# Technical Design Document

SECAM Systems
March 2010

## Executive Summary

SECAM's proposed Web Monitoring System is designed to provide a cheap, easy, and quick-response alternative, or extension, to the current video monitoring systems in the **ECS**. The SECAM System will provide each tenant with the ability to monitor their own office with only a web camera and the SECAM software provided. With this software, and a simple web camera, the SECAM System provides a motion-activated video monitoring system for a fraction of the price of other security camera alternatives.

The user will download the SECAM Secure application onto their office computer and create an account on the SECAM website. Once the account has been created, the user can link the SECAM Secure application to their account. Next, the user connects a web camera to their computer and allows the application to capture the video feed. The system is now set and when the user clicks the "monitor" button the system will provide a buffer time before the system is activated and motion detection begins. When motion is detected, the camera begins to record the video feed and saves the feed to a secure offsite database provided by SECAM. Also, the user may specify who will be notified via phone when motion is captured; such as local campus security and themselves.

The main goals of the SECAM System is to provide a cheap and secure alternative to the more expensive video monitoring methods currently in use.

## Document Purpose

This document specifies the steps and time required to develop the SECAM Web Monitoring System. This document also outlines the testing methods and cases to verify that the system satisfies the specifications. The resulting system should be extremely secure, rigid, and resistant to malicious attempts to corrupt data. From the user's perspective, the system should be easy to use, intuitive, and professional looking. This document will specify the various aspects of the SECAM Web Monitoring System's design and implementation; focusing specifically on;

- Introducing the system purpose and scope
- Specifying the main components of the system
- High-level diagrams describing the overall functionality of the system
- Breaking these main components down into modules that can be implemented
- The user interface design and description
- System use cases
- System Prototypes
- Software packages and languages used
- Functional and performance requirements
- Safety and emergency requirements and constraints
- How the system will be tested and detailed documentation for each test

# Table of Contents

# 1.0 Introduction

## 1.1 Purpose

The system will be designed to provide motion-detected video monitoring for small offices and spaces. When motion is detected, the system will notify the specified individuals of the event so that they can take action.

## 1.2 Scope

This system is not a replacement for any system already implemented. This system provides cheap security to the owners of offices where it would be too expensive to implement a full surveillance system. The system can be extended with sensors that send an on/off signal to the system.

## 1.3 Overview

The system will consist of three main software components:

- A secure database maintained by SECAM Systems
- A client application that runs on the client's computer
- A web interface that allows the user to view their video stream and change settings

These three components are the highest-level abstraction of the SECAM System. Along with the three main software components, the system requires a web camera and a computer to run on. Optionally, the system can utilize other sensors to extend its functionality. The system implementation will follow sound Software Engineering methods and documentation will be a requirement. The system will be constructed in modules, following the previously stated Software Engineering methods.

# 2.0 Design

## 2.1 General Overview

### 2.1.1 Main Components

The SECAM Web Monitoring System software is comprised of three main high-level components:

- SECAM **Secure** (A secure database system that is maintained by SECAM)
- SECAM **SysControls** (A client application that runs on the client's computer)
- SECAM **WebControls** (The web interface software that allows the user to view their data)

### 2.1.2 SECAM Secure

SECAM **SECURE** is the secure database that all recorded video is sent to. Once the data is sent to the database it is stored in a read-only state to prevent modification of video evidence. The actual database is an **Oracle** database with a shell program that provides an interface for incoming requests and outgoing transmissions. The databases will be stored at the SECAM data center locations where they are held in a **black box** state. The physical computers that contain the storage are locked in a cooled vault and are only accessible by security forces. When the doors to the vault are opened, a timestamp is automatically recorded with the identity of the person who opened the vault. This provides records of every entry and exit; which contributes greatly to the security and integrity of the system.

The hardware will consist of multiple server and storage computers in varying locations for redundancy purposes.

### 2.1.3 SECAM SysControls

SECAM **SysControls** is the client application that is installed on the user's computer. This application runs in the background and has a user interface when the user opens it. **SysControls** is the software that provides the motion detection processing and the encryption and sending of the stream to the SECAM **Secure** database. This program keeps a constant connection to the SECAM Secure database and reacts accordingly when the connection is lost. The user interface of **SysControls** displays a live feed of the web camera, some connection settings options, and a start and stop monitoring button. When the user clicks the monitor button, the software gives the user a certain amount of time to leave the room before the system activates.

### 2.1.4 SECAM WebControls

SECAM **WebControls** provides the interface for the web interface for the users. This software allows the display of a live video stream of the user's camera anywhere with an internet connection. Also, the user can modify the settings of the camera remotely from the web control interface. This software requires the user to log in with their SECAM account details.

## 2.2 Development Plan

SECAM Systems is comprised of many diverse independent components. The modular design this security system is to ensure security, integrity and to reduce system complexity. The diversity and independence of SECAM System components is both the systems greatest strength and greatest weakness. More specifically, as the quantity of independent system components increase, the more testing and verification must be done to ensure communication and protocols between these systems are correct, stable and verified. Erroneous communication between system components could lead to loss of data or loss of system integrity resulting in an inoperable system. The integrity of system data is vital to SECAM's business model, protection of our customers and clients. SECAM System's has an invested interest in developing the highest quality low cost security system on the market. To our product meets industry and corporate quality standards a development plan has been outlined followed by a testing plan. Clear and concise development and testing plans will ensure our security product will meet these expectations.

SECAM System development will be broken into a three stage process. SECAM development teams will use agile development practices with integrated testing procedures throughout the workflows to ensure deadlines and goals will be met on time. The development stages are outlined below.

The first stage of our development process is the development of SECAM **SysControls**. The **SysControls** application is the core business application. The operability of **SysControls** facilitates the recording, detecting and sending of sensitive video data. These primary use cases are the core of SECAM Systems. Without the efficient operation of this system application, all other components are rendered useless. Development for this stage will be broken down into several sub stages. The first stage will be to develop the application driver in charge of controlling, monitoring and reading data from a web camera. The second stage will be to develop the connection manager in charge of maintaining, encrypting and facilitating the transfer of data from **SysControls** to the SECAM **Secure** database. The final stage in development will be the remainder of the **SysControls** application including settings code, utility code and core application components. Integrated throughout this development process extensive testing will be done to reduce bugs, detect ways to improve system performance and verify system integrity.

The second stage of our development process is the development of **SyServe** and the SECAM **Secure** Database. These components are the second most important aspect to SECAM Systems. **SyServe** provides a robust library of algorithms and utilities to handle extensive video information while facilitating a secure socket driven encrypted communication with **SysControls** applications. The

SECAM **SECURE** database is essential for storage of user information, media information and maintaining integrity of stored data. Primary development in this stage will first focus on the development and testing of the SECAM **SECURE** database as the operability of the database is vital to **SysControls**. Once the SECAM **SECURE** database has been developed and extensively tested SECAM development teams will move on to the development of **SyServe**. The development of **SyServe** will be broken down into several stages. The first stage of development will focus on the of the communication protocol with **SysControls**. Extensive testing will be done to ensure that the protocol communication between these two systems cannot be compromised. The second stage of development is the development of compression algorithms and associated utility methods. The development of these application components are essential to ensuring the size of video file in the SECAM **SECURE** database rapidly and uncontrollably grows in size. Video files will be encoded and encrypted to ensure integrity of the data and file stored are optimally compressed to ensure quality and cost efficiency. The final stage of development will include the development of **SyServe** – SECAM **SECURE** integration and communication at the application layer and final application development for **SyServe**.

The final stage of development for SECAM System components is the development of SECAM **WebControls**. **WebControls** will use a model view controller design or (MVC). The MVC paradigm has been chosen to reduce complexity, increase security and increase efficiency of web application access. The development of our **WebControls** application will adopt an alternative development strategy. The first stage of development will focus on the Authentication controller module. The second stage in development for **WebControls** will involve the development of the Administrator controller module. The third stage of development will focus on the development of the User controller Module. The third stage will focus development the remaining utility classes and controller modules left in the system. The development order for WebControls is important as it allows SECAM developers to focus and test core system components and at the same time providing a framework and test bed to prepare for the development of critical controller modules.

Once all SECAM System components have been develop, development teams will focus on component integration testing followed closely by acceptance testing procedures.

## 2.3 System Design

### 2.3.1 Overview

The SECAM Security Camera Project is comprised of several important components. The project is broken into application components to ensure modularity, dynamic controls and security. The application components are the SECAM SysControls, the SECAM Web Controls and the SECAM SECURE database solution.

The SECAM SysControls is a natively installed java client that controls the locally installed webcam, video recording operations, streaming and notification. More specifically, SECAM SysControls provides webcam operations such as, monitoring sound to enable the camera to begin recording, initiating camera operations such as turning on and off and saving recorded data to the data base or streaming it to a mobile website.

The SECAM WebControls application is a web application designed with the purpose of providing a remote configuration and control tool. WebControls handles the modification of settings, the viewing of live camera video streams and the manipulation of the camera. Through WebControls you can remotely enable camera video streams, suppress alarm notifications and schedule camera operations.

The SECAM SECURE is our proprietary database design to store and host security camera recorded videos and associated video stream snapshots. The database is completely locked preventing the tampering of case sensitive data. Only municipal and Federal policing agencies may access this data ensuring legitimacy and validation of case sensitive video material. The database has been designed to ensure data integrity, security and efficiency. SECAM has developed an intelligent database solution that ensures the stored data is well organized, clean and secure without the human need for extensive maintenance.

SECAM SysControls is a light weight application that handles computer-camera-database operations. It can be broken down into some core classes designed to ensure low efficient memory usage and low CPU usage. This application is comprised of the classes, secam, api, connection, engine, settings and driver. Below is a table of classes, purposes and overviews:

SECAM WebControls is a robust web application designed to handle hand held live video streaming sophisticated SECAM SysControls maintenance, settings and customization. This web application has been designed to ensure security, intuitive application control and modularity. This application is comprised of the classes, ControllerAuthentication, ControllerApplication, ControllerAdministrator, ControllerUser, ControllerOperations, Notifier, Media, Status and Schedule classes. Below is a table of classes, purposes and overviews:

### 2.3.2 SysControls

#### *2.3.2.1 Class Diagram*

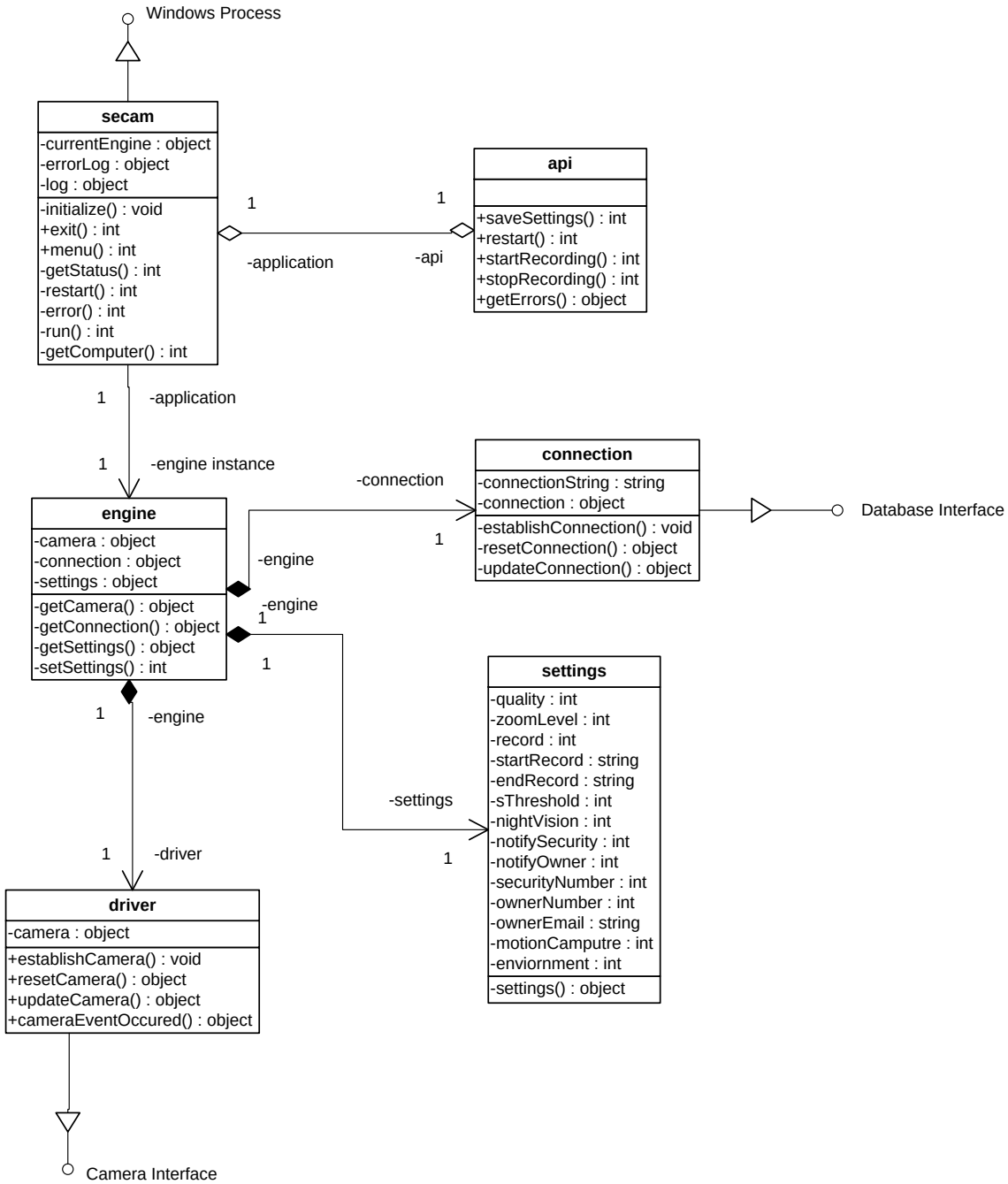


Figure 1: Class Diagram for SysControls

#### *2.3.2.2 Class Table*

The table below details each of the classes in SysControls and their purpose. This is a high-level description of each class to provide scope and context to the project.

| Class | Purpose | Overview |
|---|---|---|
| secam | • application entry point<br>• basic console UI<br>• application initialization<br>• data and error logging<br>• computer identification<br>• handles api calls | The secam class provides a entrance point for the application. Creates a process, generates logs, obtains computer network identification, holds the application instance and handles basic api calls. |
| api | • facilitates api calls<br>• executes commands from external layers | The api class is the layer used to communicate with the SECAM WebControls application facilitating external command execution. |
| connection | • establishes database connections<br>• re-establishes database connections<br>• updates database connections | The class connection is an important component in system security. The connection class is charged with maintaining a constant high speed connection to the SECAM SECURE database. |
| engine | • drives the SECAM SysControls application<br>• maintains status of application<br>• maintains settings and current configuration of the application<br>• maintains camera controls and configuration | The engine class is the primary controller for all SysControls systems. This class maintains and ensures all components are operational and properly configured if the system is updated. |
| settings | • holds data and settings instance of the application | The settings class holds all current runtime settings of the application. This instance object is created at initialization time and passed around the system to initialize various components. This instance is created from accessing the SECAM SECURE database at initialization time. |
| Driver | • holds camera api commands<br>• maintains security camera<br>• implements setting changes to camera<br>• sends control functions to the camera<br>• facilitates data transfer | The driver class is the primary class associated with managing and maintaining the security camera in an operational state. Furthermore the class also serves as a camera management tool making camera api calls and facilitating video recording. |

### *2.3.2.3 Object Class Diagram*

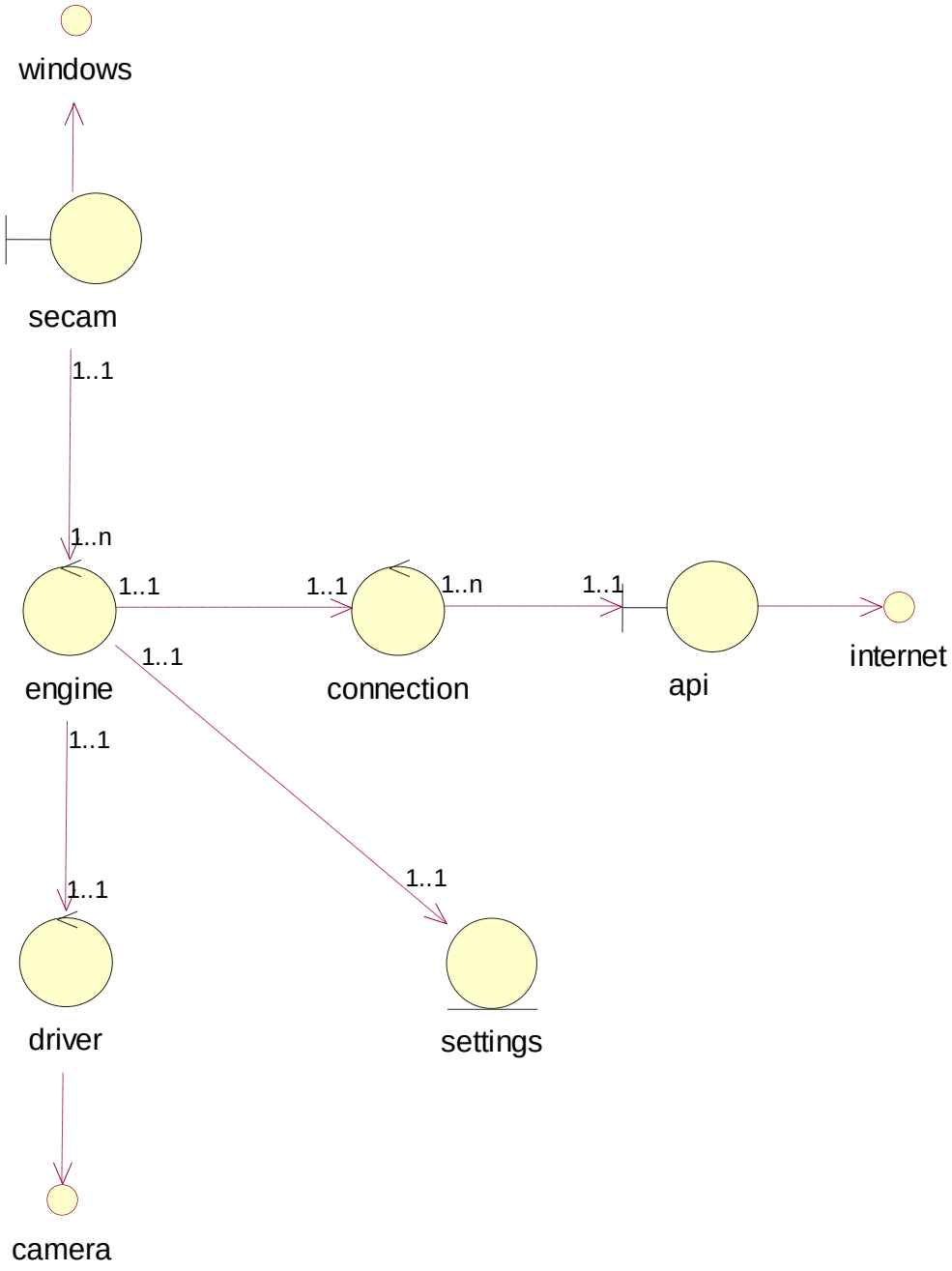


Figure 2: Object Class Diagram for SysControls

### *2.3.2.4 Initialization Sequence Diagram*

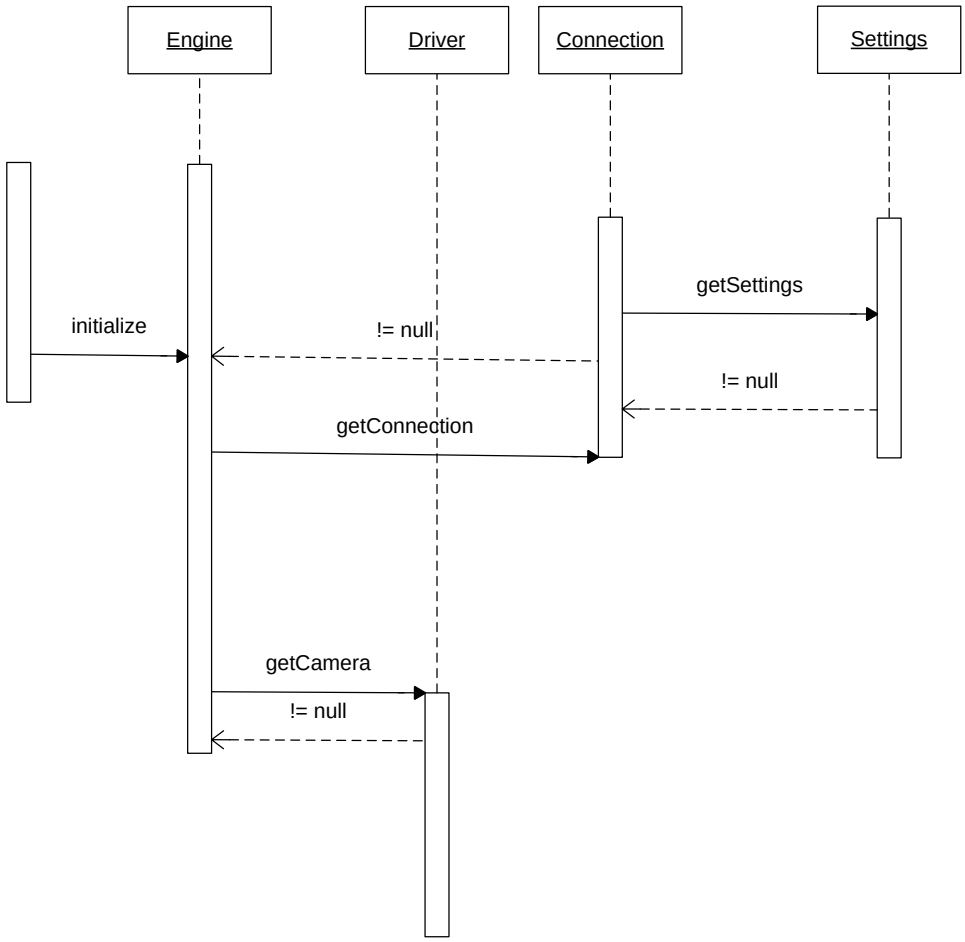


**Figure 3: Initialization Sequence Diagram for SysControls**

The above activation diagram outlines the object initialization calls for all of **SysControls** core features. This initialization procedure is critical to the operation of the **SysControls** application as this process determines vital system functions and status to ensure the application is configured and optimized correctly to satisfy operational requirements.

### *2.3.2.5 Video Streaming Sequence Diagram*

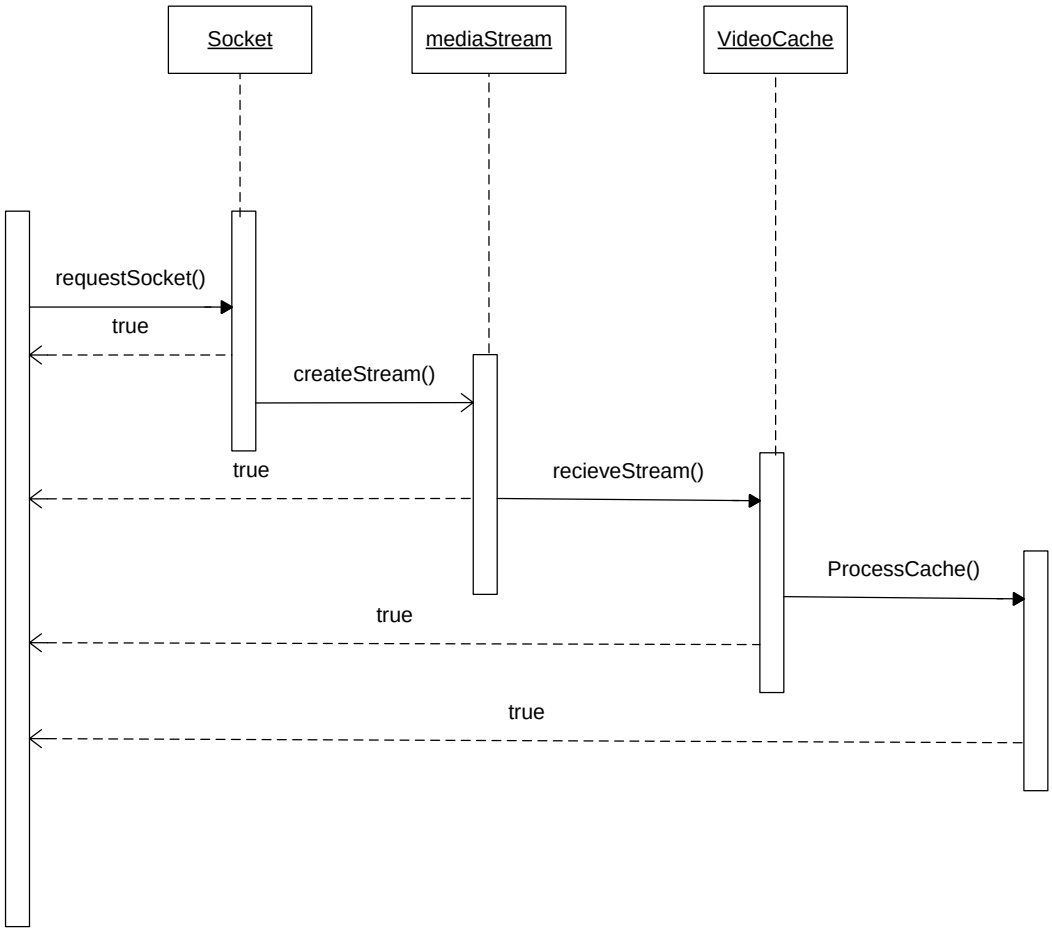


**Figure 4: Video Streaming Sequence Diagram for SysControls**

The above sequence diagram outlines the protocol and handshaking between **SyServe** and **SysControls**. This process is essential for ensuring sensitive data is not tampered with or the system or the SECAM security system is not compromised. The process can be summarized into several key steps. These steps are requesting a server socket to the database, creating and sending the stream and caching the video stream temporarily to be optimized, compressed and stored into the SECAM **SECURE** database.

### *2.3.2.6 Camera Activation Diagram*

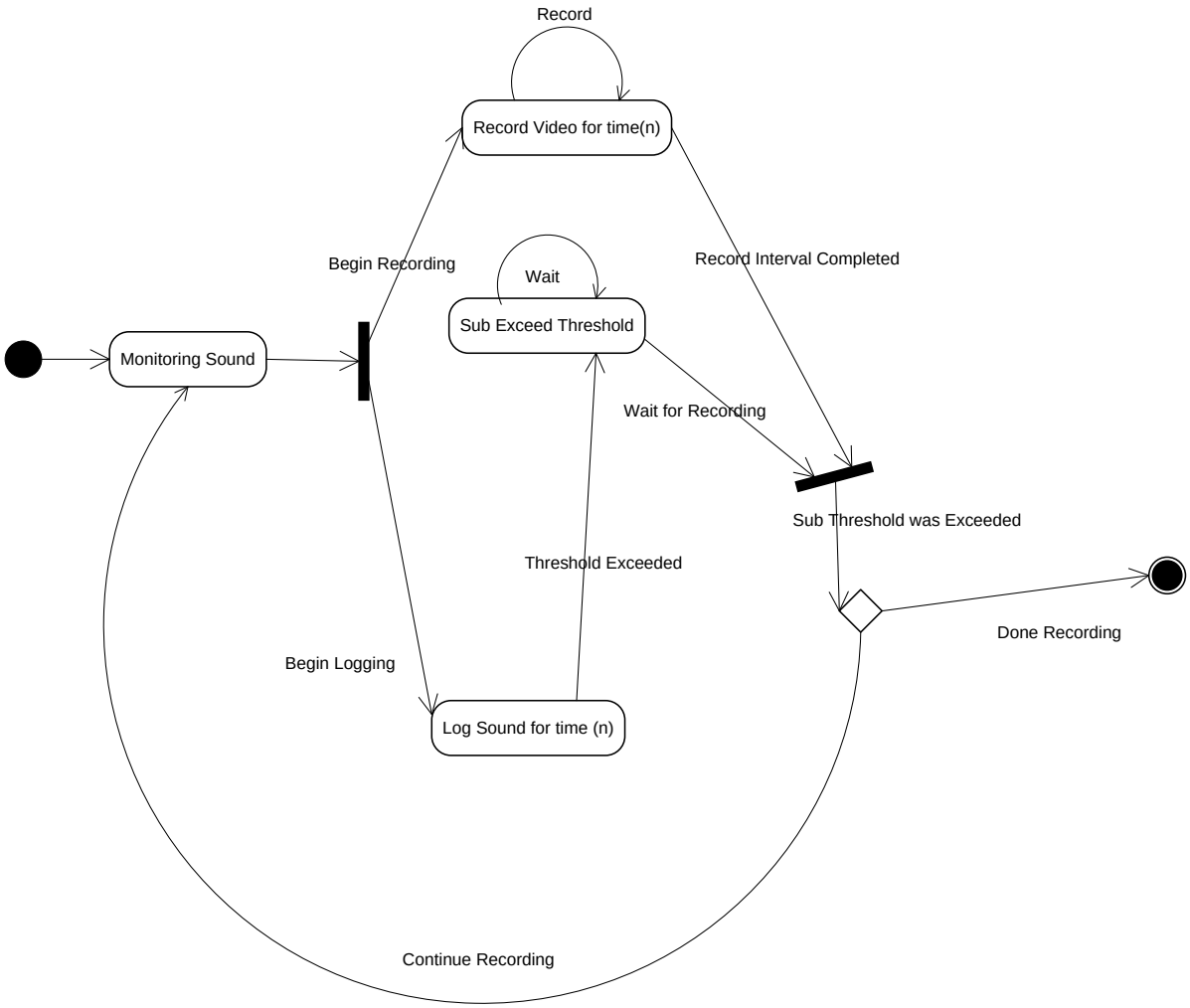


**Figure 5: Camera Activation Diagram for SysControls**

The above state chart diagrams the state process of camera activation and detection of intrusion.

### *2.3.2.7 Camera Activation State Diagram*

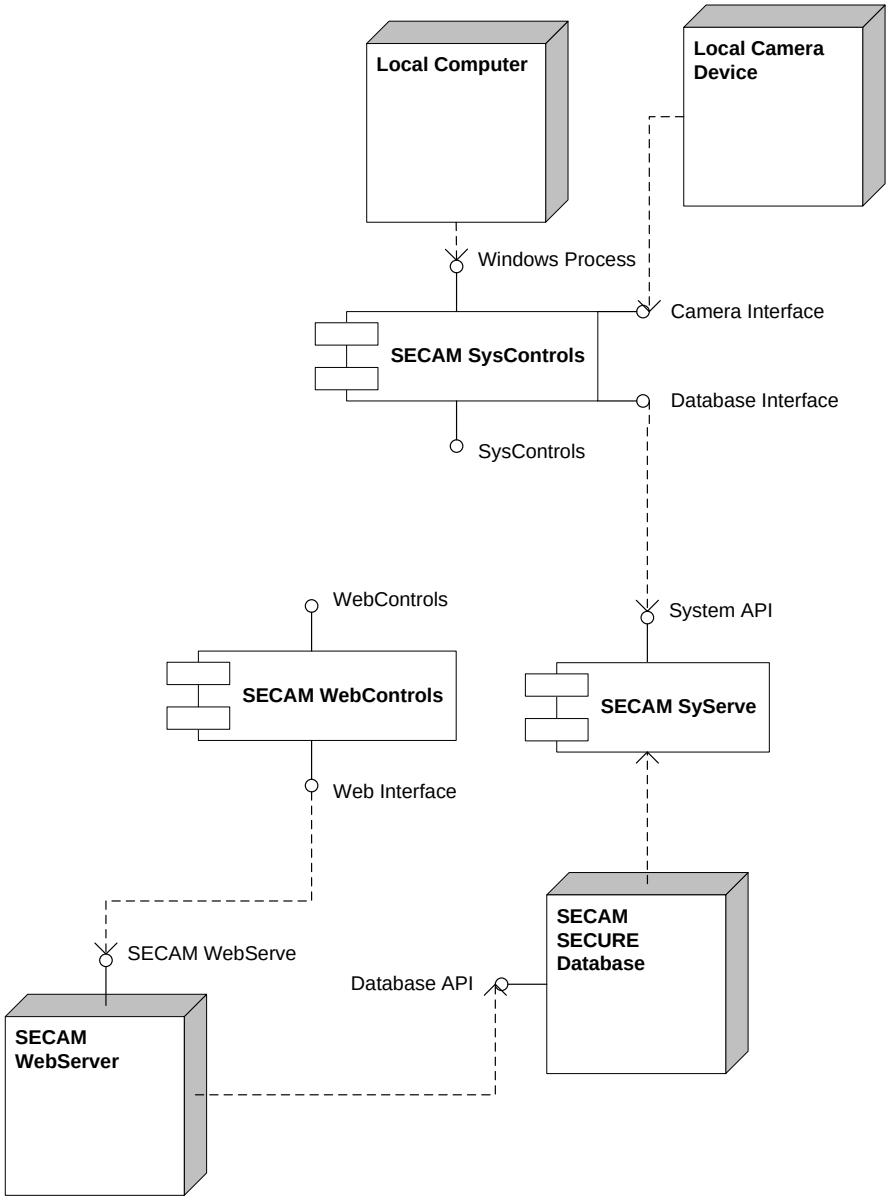


**Figure 6: Camera Activation State Diagram for SysControls**

The above diagram is an overview of all SECAM System assets and their association and interaction.

## 2.3.3 Web Controls

### *2.3.3.1 Class Diagram*

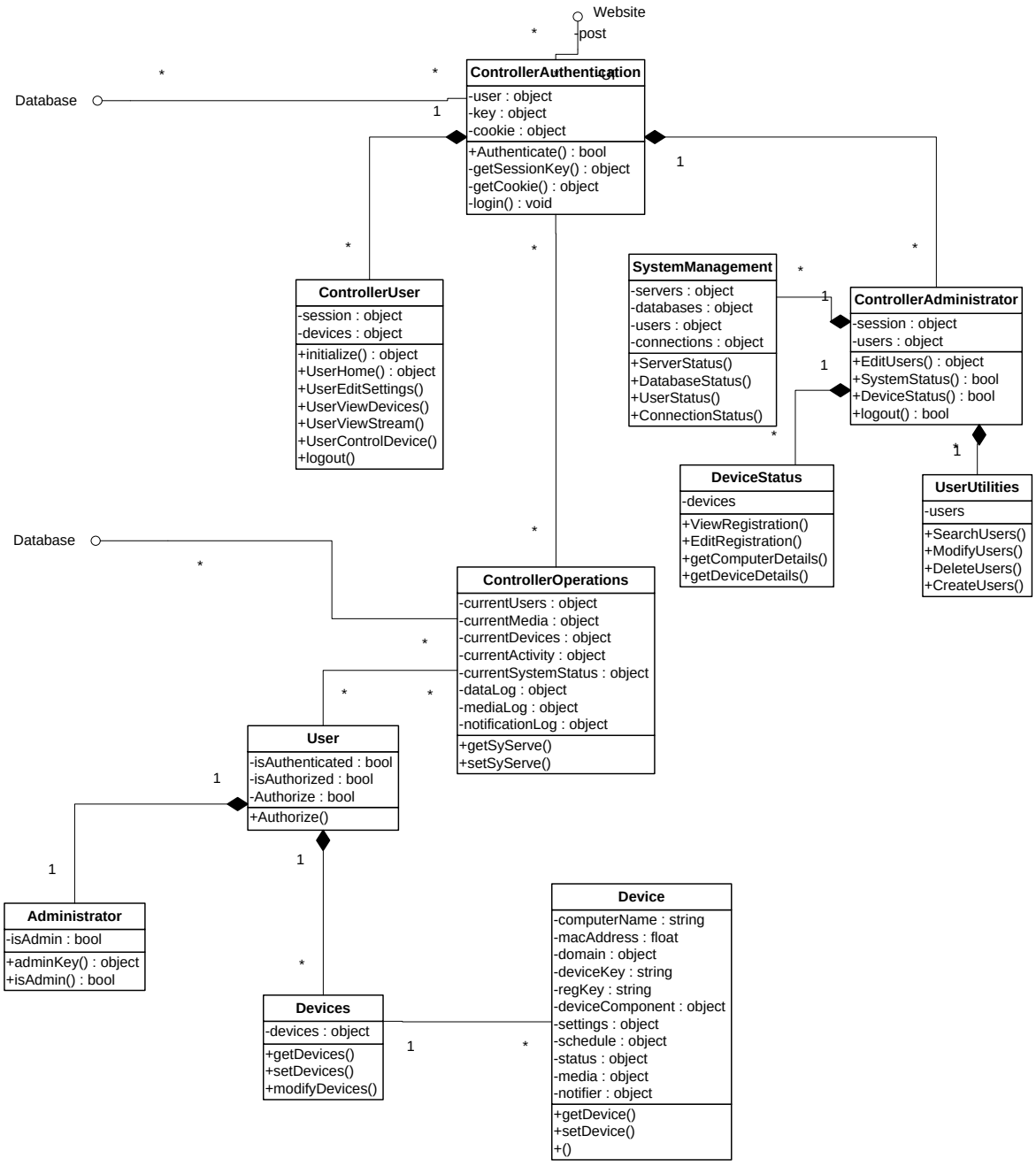


**Figure 7: Class Diagram for WebControls**

#### *2.3.3.2 Class Table*

The table below describes each of the classes in the Web Controls component of the system. This is a high-level description of each class to provide scope and context for the project.

| Class | Purpose | Overview |
|---|---|---|
| ControllerAuthentication | • handles authentication and validation operations | The ControllerAuthentication is responsible for maintaining and authenticating users to the system. |
| ControllerApplication | • handles SECAM SysControls communication<br>• API layer for internet based communication | The ControllerApplication class is responsible for handling SECAM SysControls requests. Specifically handling requests for saved settings or video streaming requests. |
| ControllerAdministrator | • handles UI request from website interface<br>• serves desired information to an Administrator<br>• handles database controls and queries | The ControllerAdministrator class is responsible for executing and handling administrator level web requests and administrator level interface functions. Specifically serving administrator level requests for data and modification. |
| ControllerUser | • handles UI request from website interface<br>• serves desired information to user<br>• handles database controls and queries | The ControllerUser class is responsible for executing and handling user level web requests and user level interface functions. Specifically serving user level requests for data and modification. |
| ControllerOperations | • handles a concurrent intermediate request handler from users and SECAM SysControls instances | The ControllerOperations class is responsible for implementing operations of all permissions level. Handles internal security measures and privacy settings. |
| Notifer | • handles multiple concurrent requests for notifications<br>• holds notification protocols and procedure information | The Notifier class is responsible for sending email, SMS and other notification features. The class is capable of generating formatted notification to user specifications and has intelligence to handle error severity and appropriate notification procedures. |
| Media | • handles media display, encryption, watermarking, streaming and other important media related security measures. | The Media class is responsible for all media, reading and writing operations associated with all SECAM applications. |

| | | |
|---|---|---|
| Status | • handles concurrent status updates for all active SECAM systems<br>• provides an efficient means for polling updates and propagating system information throughout all SECAM systems | The Status class is responsible for maintaining and handling internal status occurrences across all SECAM applications. |
| Schedule | • is responsible for maintaining scheduling information for SECAM systems<br>• handles scheduling calls for specific data requests | The Schedule class is responsible for handling and maintaining the signalling of scheduled events, or the current schedule for the all SECAM applications. |
| Settings | • handles setting information for all of SECAM's systems including facilitating<br>• handles SysControls settings<br>• handles internal system settings | The Settings class is responsible for maintaining handling and verify system settings. |

### 2.3.3.3 Object Class Diagram

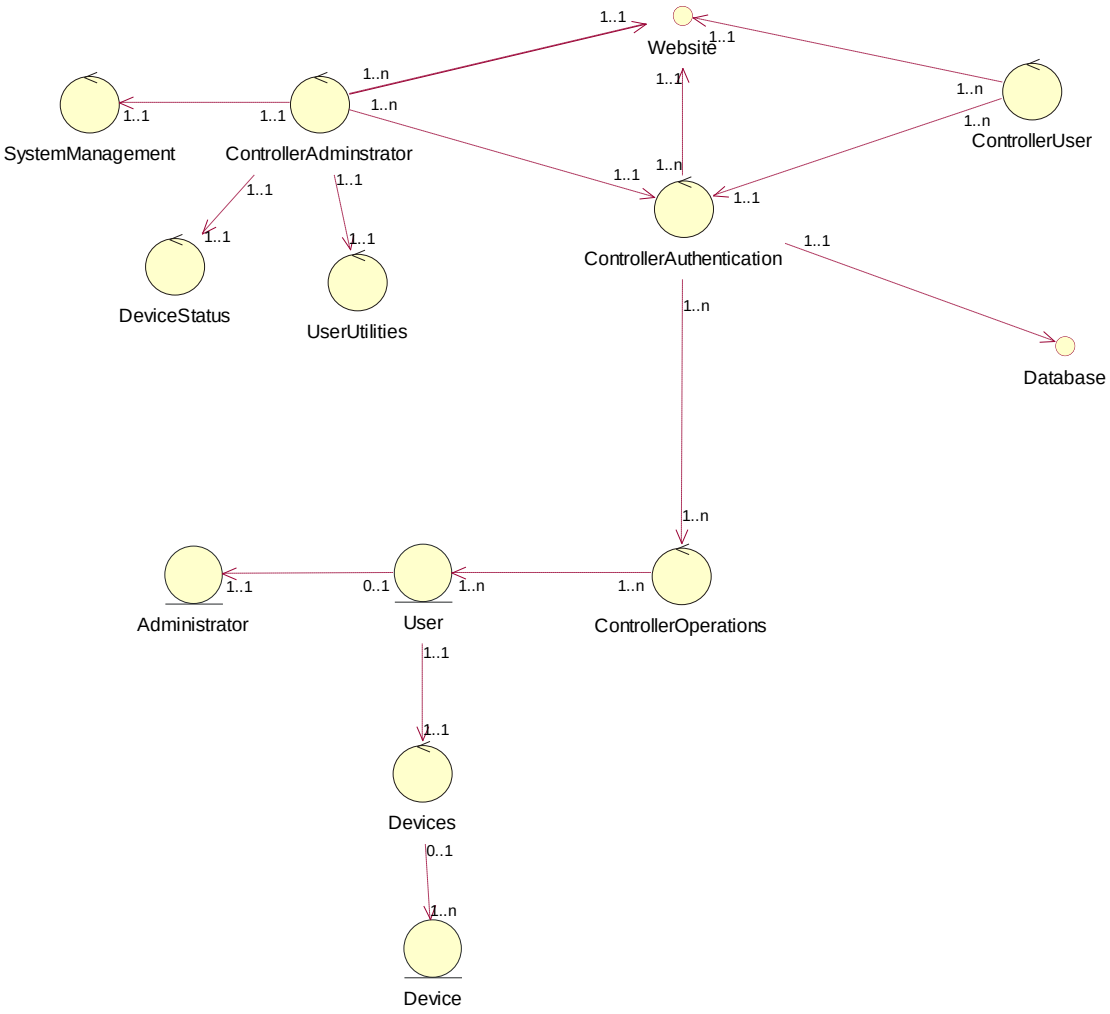


Figure 8: Object Class Diagram for Web Controls

### *2.3.3.4 Login Sequence Diagram*

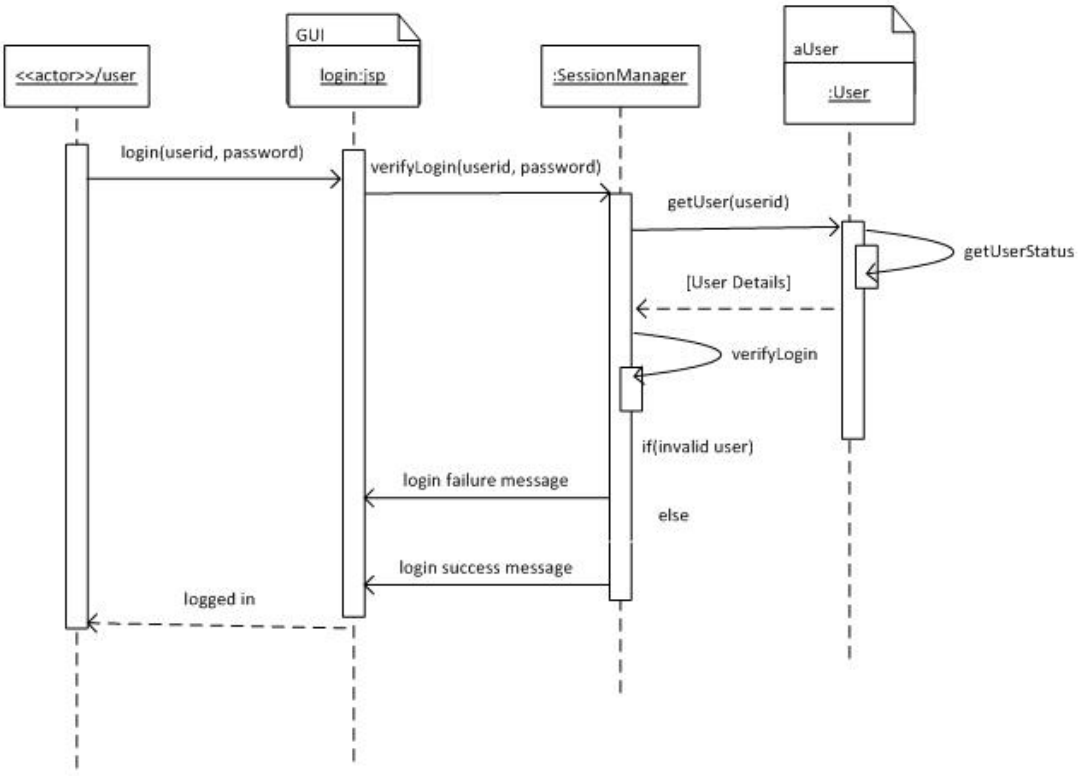


**Figure 9: Login Sequence Diagram for Web Controls**

### 2.3.4 SyServe and SECURE

#### *2.3.4.1 SECURE Entity Relationship Diagram*

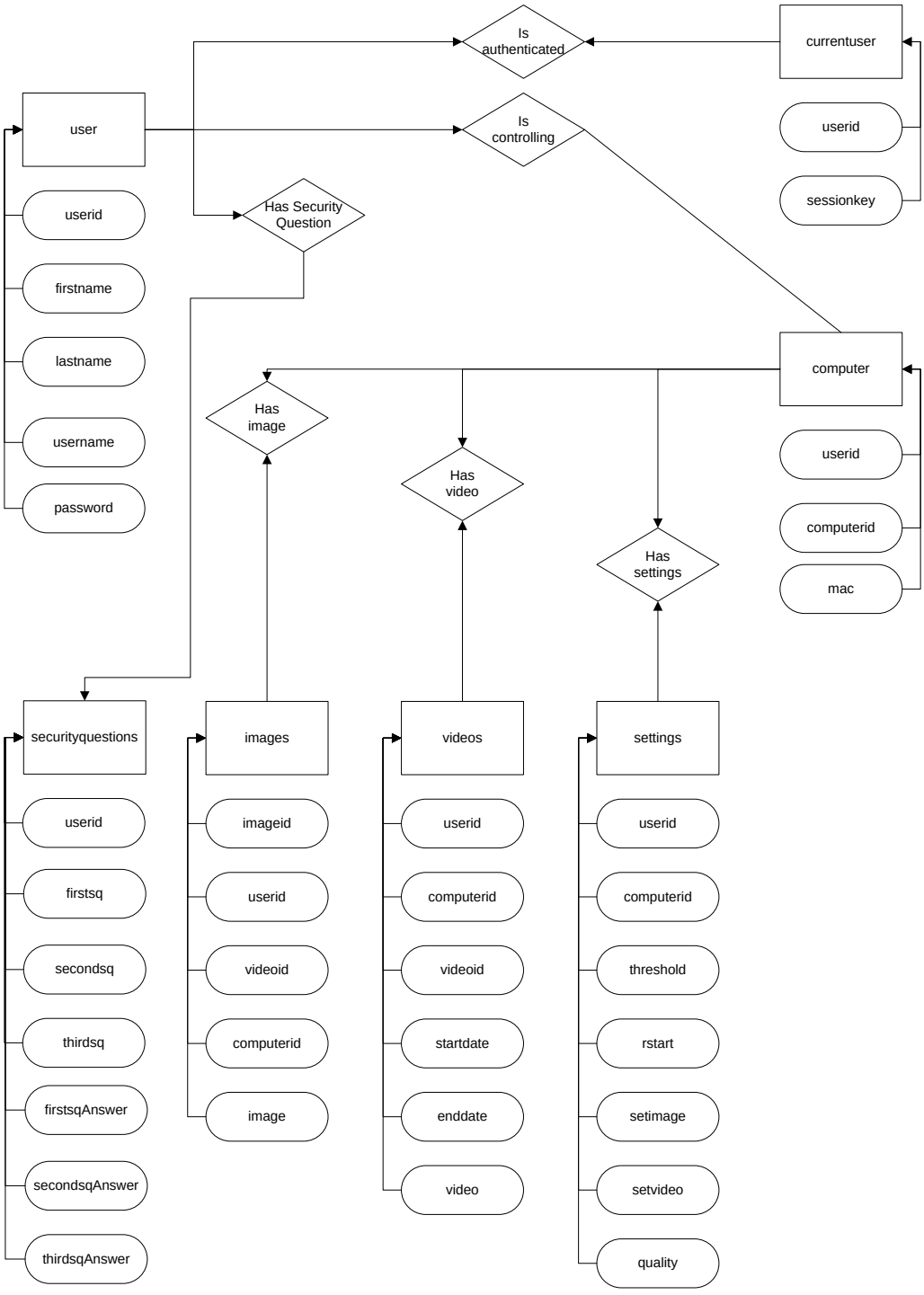


**Figure 10: SECURE ER Diagram for SyServe**

The above ER diagram portrays in generalize form the relations between entities in our SECAM Secure.

### *2.3.4.2 Validation Protocol Sequence Diagram*

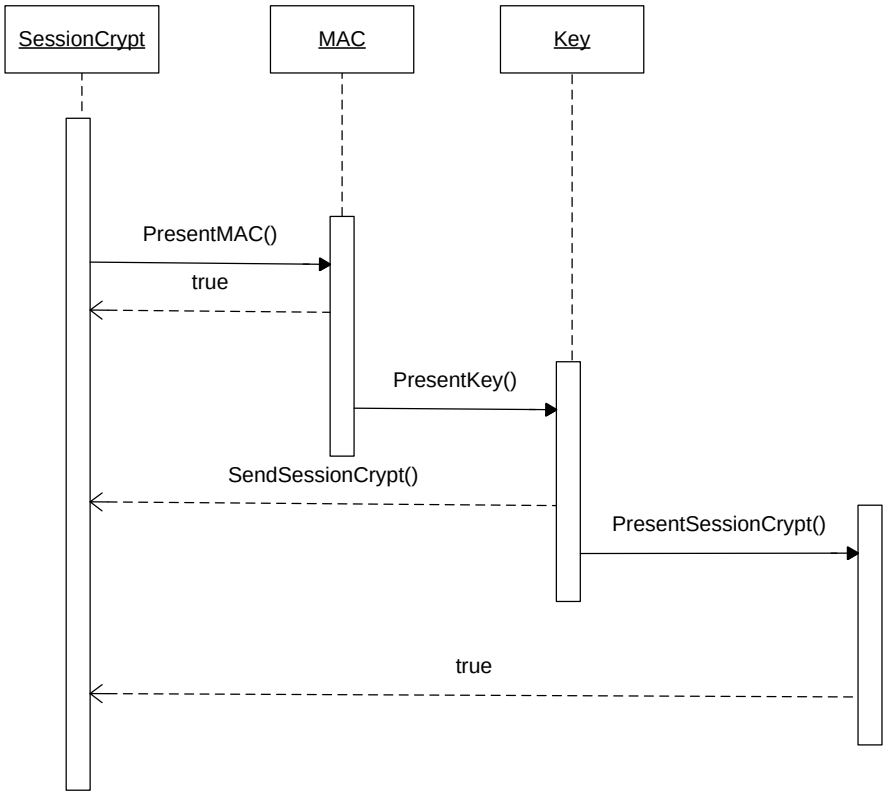


**Figure 11: Validation Protocol Sequence Diagram for Web Controls**

To ensure the integrity of the SECAM security system a process or protocol has been developed to ensure validity of incoming connections. This protocol prevents systems spoofing identification, pirating of a SECAM system and integrity of transmitted data. This protocol can be summarized into three primary stages as seen in the sequence diagram. The system is first presented with the MAC address of the computer which is crosschecked with the database. The requesting **SysControls** application then sends the registered key of the installation and the server **SyServe** responds by sending a session key for the current connection. The **SysControls** application then presents its session key every time data is transmitted. It should be noted that **SyServe** automatically detects if communication is coming from the correct IP address without handshaking with the localized **SysControls** application.

## 2.4 Modules

### 2.4.1 SysControls Modules

#### *2.4.1.1 Driver Module*

**Description:**

This is the module which deals with communication between the Desktop application and the actual camera. The driver is used by the desktop app to communicate with the underlying driver that manages the camera.

**Responsibilities:**

- Listens to system events to observes the state of camera connections
- Notifies the controller application when a camera is plugged on or off
- Maintains constant data stream from the camera to the controller application
- The data stream consists of two streams, video and sound
- Handles choosing between multiple connected cameras

**Communicates With:**

- SECAM engine module
- Camera interface

#### *2.4.1.2 Settings Module*

**Description:**

This module is responsible for moving settings between the application and the database.

**Responsibilities:**

- Gets the application settings from the database when the application starts
- Keeps setting data in the database synchronised

**Communicates With**

- Engine module

#### *2.4.1.3 Connection Module*

**Description:**

The connection module manages the connection between the **SysControl** application and the engine.

**Responsibilities:**

- Manage connections between modules

**Communicates With:**

- Is the communication manager. Talks with everything.

#### *2.4.1.4 Engine Module*

**Description:**

The engine module oversees the connection, settings and driver modules.

**Responsibilities:**

- Setting up connection with the database
- Initialising the camera and establishing a constant data stream with the camera
- Retrieving settings from the database and storing them in the database

**Communicates With:**

- Connection module
- Driver module
- Settings module

### 2.4.2 Web Controls Modules

#### *2.4.2.1 Controller Administrator Module*

**Description:**

This is the module which controls the actions of administrators within the system.

**Responsibilities:**

- Communicates with the database to perform user authentication during login
- Performing account management tasks including creation, deletion and editing
- Deletion deactivates the account unless the user explicitly specifies that they want their account permanently deleted. In which case their information and data is permanently erased from the database.
- Provides an interface to communicate with the desktop application in order to change user settings and manage the security devices.
- Manages computers that are registered with the system and the associated security cameras

**Communicates With:**

- Controller authentication

### *2.4.2.2 Controller Operations Module*

**Description:**

This controls the interaction between the website and the database.

**Responsibilities:**

- Handles user and admin authentication
- Handles getting device information and editing it
- Enables the user to view data from active cameras
- Handles system function logs
- Handles changing the system settings

**Communicates With:**

- The database (**SECURE**)

### *2.4.2.3 Controller Authentication Module*

**Description:**

This module authenticates the user when they login and oversees all operations of the web application.

**Responsibilities:**

- Creates a connection with the database
- Does login authentication for all users including administrators
- Authentication is done using IP address, MAC address and user credentials
- Creates user sessions
- Oversees all website modules

**Communicates With:**

- The database
- Underlying modules

# 3.0 Use Cases

## 3.1 Create an Account

**Use case name:** Create an account
**Scope:** System
**Primary actor:** SECAM Administrator
**Stakeholders and interests**: All the administrators of SECAM. An administrator wants to create an account for a customer (customer: UVic ECS staff and Security Personnel).
**Preconditions:** This primary actor has to be logged on at the SECAM facilities through a SECAM website
**Post conditions:** The customer's personnel information will be saved to the SECAM database. The system will send an email containing default password and username to the customer.

**Main Success Scenarios:**

1. Click manage accounts tab
2. Click create account button
3. Fill in the form
4. Click save button
5. Click yes button of the pop up message

**Extensions:**

- **5a.** If the form is incomplete; the actor will remain in the form page with incomplete important fields highlighted in red.
- Repeat 3, 4 and 5
- **5b.** Confirming the save action by clicking "no" to the pop up message will cause the system to remain in the same form page
- If the user does not want to create an account anymore, then the actor will click the cancel button

### 3.1.1 Create User Sequence Diagram

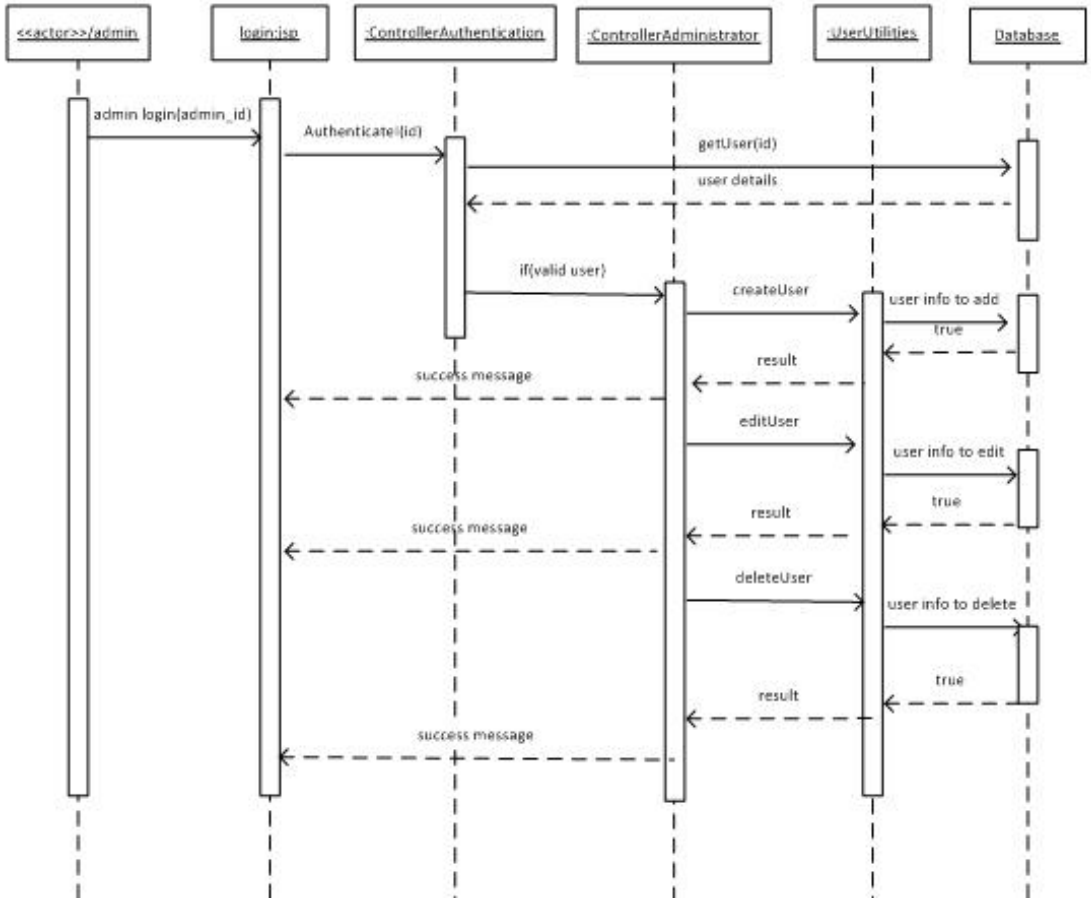


Figure 12: Create User Use Case Sequence Diagram

## 3.2 Edit Account

**Use case name:** Edit an account
**Scope:** System
**Primary Actor:** SECAM administrator
**Stakeholders and interests:** All administrators of SECAM. An administrator wants to edit an account of a customer.
**Preconditions:** This actor has to be logged on at the SECAM facilities through a SECAM website and a customer wants to change his or her personal information
**Post conditions:** The changes on the customer's account will be saved to the SECAM database

**Main Success Scenarios:**

1. Click manage accounts tab
2. Click customer accounts link
3. Click on the account link you want to modify
4. Click edit button at the bottom of form containing the customer's information
5. Make changes on the fields requested by the customer
6. Click save button
7. Click "yes" button on the pop up message

**Extensions:**

- **7a.** If an administrator leaves some of the form fields incomplete, then he or she will be notified by the system; the system will not transition to a different page and incomplete fields will be highlighted in red.
- Repeat step 6 and seven

## 3.3 Delete Account

**Use case name:** Delete an account
**Scope:** System
**Primary actor:** SECAM Administrator
**Stakeholders and interests:** All SECAM administrators. The customer wants SECAM administrator to delete their account from SECAM database
**Preconditions:** An administrator is logged on at the SECAM facilities. A customer no longer wants to use SECAM services.
**Post conditions:** The customer's account will be deleted from the SECAM database

**Main Success scenarios:**

1. Click manage accounts tab
2. Click customer accounts link
3. Click on the account name you want to delete
4. Click delete account button
5. Click "yes" button of the pop up message displayed

## 3.4 Deactivate Account

**Use case name:** Deactivate an account
**Scope:** System
**Primary actor:** SECAM administrator
**Stakeholders and interests:** All SECAM administrators. An administrator wants to deactivate an account for a customer.
**Preconditions:** An administrator is logged on at the SECAM facilities. The customer bridged the contract.
**Post conditions:** The customer's account will appear on the deactivated accounts list link. The system will send an email to the customer notifying the customer that their account has been deactivated and they should contact SECAM for more information.

**Main Success scenarios:**

1. Click manage accounts tab
2. Customer accounts link
3. Click on the account of interest
4. Click deactivate account
5. Click "yes" button of the pop up message displayed

## 3.5 Renew Account

**Use case name:** Renew an account
**Scope:** System
**Primary actor:** SECAM administrator
**Stakeholders and interests:** All SECAM administrators. An administrator wants to activate/renew a customer's account
**Preconditions:** An administrator is logged on at SECAM facilities. A customer requests for an account renewal
**Post conditions:** The system resets the account username and password. The system also sends an email consisting of the username and password.

**Main Success scenarios:**

1. Click manage accounts tab
2. Click deactivated accounts link
3. Click on the account you want to renew
4. Click activate button
5. Click yes button of the pop up message displayed

## 3.6 Viewing a Live Streaming Video

**Use case name:** Viewing a live streaming video
**Scope:** System
**Primary actor:** Campus Security Personnel
**Stakeholders and interests:** A security personnel wants to view a live streaming video of a certain room
**Preconditions:** A campus security personnel is logged on his or her SECAM account.
**Post conditions:** A streaming video is displayed

**Main success scenarios:**

1. Click a link of green flagged room
2. Click play button

### 3.6.1 View Live Stream Sequence Diagram

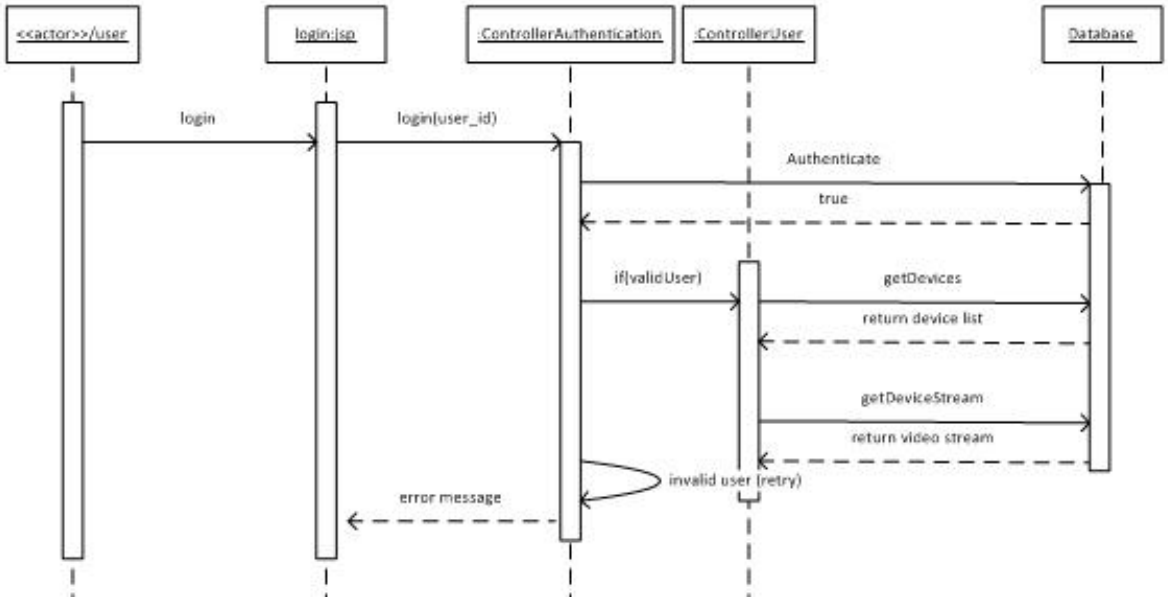


Figure 13: View Live Stream Use Case Sequence Diagram

## 3.7 Account Management

**Use case name:** Account management
**Scope:** System
**Primary actor:** Campus Security Personnel
**Stakeholders and interests:** A security personnel wants to change his or her username and password
**Preconditions:** A security personnel logs into her or his SECAM account for the first time after creating an account
**Post conditions:** The new username and password is saved to the SECAM database. A security personnel is successfully logged into the system. Message is displayed notifying the security personnel of successful change of password and username.

**Main success scenarios:**

1. Enter the URL for SECAM
2. Enter the username and password provided in the email
3. Click login button
4. Enter new username and password
5. Enter same password as in 4.
6. Click login
7. Select a security question
8. Provide the answer of the selected question

**Extensions:**

- **6a.** If the two passwords entered in 4 and 5 do not match, the system will not transition from this page and password field will be highlighted in red
- **7a.** Repeat 7 and 8 for two times selecting different questions

## 3.8 Login

**Use case name:** Login
**Scope:** User, security personnel
**Primary actor:** User, security personnel
**Stakeholders and interests:** any registered user wanting to login to the SECAM application
**Preconditions:** user must be registered i.e. should exist in the database
**Post conditions:** Successful execution results in the successful login. Unsuccessful login returns the user to the login screen.

**Main Success Scenarios:**

1. User enters their login
2. User enters their password
3. User presses the login button
4. A correct combination of user name and password results in the user being redirected to the web page displaying the control panel for the SECAM application.
5. An incorrect combination of username and password result in the user being redirected to the login page with an error message “an incorrect username and/or password have been entered”. If the user enters a wrong password more than three consecutive times, they are locked out of the system and an error message asking them to contact the administrator is shown.

### 3.8.1 Login Sequence Diagram

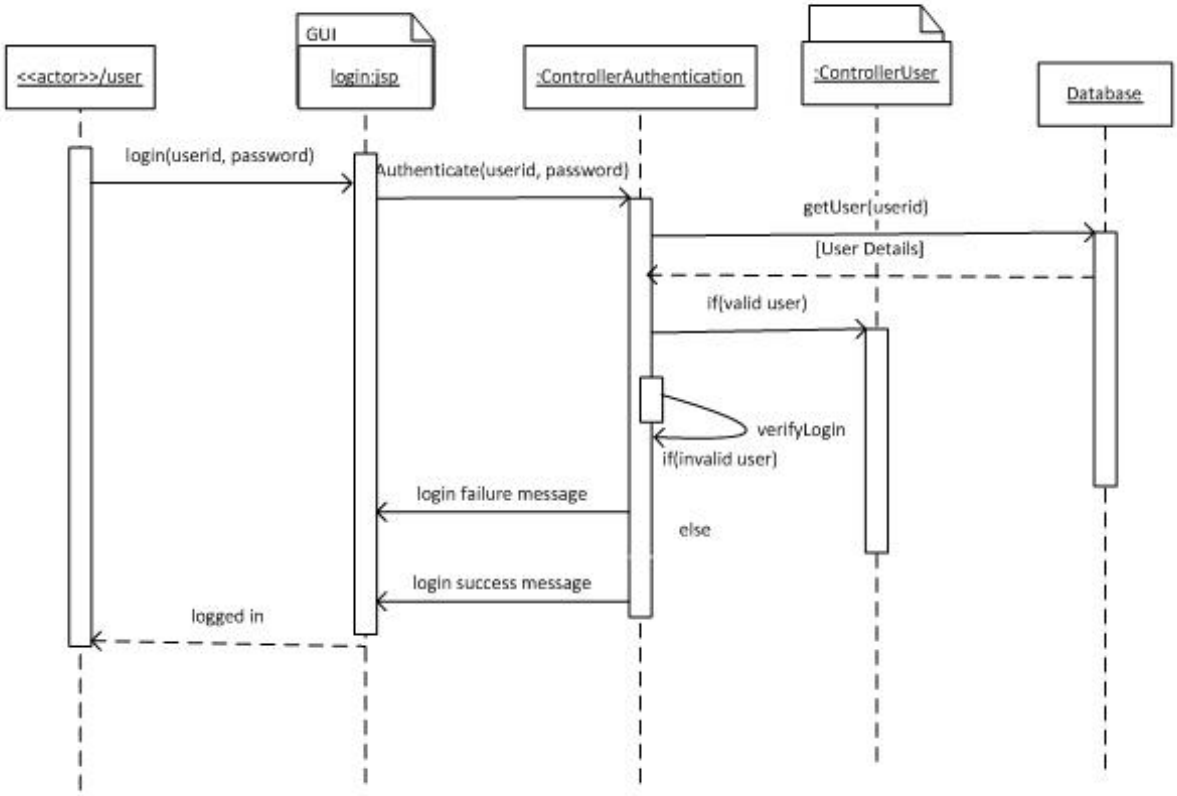


Figure 14: Login Use Case Sequence Diagram

## 3.9 Reset Password

**Use case name:** Reset Password
**Scope:** User, security personnel
**Primary actor:** User, security personnel
**Stakeholders and interests:** any registered user wanting to reset the password
**Preconditions:** User must be registered i.e. should exist in the database
**Post conditions:** Successful execution results in user resetting his password. If the user is unable to answer all three security questions, they are locked out of the system.

**Main Success Scenarios:**

1. User clicks on link “Reset Password”
2. Users correctly answers the first security question
3. Users correctly answers the second security question
4. Users correctly answers the third security question
5. User is asked to enter the new password
6. User is asked to renter the new password
7. A success message is displayed to the user
8. If the user answers any question incorrectly three times, they are brought back to the login screen and asked to contact an administrator to reset their password.

**Extensions:** None

### 3.9.1 Reset Password and Change Password Sequence Diagram

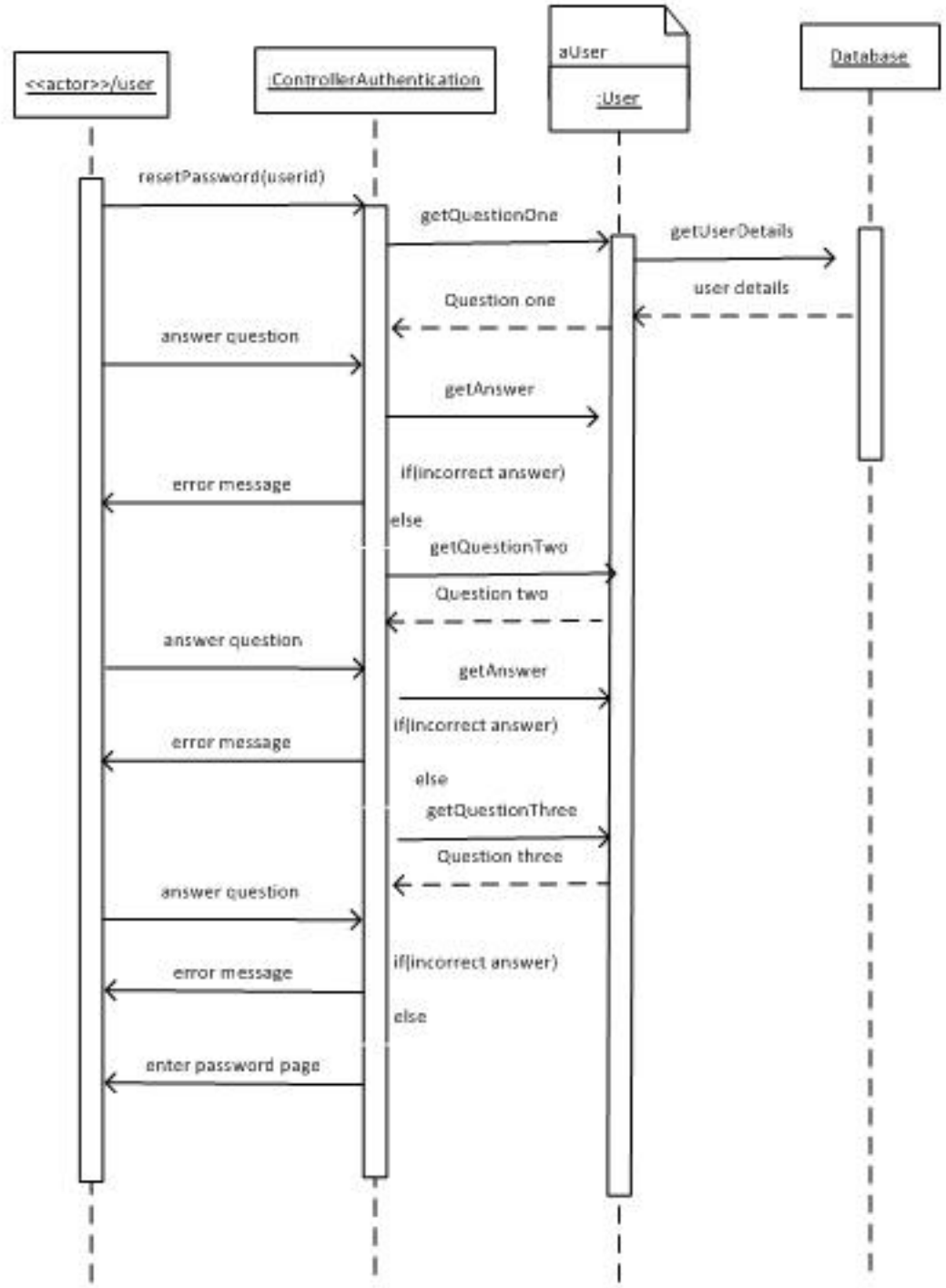


Figure 15: Reset and Change Password Use Case Sequence Diagram

## 3.10 Choose Camera Settings

**Use case name:** Choose Camera Settings
**Scope:** User
**Primary actor:** User
**Stakeholders and interests:** Any registered user wanting to customize the webcam settings
**Preconditions:** User must be registered and should have the client application installed on the local machine
**Post conditions:** Successful execution results in user successfully choosing all settings for the webcam

**Main Success Scenarios:**

1. User successfully log into the SECAM application
2. Users chooses a value from the video quality resolution from the drop down list
3. Users chooses a value for motion capture from the drop down list
4. Users chooses a value for digital zoom from the drop down list
5. User chooses a value for record time from the drop down list
6. User chooses to send video or images from the drop down list
7. User chooses a value for sound threshold
8. User chooses yes/no for notifying security personnel
9. User clicks apply button to commit the changes

**Extensions:** None

### 3.10.1 Change Camera Settings Sequence Diagram

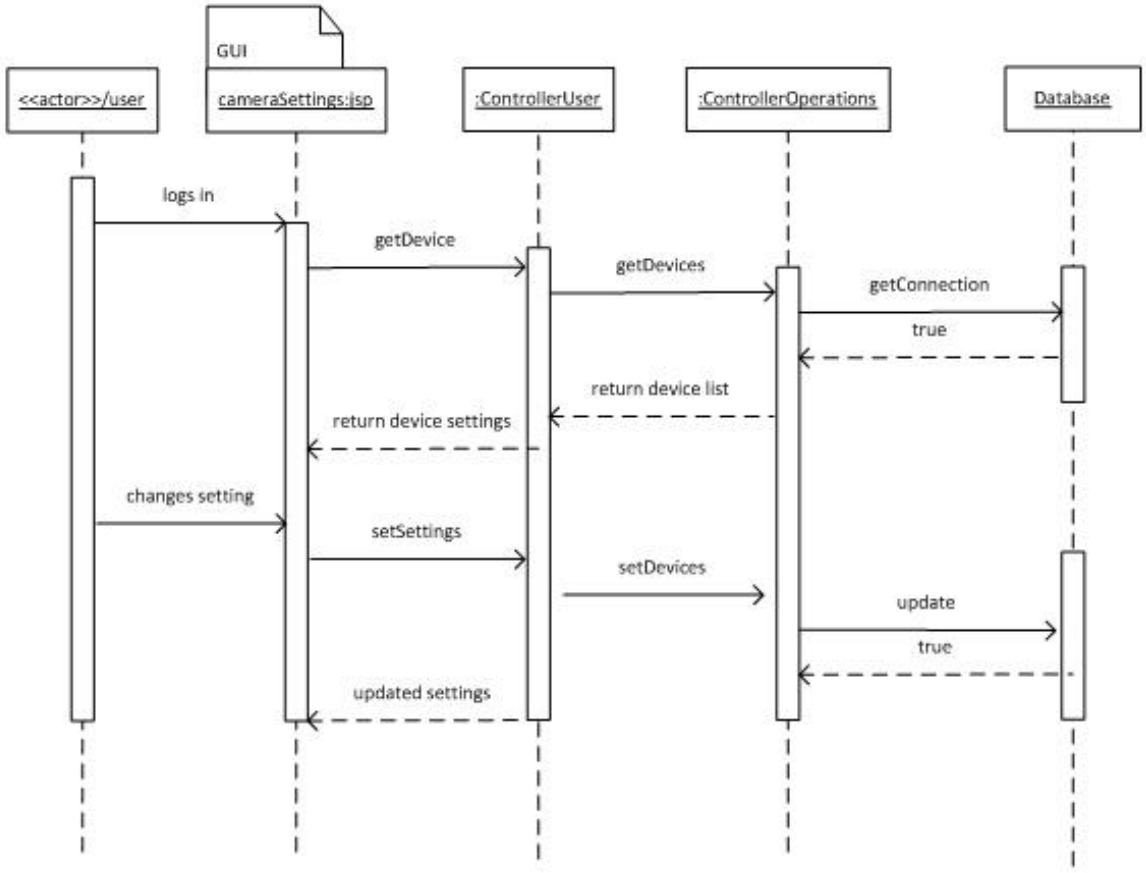


Figure 16: Change Camera Settings Use Case Sequence Diagram

# 4.0 Prototypes

## 4.1 SysControls

The SysControls user interface will display all pertinent information to the user in a structured fashion. When the user interface is not activated, SysControls runs in the background. An icon on the task bar signifies that the system is still running.

### 4.1.1 Icon Prototype

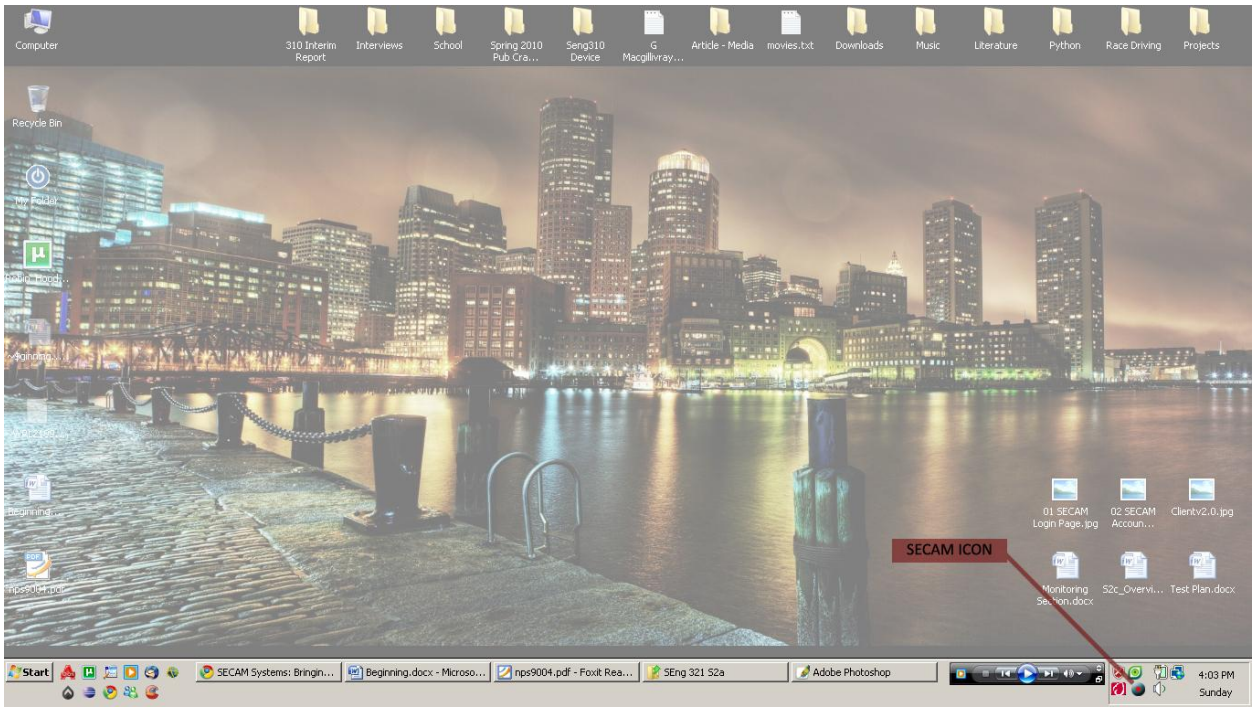


**Figure 17: A view of the icon displayed when the user interface of SysControls is not activated. The icon displays a red dot when activated and a green dot when it is not activated.**

### 4.1.2 User Interface Prototype


Figure 18: This displays the user interface for SysControls. Included are the various menu items included.

## 4.2 Web Controls

### 4.2.1 Login Page

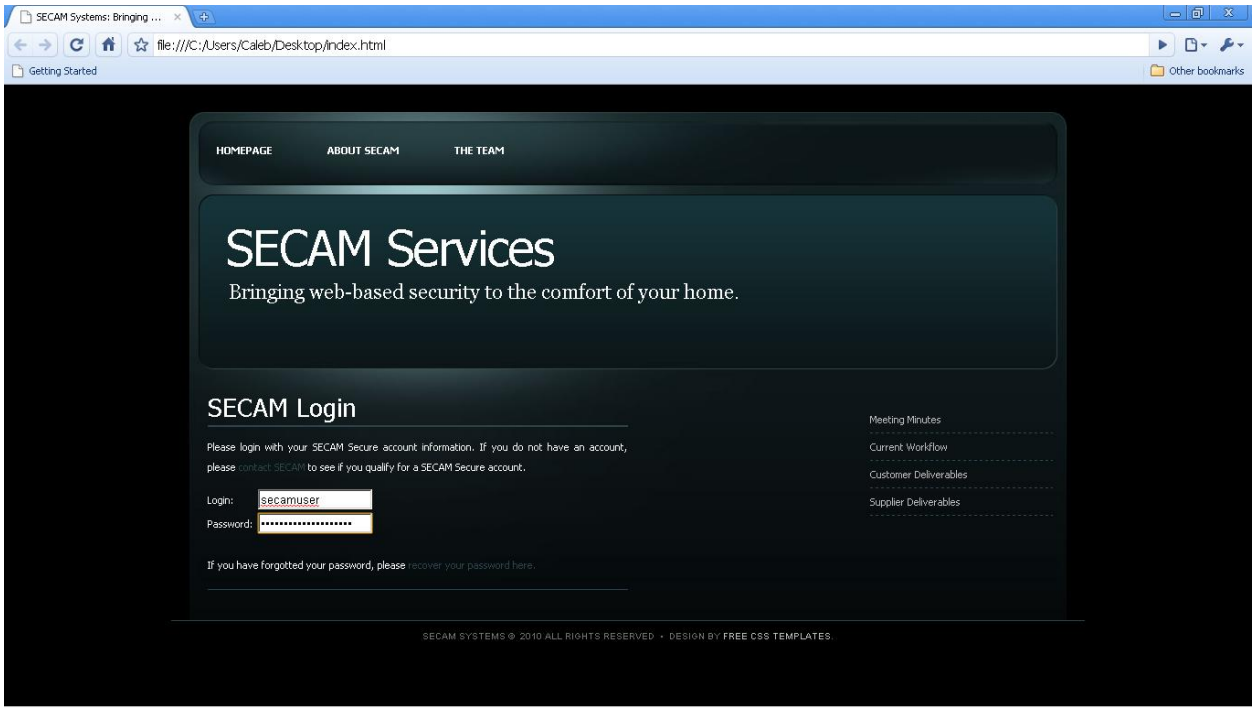


Figure 19: This displays the login page for the SECAM Web Monitoring Service.

### 4.2.2 Account Page

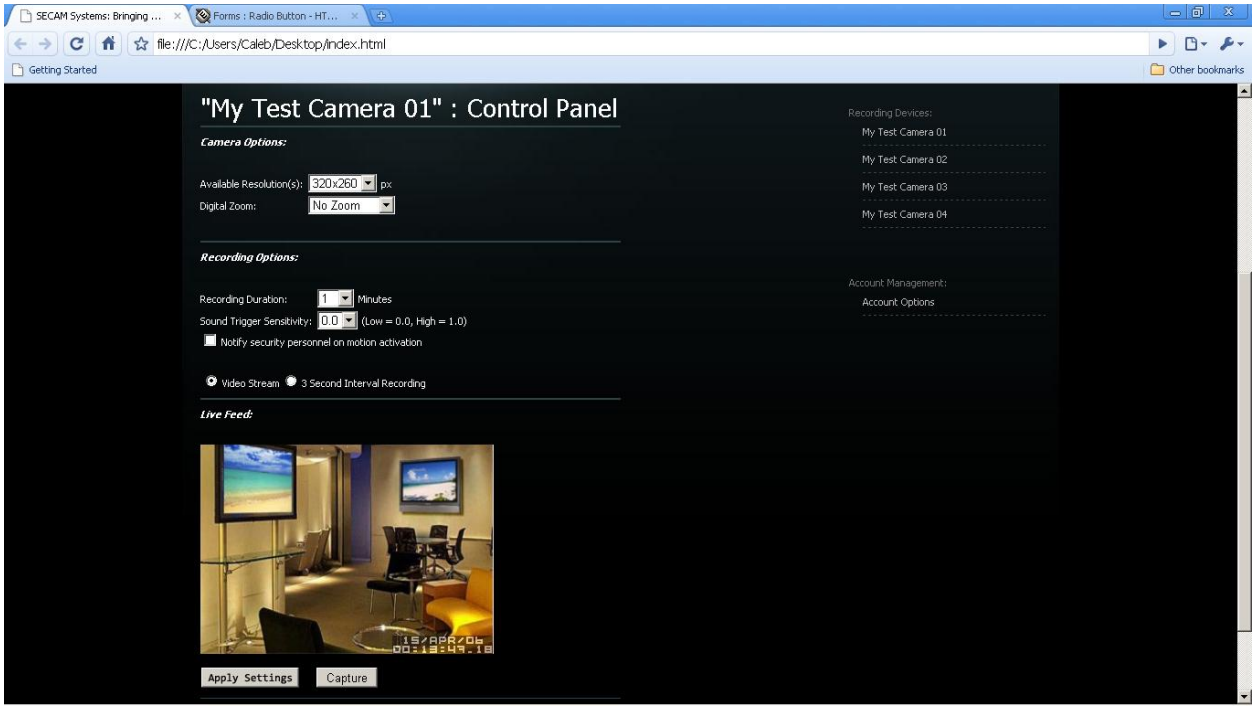


**Figure 20: This displays the options available to the user once the user logs into the SECAM Web Monitoring System. It provides accessibility to multiple cameras, and each of their options. Also, a live stream is provided of the currently selected camera.**

# 5.0 Implementation Specifics

## 5.1 Software Packages

### 5.1.1 Vision

The client application, SECAM **SysControls**, provides the motion detection for the system based on the input from the connected web camera. There are many software packages already in commercial use for vision analysis and developing one internally would be redundant and inefficient. Therefore, SECAM has elected to use the **OpenCV**. **OpenCV** was selected due to its extensive commercial and academic use. Also, **OpenCV** provides libraries for various languages such as C, C++, and Python. This is more portable and flexible when the system needs to be expanded.

### 5.1.2 Video Encryption

Along with motion detection, SECAM **SysControls** must encrypt the video stream before sending it to the secure database. This ensures the integrity of the data when it is received by the **SECURE** database. Again, there are many libraries already extensively used for this task and it would be redundant for SECAM to develop such a library internally. Therefore, SECAM has elected to use the **SSL** protocols. This implementation is available in multiple languages and has an extensive API and online community.

## 5.2 Languages

### 5.2.1 Java

Java will be used for the bulk of **SysControls**, **SECURE** database interface, and **WebControls**. However, some of the video processing will have to be developed in other languages for the sake of efficiency.

### 5.2.2 C

The video processing will be processed in an application developed in C for efficiency's sake; and because **OpenCV** provides extensive libraries in C for visual analysis.

# 6.0 Requirements

## 6.1 Functional

### 6.1.1 Functional Specifications

The SECAM system offers affordable security for offices and laboratories in the University of Victoria. Unlike traditional video monitoring systems, the SECAM system offers a more affordable and reliable security service because video recording occurs only when motion or sensors are triggered as opposed to recording twenty four hours every day. The system is reliable in that the recorded video is stored offsite. In the general model of the system, after the system has been set up and monitoring is enabled, the system is initially idle and will only start recording when a user, intruder in this case, activates the switch/sensor that activates video recording. When recording begins, the system notifies the right people that there is an intruder in the monitored place. The notification should clearly indicate the location of the intrusion.

The intended users of this system include university employees who own offices, laboratory administrators, university security personnel, teaching assistants who manage laboratories and the administrators of the system. Students are excluded from the users of the system.

The main objective of the SECAM system is to provide an affordable and reliable security and monitoring service for offices and laboratories in the University of Victoria. Students have twenty four hour access to most laboratories in the esc building especially in laboratories which only have computers. These laboratories have video surveillance and if any recording is done, it is most likely the case that it happens all the time. Recording video all the time is very expensive and a solution is needed to make a system which offers the benefits of twenty four hour recording but at a much affordable cost. The SECAM system addresses this problem. There are also other laboratories in the esc building in which students are not allowed access outside school hours since those laboratories are deemed to have expensive equipment. With a system like SECAM installed in those laboratories students could be granted access to the laboratories.

The most important features of the system include triggering video recording through some kind of a sensor, storing video at an offsite location, notifying intended personnel when video recording is triggered and allowing the user to view live video steam of their videos when recording has been triggered and allowing access to the system through a web based interface.

The hardware that the system uses and interacts with includes a computer, a webcam, and some servers to store the recorded videos and possibly a sensor that turn recording on.

### 6.1.2 Functional Requirements

#### *6.1.2.1 Trigger Recording on Motion*

**Requirement Name:** Motion-Triggered Recording
**Functional Requirement ID:** 001

**Summary:** When the system is activated, any motion in view of the camera should be recognized and trigger the recording of video.

**Rationale:** This is an essential requirement of the system. If the system cannot detect motion and record video when needed, it is useless.

#### *6.1.2.2 Trigger Recording on Sensor Signal*

**Requirement Name:** Sensor-Triggered Recording
**Functional Requirement ID:** 002

**Summary:** When a sensor sends a signal to SysControls, the client program, the recording of video should be triggered.

**Rationale:** This allows for expansion of the system to meet the needs of the environment.

#### *6.1.2.3 Notify Specified Personnel on Recording*

**Requirement Name:** Notification of Recording
**Functional Requirement ID:** 003

**Summary:** When recording starts due to a sensor trigger or motion, the system should notify the user every time, and notify any security personnel that the user has specified in the camera settings interface.

**Rationale:** This provides a form of action based on the detection of intruders. Not only will it record them, but it will also notify others that there is someone in the room. If the room is of specific importance and exclusivity, the user can even specify the system to notify security personnel directly so that they can react accordingly.

#### *6.1.2.4 Stream Live Video on Web Interface*

**Requirement Name:** Stream Live Video
**Functional Requirement ID:** 004

**Summary:** The user is able to log into their account via the web interface and view a live stream of their camera.

**Rationale:** This gives the user, or security, the needed context in which to approach the intruder. Also note that once the video is triggered, it is saved to the **SECURE** database.

#### *6.1.2.5 Change Camera Settings on Web Interface*

**Requirement Name:** Change Camera Settings
**Functional Requirement ID:** 005

**Summary:** The user is able to change various camera settings such as resolution and zoom (but NOT move the camera as most cameras do not support this capability).

**Rationale:** This allows for limited control of the camera while still in a remote location. This can be very useful when the user is viewing the live stream.

## 6.2 Performance Requirements

### 6.2.1 Video Quality

One of the systems' necessary performance requirements is that the minimum resolution of the captured video be at a level that is acceptable. A video quality of 300 by 300 pixels at frame rate of 30 frames per second is the systems' minimum allowed video quality. Having such a performance requirement as this one is necessary to insure that the captured video is useful and comprehensible.

### 6.2.2 Computer Specifications

The computer needed to run SECAM application is as follows should have a processor speed of more than 2GHz single core or 1.2GHz if it has multiple cores, it should have a memory of 2GBytes and be running windows. These requirements will be important because the system is required to run smoothly with minimum delay on all operations.

### 6.2.3 Data Security

It is required that the system transmit recording data to the company servers as soon as recording starts with little or no delay. A delay in this part of system is intolerable because the idea behind the system is to both provide a way to prevent an intruder from having access to video surveillance recordings and to provide the a location or way to prevent anyone from tampering with the recorded video so that it can be used for legal purposes.

## 6.3 Safety and Emergency Requirements

### 6.3.1 Handle the Loss of Power to SECURE Database

The SECURE database will be stored at the SECAM facilities, and, as such, SECAM has control over the hardware use. The main SECURE database will have backup power supplied to it in the event of a power outage. This will be in the form of a **UPS** (Uninterruptible Power Supply) that will supply power immediately to the database on power failure. However, the **UPS** is a temporary measure until the back-up generators are activated. This allows for no data loss whatsoever due to the multi-tiered power back-ups.

### 6.3.2 Handle the Migration and Maintenance of SECURE Database

When a portion of the SECURE database needs to migrate to a new location, or if maintenance is required, the section is powered down. The client applications, SysControls, will not lose connection to the SECURE database because they are communicating with the shell of the database and never talking directly with the database. This means that they will never know that maintenance or migration is happening. This is achieved through redundancy of essential components and by never taking the entire system offline at once. Only sections of the SECURE database will be down at once; never the entire system.

### 6.3.3 Handle the Loss of Internet Connection by SysControls

Due to the nature of the Web Monitoring System, it is impossible for SECAM to control the hardware aspect of the client's environment. Because of this, robust software design is essential. The loss of connection to the internet, and subsequently the SECURE database, will result in the client application recording any video to the user's hard drive and will send the data to the database when connection is re-established. The video is saved in its encrypted state to secure the integrity of the saved data.

### 6.3.4 Handle the Loss of Power of SysControls

Due to the nature of the Web Monitoring System, it is impossible for SECAM to control the hardware aspect of the client's environment. Therefore, SECAM cannot control the type of contingency methods used on the client's end of the software. All SECAM can do is strongly suggest that the clients use a **UPS** to provide power in the event of a power loss, and ensure that they have a reliable connection to the internet.

# 7.0 Constraints

## 7.1 Component Communication

Each component will communicate via **SSL**, or TLS v1.2, to secure their communications. Once their correspondence is secured, the components will transmit data between each other using **SSL** and **HTTPS**.

## 7.2 Sensor Communication

Sensors can be added to the system to expand the functionality of the detection methods. Sensors will have to transmit a binary signal to SysControls; transmit a 1 if detection has occurred and 0 if there has been no detection. This simplifies the inputs to the system while still providing a method to expand its functionality.

## 7.3 Camera Movement

Due to the various types of web cameras, and SECAM's desire to make their software available to the widest possible audience, the control of camera movement is not supported in the SECAM Web Monitoring System. This greatly simplifies the client software and user interfaces for the cameras. However, SECAM has not completely rejected the notion and the modular nature of the SysControls client software may allow for expansion in this direction if need be.

# 8.0 Testing

## 8.1 Objectives

The testing objectives for the SECAM Web Monitoring System are designed to fully realize the specifications of the system. This reflects the requirements specified in the previous documents and above. Special care will be taken in the aspects of validation, security, rigidity, and reliability. This will contribute to the creation of a high quality product that is usable and secure for users. The testing procedure will be broken down into various phases and executed at various times within the development process. This allows for a multi-level and multi-view approach to the testing of the system which will provide a superior perspective for detecting bugs and logic errors.

## 8.2 Monitoring and Correction Procedures

### 8.2.1 The Plan

SECAM Management has decided to implement an agile development methodology.

Each team member will have long and short-term goals. The long-term goals will include larger-scale production such as an entire module set. Short-term goals will include production that can be completed in a few days such as a single module. Each goal will have a deadline. This provides the team member with a general perspective to which they can budget their time efficiently. Team members will report their progress to the rest of the team daily in SCRUMs.

In the SCRUM, team members will describe what they did the day before and what they intend to do today. This provides management with the ability to quickly identify problem areas and assign other members to assist in developing a solution. Also, by having the team members describe what they have done; management can keep a high-level eye on who is working efficiently and who is not. This is very useful in monitoring the amount of work being done by each team member. Also, other team members may want clarification on certain aspects of modules and this provides members with the ability to do so.

By requiring each team member to describe what they have done, the team member gives the rest of the team a perspective into their progress. This allows the distribution of knowledge and provides an open context for team members to ask questions about the work of other team members. By having sufficient knowledge of the work being completed, or not completed, members are provided a means to motivate each other and keep on track as they may be waiting for the completion of others' modules. This changes the management structure of the team from having the manager keep members on track, which may still happen on occasion, to having the team members keep each other on track. For example, if in the SCRUM 'Bob' keeps saying that he is still working on his module, 'Sally', who is developing a module that requires Bob's module, will keep asking Bob when he is done. This makes the timeline of the product a team responsibility.

### 8.2.2 Code Review

A first-level filter to inefficient and error-prone code is a quick code review by another team member. This will be done without the assistance of the original developer as the review will also test the level of documentation and comments for the code. Code review is a good way to filter out top-level logic errors, bugs, and documentation problems. Code review will be assigned to random team members whenever a module has been completed. This means that code review will be prevalent throughout the development process and will work in a dynamic nature. The emphasis here will be on a quick overview of the module by another team member to see if they identify any errors that the developer may have overlooked. This can be costly if too much time is spent on it so members will be encouraged to only do quick reviews.

### 8.2.3 Bugs and Bug Reporting

Bugs will be tracked through a rigid QA process. QA will branch every two weeks. They will run each branch through the defined test cases and will create JIRA tickets for each failure of requirements. Theses tickets will describe the module and class of the failure and any pertinent comments on the matter by the QA tester. The JIRA tickets will be sent daily to the team leader who will review them and assign them to the required team member. Team members will be notified via e-mail of their JIRA ticket assigning and they can log into JIRA to view and comment on their ticket list. The comments are sent to the QA tester of the ticket which provides a direct avenue for the developer to ask further questions and talk to the creator of the ticket personally. Once a ticket is completed, the team member responsible for the ticket will close the ticket in JIRA. This sends the ticket back to QA for review. QA tests the tickets when there is a new branch and provides feedback to the team leader on resolved tickets. This feedback will either be more tickets or comments describing the completion of the ticket. This gives the team leader a perspective of the amount of tickets being completed and how many are not fixing the underlying problems.

## 8.3 Testing Overview

Each module must conform to a list of basic tests. This list is generic enough to apply to all modules, but rigid enough to provide a high-level start to the testing process in an accurate manner. The testing will be divided into the following sections:

1. Unit Testing
2. Integration Testing
3. Functional Testing
4. Performance Evaluation

### 8.3.1 Unit Testing

Unit testing is the responsibility of the developer. Each module must be tested against a basic list of tests. This verifies that each module satisfies the minimum requirements for it to be accepted into the branch. Considering that each module may be significantly different, this list is only a starting place and each developer is required to expand the test list to fully verify the integrity of each module. Before development of a module, each developer is required to provide a list of basic test cases that are specific to their module. These test cases should encompass the main aspects of the module and test them thoroughly. Upon completion of development, the developer is required to create a separate list of test cases that will further verify the integrity of the module. This ensures that the module is thoroughly tested and all results are documented for the code reviewers.

**Basic Unit Testing List**

- Input validation and handling implemented
- Output format and standards correct
- Coding and documentation standards used
- Error handling and data handling implemented
- Does it satisfy all of the given requirements for the module

### 8.3.2 Integration Testing

Integration testing requires that the associated modules be completed and unit tested. Integration testing will verify the integration of the various modules. This requires a review of the unit testing documentation, the long-term requirements, and the standards that apply to the modules. With these documents in hand, the testers can begin to test the integration of the modules. This is done by starting with a pre-defined list of generic integration test cases and then defining further test cases based on the specific module functionality.

**Basic Integration Testing List**

- Does the output of module A correctly match the inputs of module B
- Do both modules conform to integration standards
- Do both modules have the required documentation
- Does module A call the correct functions in module B for the required functionality
- Is the output of the combined system within the requirements specified
- Is the input of the system a correct mapping to the output of the integrated system
- Does the integration create any new bugs or unintended behaviour

### 8.3.3 Functional Testing

Functional testing will be broken down into two sections; the preliminary functional test design early in the development process and the execution of the designed functional tests when the bulk of the system has been completed. This tests the entire system and its specifications.

The design of the functional testing will focus on the actual functionality of the system. That is, they will test the inputs and outputs of the system to verify that they conform to the system specifications. Designing the functional tests early in the development process allows for early detection of specification problems that might be not be discovered until the actual execution of the functional testing.

The second section, the execution of the functional tests, requires the completion of the designed functional tests. Each test is executed and the result is either a pass or fail. If all functional tests pass, the system is said to pass the functional testing.

### 8.3.4 Performance Evaluation

Performance evaluation will be subject to the module being tested. As with the other testing, the evaluation will start with the testing of the basic performance requirements for all modules. Further performance evaluation will be required for the various modules and further testing is required to be specified before testing begins. The designers of the tests must consider various aspects of the system such as security, bandwidth speed, storage requirements, and fail-safes and back-ups. Some of the previously mentioned aspects may not pertain to certain modules and, thus, can be omitted from further testing.

**Basic Performance Evaluation List**

- Does the system provide the required security
- Does the system communicate to other modules in a secure manner
- Does the system handle user error
- Does the system handle disconnection of other modules eloquently
- Does the system satisfy the required speed specifications
- Does the system exceed storage specifications
- Ranges: max, min values
- Does the system include adequate fail-safes
- What happens if there is a power outage
- What happens if the user leaves their account logged in while on the web interface
- What happens if the system is disconnected from internet

## 8.4 Test Cases

### 8.4.1 User Login

**Use Case reference:** Login
**Test case Name:** User Login
**Test case Date:**
**Test case Phase:**
**Test case Purpose:** To ensure that only valid users are able to log into the SECAM application
**Test case Function:** To verify that only a correct/valid combination of username and password allow a user to log into the SECAM application.

**Test case Input:** An alphanumeric username and an alphanumeric password
**Test case expected Output:** For a registered user entering the correct combination of username and password, the control panel of the SECAM application is shown. For an incorrect combination, the user remains on the login page and an appropriate error message is shown.

**Test case Procedure:**

| Test case input | Test case output |
|---|---|
| An incorrect username and an incorrect password | The user is remains on the login page and the following error message is shown: "The username does not exist" |
| A correct username but the wrong password for the username is entered | The username remains on the page and the following error message is shown: "The password entered is incorrect" |
| A blank username and password is entered | The user remains on the login page and the following error message is shown: "Please enter a valid username" |
| A correct username but the wrong password is entered more than 3 times | The user remains on the login page and the following error message is shown: "You have been locked out of the system. Please contact the administrator to get access to the system" |

**Test case Documented Result:**

### 8.4.2 Reset Password

**Use Case reference:** Reset Password
**Test case Name:** Reset Password
**Test case Date:**
**Test case Phase:**
**Test case Purpose:** To ensure that the user is able to reset their password successfully
**Test case Function:** To verify that a user is able to reset their password if, and only if, they are able to answer their three security questions.

**Test case Input:** User's answers to the three security questions
**Test case expected Output:** Webpage to enter new password or an error message asking the user to contact the administrator to reset their password.

**Test case Procedure:**

| Test Case Input | Test Case Output |
|---|---|
| Click the "reset password" link/button | Webpage displaying the first security question along with space to enter the answer is shown |
| Enter the answer to the first security question correctly | The second security question is shown second |
| Enter the answer to the first security question incorrectly | Webpage with the first security question is shown along with the following error message "This is not the correct answer. Please try again" |
| Enter the answer for the second security question correctly | The third security question is shown to the user |
| Enter the answer for the second security question incorrectly | Webpage with the second security question is shown along with the following error message "This is not the correct answer. Please try again" |
| Enter the answer for the third security question correctly | Screen to enter the new password is shown |
| Enter the answer for the third security question incorrectly | Webpage with the third security question is shown along with the following error message "This is not the correct answer. Please try again" |
| Answer the first or the second or the third security question incorrectly for three consecutive | A webpage with the following error message is shown: "You have been locked out of the system. Please contact an administrator to reset your password" |

**Test case Documented Result:**

### 8.4.3 Validate Password

**Use Case reference:** Login
**Test case Name:** Validate password
**Test case Date:**
**Test case Phase:**
**Test case Purpose:** To ensure user enters valid password
**Test case Function:** Verify that no password entered by the user meets all the required constraints.

**Test case Input:** A valid password entered two times
**Test case expected Output:** A success message informing the user that their password has been successfully changed.

**Test case Procedure:**

| Test Case Input | Test Case Output |
|---|---|
| Enter a password less than 8 characters long | The following error message is shown: "The password must be exactly between 8 and 18 characters long" |
| Enter a password longer than 18 characters | The following error message is shown: "The password must be exactly between 8 and 18 characters long" |
| Enter a password between 8 and 18 characters long but with no numerals. | The following error message is shown: "The password must have at least one numeral" |
| Enter a password between 8 and 18 characters long but with no special characters. | The following error message is shown: "The password must have at least one special character" |
| Enter a password that meets all constraints | A screen prompting the user to renter the new password |
| Enter the wrong password in the screen where the old password is to be entered again | The following error message is shown: "The two passwords do not match" |
| Renter the password correctly. | The following message is shown: "Your password has been successfully changed. Please use the new password to log into the system" |

**Test case Documented Result:**

### 8.4.4 Validate Camera Settings

**Use Case reference:** Choose Camera Settings
**Test case Name:** Validate Camera Settings
**Test case Date:**
**Test case Phase:**
**Test case Purpose:** To check if a registered user can customize the camera settings
**Test case Function:** To verify that the settings chosen by the user are successfully saved in the database.

**Test case Input:** The camera settings entered manually by the user
**Test case expected Output:** Changes are reflected in the camera settings

**Test case Procedure:**

| Test Case Input | Test Case output |
|---|---|
| Choose a value of Video Quality resolution from the drop down list. Click Apply button. Logout of the application. Login again into the application. | After re-logging into the application, the value of the video quality resolution should be the one chosen. |
| Choose a value of Motion Capture from the drop down (yes/no). Click apply button. Logout of the application. Login again into the application. | After re-logging into the application, the value of the Motion Capture should be the one chosen. |
| Choose a value for Digital Zoom from the drop down. Click apply button. Logout of the application. Login again into the application. | After re-logging into the application, the value of the digital zoom should be the one chosen. |
| Choose a value for Record Time from the drop down. Click apply button. Logout of the application. Login again into the application. | After re-logging into the application, the value of the record time should be the one chosen. |
| Choose a value of Video or Images from the drop down (Video or images). Click apply button. Logout of the application. Login again into the application. | After re-logging into the application, the value of the video or images should be the one chosen. |
| Choose a value for Sound Threshold from the drop down. Click apply button. Logout of the application. Login again into the application. | After re-logging into the application, the value of the sound threshold should be the one chosen. |
| Choose yes, from the drop down for Notify Security Personnel. | The screen to enter the contact details for the security personnel should be shown next. |

**Test case Documented Result:**

### 8.4.5 Create Account

**Use case reference:** Create Account
**Test case name:** Create Account
**Test case date:**
**Test case phase:** black box testing
**Test case purposes:**

- To make sure the customer's information can be retrieved from the database without errors.
- To make sure the system generates and sends an email containing the correct username and password.
- To make sure the system displays the created account under customer account

**Test case function:**

**Test case input:** Customer's first name, surname, birth date, company name, customer's company email address, company address
**Test case expected output:**

- The created account must be displayed under customer accounts
- The correct customer's personal information must be displayed
- The email generated by the system must be displayed on the customer's email when he/she logs into his or her email account

**Test case procedure:** Assuming an administrator is logged on at SECAM facilities

| Test case input/action | Test case output/system response |
|---|---|
| 1. Click manage accounts tab | List of active customer accounts is displayed |
| 2. Click create account button | A empty incomplete form is displayed |
| 3. Enter customer's first name, surname, birth date, company name, customer's company email address, company address<br>4. Click save button | -A pop up message saying "do you want to save changes to this account?" is displayed<br>-the system generates and sends a default password and username to the customer's company email address |
| 5. Click yes button | A form showing the greyed personal information is displayed |
| 6. Click customer accounts link | List of active accounts is displayed |
| 7. Click on the newly created account | A form showing the saved personal information is displayed |
| 8. open new browser and login to your email account | A message containing the generated username and password should appear in inbox list |

**Test case documented results:**

### 8.4.6 Edit Account

**Use case reference:** Edit Account
**Test case name:** Edit Account
**Test case date:**
**Test case phase:** Black box testing
**Test case purpose:** To make sure changes made in an account will be saved to the database
**Test case function:**
**Test case input:** Dummy personal information
**Test case expected output:** Any changes made in the account must be reflected in the displayed account
**Test case procedure:** Assuming an administrator is logged on and there is an account to modify

| Test case input/action | Test case output/system response |
|---|---|
| 1. Click manage accounts tab | List of active customer accounts is displayed |
| 2. Click on a name of account to modify | A form showing the customer's greyed personal information is displayed |
| 3. Click edit button | A form showing the customer's *ungreyed* personal information is displayed |
| 4. Make changes in the targeted fields<br>5. Click save button | A pop up message saying "do you want to save changes to this account?" is displayed |
| 6. Click yes button | -A form showing the greyed personal information is displayed<br>- A message saying "congratulations! Your changes have been saved" at the top of this form is displayed. |
| 7. Click customer accounts link | List of active accounts is displayed |
| 8. Click on the name of the account you just created | Changes in the form displayed should be visible |

**Test case documented results:**

### 8.4.7 Delete Account

**Use case reference:** Delete Account
**Test case name:** Account Deletion
**Test case date:**
**Test case phase:** Black box testing

**Test case purpose:**

- To make sure the deleted account does not appear on the customer accounts list
- To make sure the login fails on a deleted account

**Test case function:**

**Test case input:** none
**Test case expected output:**

- The deleted account must not appear under customer accounts
- The login for this account must fail

**Test case procedure:** Assuming an administrator is logged on at SECAM facilities and there is a dummy account to delete

| Test case input/action | Test case output/system response |
|---|---|
| 1. Click manage accounts tab | List of active customer accounts is displayed |
| 2. Click on the name of an account you wish to delete | A form showing the customer's greyed personal information is displayed |
| 3. Click delete button | A pop up message saying "do you want to delete this account?" is displayed |
| 4. Click yes button | -List of active accounts is displayed<br>-The deleted account should not appear on this list |
| 5. Log out and login using the deleted account's credentials | A message saying "invalid credentials" is displayed |

**Test case documented results:**

### 8.4.8 Deactivate Account

**Use case reference:** Deactivate an account
**Test case name:** Freeze account
**Test case date:**
**Test case phase:** Black box testing
**Test case purposes:**

- To make sure the deactivate account does not appear customer accounts list
- To verity that the deactivated account is displayed in the deactivated accounts list
- To make sure that the system generates and sends an email notifying the customer of his or her deactivated account
- To make the system rejects the credentials of a deactivated account

**Test case function:**

**Test case input:** none
**Test case expected output:**

- The deactivated account must be displayed on the deactivated account list
- A deactivated must not appear on the customer accounts list
- The generated email must appear on the customer's email account
- The system must reject username and password for this deactivated account

**Test case procedure:** Assuming an administrator is logged on and there is a dummy account to deactivate

| Test case input/action | Test case output/system response |
|---|---|
| 1. Click manage accounts tab | List of active customer accounts is displayed |
| 2. Click on the name of the account you want to deactivate | A form showing the customer's greyed personal information is displayed |
| 3. Click deactivate account | -A pop up message saying "do you want to delete this account?" is displayed<br>- A message saying "Your account has been deactivated. Please contact SECAM for more information" should be sent to customer's company email address |
| 4. Click yes button | -List of active customer accounts is displayed<br>-The deactivated account should not appear on this list |
| 5. Click deactivated accounts link | -List of deactivated customer accounts is displayed<br>-The deactivated account should appear on this list |
| 6. Log out and login using the deactivated account's credentials | A message saying "invalid credentials" is displayed |
| 7. Open new browser window and login into your email account | A message from SECAM saying "Your account has been deactivated. Please contact SECAM for more information" should appear in your inbox |

**Test case documented results:**

### 8.4.9 Renew Account

**Use case reference:** Renew Account
**Test case name:** Activate Account
**Test case date:**
**Test case phase:** Black box testing
**Test case purpose:**

- To make sure the customer's personal information and account permissions has not changed since the account freeze
- To ensure that the customer has access to their account again
- To make sure the renewed account is displayed on the list of active accounts and does not appear on the list of deactivated accounts

**Test case function:**
**Test case input:** none
**Test case expected outputs:**

- The renewed account must be visible on the list of active accounts
- It must also disappear in the list of deactivated accounts
- The activated account must contain correct customer's personal information
- The system must grant access to the account if the correct credentials for this account are provided

**Test case procedure:** Assuming an administrator is logged on and there is dummy deactivated account

| Test case input/action | Test case output/system response |
|---|---|
| 1. Click manage accounts tab | List of active customer accounts is displayed |
| 2. Click deactivated accounts link | -List of deactivated customer accounts is displayed |
| 3. Click on the name of the account you want to renew/activate | A form showing the customer's greyed personal information is displayed |
| 4. Click renew button | -A form showing the customer's greyed personal information is displayed<br>-A message saying "Your account is now active" is displayed at the top of the form |
| 5. Click deactivated accounts link | -List of deactivated customer accounts is displayed<br>-The renewed account should not appear on this list |
| 6. Click customer accounts | -List of active customer accounts is displayed<br>-the newly renewed/activated account name should appear on this list |
| 7. Login using the correct renewed account's username and password | Your control panel page is displayed |

**Test case documented results:**

### 8.4.10 View Live Video Stream

**Use case reference:** viewing a live streaming video
**Test case name:** multiple live videos
**Test case date:**
**Test case phase:** black box testing
**Test case purpose:** to make sure security personnel can switch between two streaming videos without system errors
**Test case function:**

**Test case input:** none
**Test case expected output:** the system must display one video after the other or in multiple browser tabs without delay or errors

**Test case procedure:** Assuming security personnel is logged on and there are videos streaming

| Test case input/action | Test case output/system response |
|---|---|
| 1. Click a link of a green flagged room | A video display is displayed |
| 2. Click play button | The video plays without delay |
| 3. Click a different link of a green flagged room | A video display is displayed |
| 4. Click play button | A different video from 2. plays without delay |

**Test case documented results:**

## 8.5 Testing Schedule

The testing methods used to evaluate the integrity and validity of the various components are broken down into five stages:

1. Stage 1 Testing: Unit Testing
2. Stage 2 Testing: Module Integration Testing
3. Stage 3 Testing: Functional Testing
4. Stage 4 Testing: Component Testing
5. Stage 5 Testing: Performance Evaluation

### 8.5.1 Unit Testing (Stage 1 Testing)

Unit testing will be executed whenever a module has been completed. This provides the first level of screening for bugs and logic errors. Also, this is the only method that provides immediate feedback to the developer. If a module fails unit testing, the developer is notified immediately and must fix the errors before continuing. No documentation is produced for this test as the developer executes this test himself.

### 8.5.2 Module Integration Testing (Stage 2 Testing)

Integration testing will occur when a module has been completed and has passed its unit tests. Integration testing will test the module and its compatibility with the other existing modules. This will also test the module against the specifications provided in the various requirements documents. When multiple modules are completed, each module must repeat integration testing when a new module has been added. This ensures that no added module has corrupted the integrity of the previous build. If the module fails integration testing it is sent back to the development team that created it with the testing documentation. This provides them with the feedback required to correct the error and continue their progress. The documentation produced during this test is a list of all functions and modules tested and which ones passed and failed and why. This can be a very detailed document as each error will be specified in detail.

### 8.2.3 Functional Testing (Stage 3 Testing)

Functional testing will occur when the majority of the modules for a certain component have been completed and integration tested. Functional testing will test the semi-completed, or completed, component against the requirements specified in this document and previous documents. Also, this test will provide an overview of the performance requirements so that the development team will have feedback on their progress. This ensures that the components act as they should and that they do not produce any unintended results. If a component fails functional testing, the component is sent back to the development team with the testing documentation for correction and completion. The documentation produced at this stage of testing is a list of modules tested, the inputs used, the output produced, and any errors encountered during testing. Functional testing is repeated after each completion of modules.

### 8.2.4 Component Integration Testing (Stage 4 Testing)

Component integration testing is very similar to module integration testing. Once a component is completed, it must be tested against the communication standards provided and their compatibility with other components. Each component must communicate with other components in their specified ways. The goal of this stage of testing is to verify that each component satisfies its requirements in conjunction with communication specifications. If a component fails the component integration testing, it is sent to the development team with the created test documentation. The documentation provided by this test is a list of communication standards, and tests of these standards, and the results for each test. Also, the specifications of each error in the test and any pertaining documentation provided during the test such as logic error comments.

### 8.2.5 Performance Evaluation (Stage 5 Testing)

Once a component has satisfied the functional testing requirements and component integration testing, it is sent for performance evaluation. This is a high-level evaluation and can only be executed on a semi-completed system. This test examines the system and compares it to the performance requirements specified in this document. If the system fails a performance evaluation, the testing documentation is sent to the development team and the group that developed the problem area is assigned the task of correction. At this point in the development process it would be extremely costly to recreate an entire section of a component. Because of this, it is extremely important that the testing team provide adequate feedback of the development team's progress of the performance requirements during the functional testing stage. The documentation provided during this test is a list of the performance requirements, the list of tests that evaluate each performance requirement, and the result of each test; also, further comments and detailed error specifications may be added by the testing team if they have them.

## 9.0 Conclusion

Therefore, the implementation of the SECAM Web Monitoring System will focus on security, rigidity, scalability, maintainability, and reliability. The various aspects of this document, and their detailed nature, will provide developers with an excellent perspective and guide in implementing this system.

# Glossary

**AES:** AES, or Advanced Encryption Standard, is the encryption standard adopted by the U.S. government. It is fast and secure. (http://en.wikipedia.org/wiki/Advanced_Encryption_Standard)

**Black Box:** A black box system is a system that can only be viewed in terms of input and output. This means that you cannot see the internal workings of the system, just what it produces given a certain input. This is useful when security is a requirement.

**ECS:** The Engineering and Computer Science building at the University of Victoria, BC, Canada.

**HTTPS:** HTTPS is a combination of the Hypertext Transfer Protocol (HTTP) and SSL/TLS to provide encryption and secure identification of servers. (http://en.wikipedia.org/wiki/Https)

**OpenCV:** OpenCV is a library of methods for real-time computer vision. (http://opencv.willowgarage.com/wiki/)

**SECURE:** SECAM Secure is the secure database that stores the recorded video logs of the system.

**SSL:** SSL, or Secure Socket Layer, which is now TLS (Transport Layer Security), is a cryptographic protocol used to encrypt data for communication over such mediums as the internet. (http://en.wikipedia.org/wiki/Transport_Layer_Security)

**SysControls:** SysControls is the client application of the SECAM Web Monitoring System. It is installed on the user's computer, runs in the background, and sends the video to the secure database.

**SyServe:** See **SECURE**.

**UPS:** A UPS, or Uninterruptible Power Supply, is a back-up power supply that protects devices against surges, and provides 5 to 15 minutes of back-up power; enough time for auxiliary power, such as a generator, to be activated. You can purchase UPS devices at many computer stores. (http://en.wikipedia.org/wiki/Uninterruptible_power_supply)

**WebControls:** SECAM WebControls is the web interface of the SECAM Web Monitoring System. It provides users with the ability to view their video streams and change their settings anywhere there is an internet connection.

# Table of Figures

## Contributing Individuals

| | |
|---|---|
| **Caleb Shortt:** | General Editor, Introduction, Prototypes, Reporting, Monitoring, Correcting, Testing, Implementation |
| **Moffat Sehudi:** | Test Cases, Modules, Pseudo-code, Diagrams, Performance Requirements |
| **David Cheperdak:** | Class Diagrams, UML Diagrams, Sequence Diagrams, Diagrams, Development Plan |
| **Vikram Sandhu:** | Use Cases, Test Cases, Sequence Diagrams, Diagrams |
| **Tubego Phamphang:** | Use Cases, Test Cases, Sequence Diagrams, Modules, Diagrams |