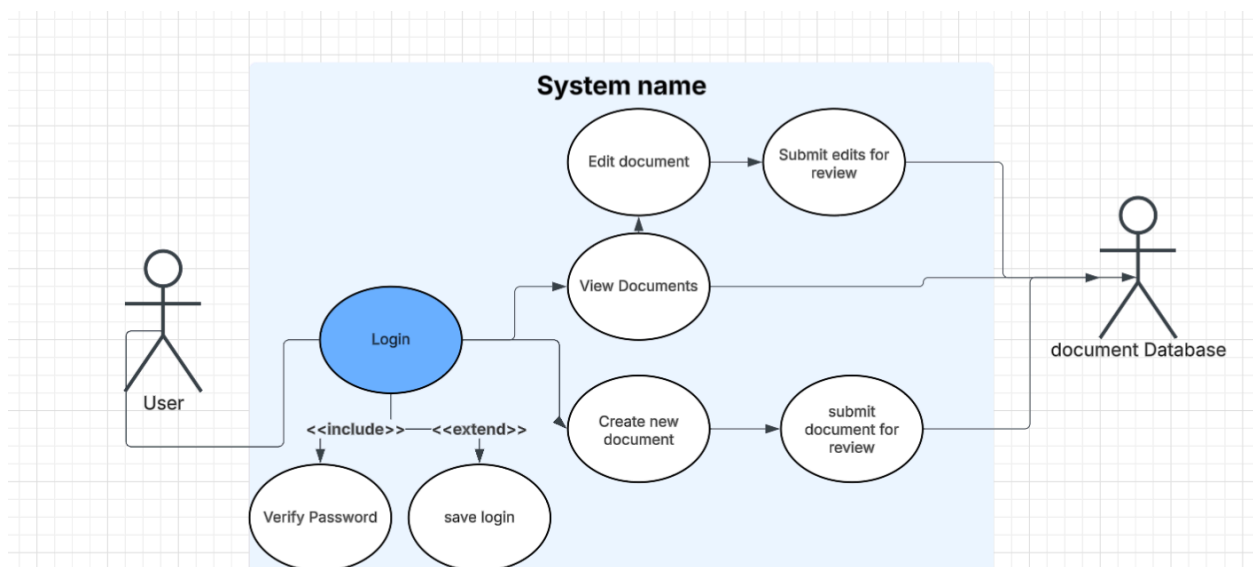


Figure 1. Use case diagram explaining how an instructor would interact with the website

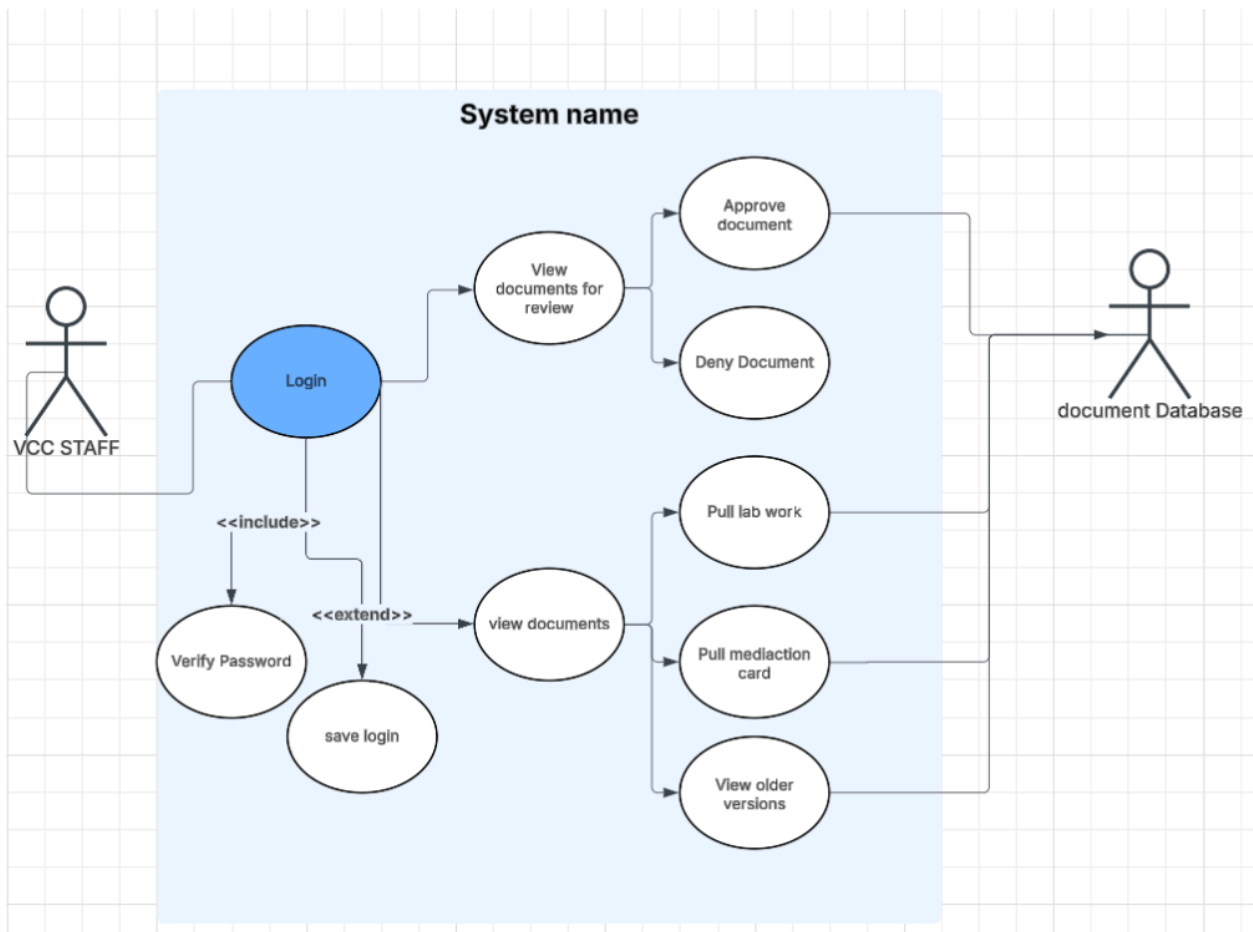


figure 2. Use case diagram explaining how a VCC staff member will interact with the website

UC-1: edit Documents to meet VCC standards

Actor	Approved user
Pre condition	User is logged in
Post condition	User can view and edit scripts that didn't get accepted
Basic path	User clicks on profile User selects MY scripts User selects rejected scripts filter User is given a list of rejected scripts and option to edit them
Related Req	Login server Document database Database interfacing

UC-2: Upload new script

Actor	Approved User
Pre condition	User is logged in and on the home page
Post condition	User has submitted a new script for review
Basic path	User clicks "create new script button" User fills out appropriate text entry boxes User clicks submit for review button
Related Req	Login server Document database Database interfacing

UC-3: Approving or rejecting Scripts

Actor	Approved VCC staff
Pre condition	User is logged in and on the home page
Post Condition	Pending approval tag is removed added to main search
Basic Path	User clicks on the script search page User selects the "pending approval filter" User selects deny or approve button
Related Req	Login server Document database Database interfacing Search page

UC-4: printing lab results and medication cards

Actors	Approved VCC staff
Pre Condition	User is logged in and on the search page
Post Condition	User has printed out lab results and medication card

Basic Path	User locates the script they are looking for User clicks the print medication card button
Alternate path	From step 1 User clicks into document result User locates desired text box User clicks print button
Related Req	Login server Document database Database interfacing Search page

II.2. Functional Requirements

Priority Scale for Functional Requirements:

Priority Level 0: Essential and required functionality

Priority Level 1: Desirable functionality

Priority Level 2: Extra features or stretch goals

II.2.1. Script Access

FR-01 - Print Documents: The system shall allow signed in users to automatically print medicine cards and door notes from specified scripts.

Source: This is directly related to a requirement from the client.

Priority: 0

FR-02 - Find lab results: The system shall allow signed in users to pull lab results from specified scripts.

Source: This is directly related to a requirement from the client.

Priority: 0

FR-03 - Script Generation: The system shall allow users to retrieve entire scripts with the standardized format and font.

Source: This is directly related to a requirement from the client.

Priority: 0

FR-04 - Script Search: The system shall allow the user to search for scripts based on simulated conditions.

Source: This is directly related to a requirement from the client.

Priority: 1

Script Version Control: The system shall keep records of the previous versions of the script so that previous changes can be observed.

Source: This is directly related to a requirement from the client.

Priority: 1

Script Format Flexibility: The system shall export scripts to multiple formats (PDF, Word, plain text)

Source: This came from AI used to fill functionality gaps.

Priority: 2

II.2.2. Script Management

Script Creation: The system shall allow signed in users to create new scripts in a standardized format.

Source: This is directly related to a requirement from the client.

Priority: 0

Age update: The system shall automatically update the age of patients played by the same actors each year to keep ages consistent.

Source: This is directly related to a requirement from the client.

Priority: 1

Script Status: The system shall allow users to set script status (draft, review, approved, archived).

Source: This came from AI used to fill functionality gaps, and the clients liked the idea.

Priority: 2

Script Editing: The system shall allow users to edit with change tracking and comments.

Source: This came from AI used to fill functionality gaps, and is included as a stretch goal.

Priority: 2

Validate Script: The system shall validate required fields before allowing script finalization.

Source: This came from AI used to fill functionality gaps.

Priority: 2

II.2.3. Actor Management

Actor Search: The system shall allow the user to search previous patient actors.

Source: This is directly related to a requirement from the client.

Priority: 1

Store Actor Emails: The system should store previous actor emails.

Source: This is directly related to a requirement from the client.

Priority: 1

II.3. Non-Functional Requirements

Non Functional Requirement		Description
Security [NFR-01, NFR-06, NFR-08, NFR-10]		The system shall: (a) restrict database access to signed-in users, (b) be protected against common malicious attacks, (c) implement role-based access control with audit logging, and (d) support single sign-on (SSO) integration with university systems.
Performance [NFR-03, NFR-13]		The system shall fetch webpages and scripts in under 1 second, and handle files up to 50MB in size.
Scalability [NFR-04]		The system should handle at least 100 concurrent users.
Availability [NFR-11, NFR-12]		The system shall maintain 99.9% uptime during business hours. The system shall also support database backups with point-in-time recovery
Maintainability [NFR-02]		The system should allow future maintainers to implement outlook integration

Accessibility [NFR-05]		The system should adhere to WSU's accessibility requirements
Usability [NFR-15]		The system shall provide comprehensive error messages and help documentation
Compatibility [NFR-17]		The system shall maintain compatibility with major browsers (Chrome, Firefox, Safari, Edge)

III. System Evolution

The design of the system is based on several assumptions about hardware, software, and user needs. While these assumptions hold true at the time of development, changes in the technology or client requirements may introduce risks or necessitate future modifications. Identifying these anticipated change points avoid rigid design decisions and supports long term adaptability

1. Hardware Evolution

- a. Assumption: The system will run on modern Windows-based faculty computers with standard performance (8GB + RAM, stable internet).
- b. Risk Point: If the university upgrades to new hardware platforms, the system's performance and compatibility need
- c. Anticipated Change: Optimization for cross platform access could become necessary in the next 2-3 years.

2. Software Dependencies

- a. Assumption: The system integrates with Microsoft Outlook and university authentication systems.
- b. Risk Point: Future updates to Outlook APIs or changes in SSO protocols may require significant refactoring.
- c. Anticipated Change: The system should be built modularly to allow maintainers to swap or upgrade integration components without breaking core functionality

3. User Needs and Workflows

- a. Assumption: Faculty and simulation specialists will primarily use the system for script creation, management, and standardized patient coordination
- b. Risk Point: If future users such as students or administrators require additional access levels or workflows, role based permissions may need to expand

- c. Anticipated Change: Expansion of RBAC and reporting features to support broader user groups

4. Security and Compliance

- a. Assumption: Current security measures (sign in requirement, protection from common malicious attacks) meet institutional standards.
- b. Risk Point: If regulatory requirements change (HIPAA, FERPA, or IT policy change), additional compliance features may be required.
- c. Anticipated Change: Enhanced compliance monitoring and automated reporting.

5. Scalability and Performance

- a. Assumption: The system is designed for up to 100 concurrent users and file uploads of up to 50MB
- b. Risk Point: If adoption grows beyond the department (e.g., university-wide rollout), higher scalability thresholds may be required.
- c. Anticipated Change: Support for cloud based hosting, load balancing, and database sharding to handle larger workloads.

6. Accessibility

- a. Assumption: The system needs to meet Washington State University's accessibility requirements at launch.
- b. Risk Point: Accessibility standards may evolve, and user expectations for inclusive design may grow.
- c. Anticipated Change: Regular audits and adjustments to maintain compliance with evolving accessibility standards

IV. Glossary

CSS- A styling language used in web development to control the appearance of a website, including colors, fonts, layouts, and responsiveness. It works alongside HTML and JavaScript to make websites visually appealing and user friendly.

Go- An open-source programming language developed by Google. It is designed for building fast, scalable applications, particularly useful in backend services and systems where performance and concurrency are important.

IEEE- A professional association dedicated to advancing technology. In academic and professional contexts, IEEE is often referenced for its widely used standards, technical papers, and citation style

Webserver- A computer program or hardware device that delivers web pages to users through a browser. It processes requests (such as clicking a link) and responds with the correct content (such as text and images)

VCC- The Virtual Clinical Center (VCC) is the central point for all simulation-based training at the Elson S. Floyd College of Medicine. The VCC uses Standardized Patient Simulation, which involves individuals playing the role of a patient. These standardized patient encounters assist in the development and assessment of humanistic and clinical reasoning skills

MongoDB- A NoSQL database that stores information in flexible, JSON-like documents instead of traditional tables. It is often used for web applications because it can handle large amounts of unstructured data efficiently.

REST API- A set of rules that allows software applications to communicate over the web. A REST API provides access to data and services using standard HTTP methods (GET, POST, PUT, DELETE), making it easy for different systems to integrate.

Medication Card- A summary document that lists a patient's prescribed drugs, doses, schedules, and allergies. In medical simulations, these cards help students or participants quickly understand a patient's medication history and treatment context.

SharePoint- A Microsoft platform for document management and collaboration. It allows organizations to store, organize, and share files online, but without strict formatting rules, collections of documents can become disorganized or difficult to navigate.

V. References

- [1]“Virtual Clinical Center,” *Elson S. Floyd College of Medicine*, 2025.
<https://medicine.wsu.edu/about/departments-units/vcc/> (accessed Sep. 13, 2025).