CARLETON UNIVERSITY
SCHOOL OF COMPUTER SCIENCE
COMP 3004 – OBJECT ORIENTED SOFTWARE ENGINEERING

CUBook

Requirements Analysis Document

Team Redundancy Team Squad
Andrew Thompson
Marton Kilian
Sina Keang

Submitted to:
Dr. Christine Laurendeau

Contents

1. Introduction……
    1.1 Purpose of System……
    1.2 Overview of Document……
2. Proposed System……
    2.1 Overview……
    2.2 Functional Requirements……
    2.3 Non-Functional Requirements……
    2.4 System Models……
        2.4.1 Use Case Model……
        2.4.2 Object Model……
            2.4.2.1 Data Dictionary……
            2.4.2.2 Class Diagrams……
        2.4.3 Dynamic Model……
        2.4.4 User Interface……
3. Glossary……

Figures

Figure 1 – High-Level Use Case Diagram
Figure 2 – Login Use Case Diagram
Figure 3 – Edit Info Diagram
Figure 5 – Student Flash Board Diagram
Figure 6 – Professor Flash Board Diagram
Figure 7 – Upload Picture Diagram
Figure 8 – NewsFlash / FlashSeg Error Diagram
Figure 9 – Initiation Participation Diagram

Figure 10 – Entity Objects
Figure 11 – Control / Boundry Objects for Login
Figure 12 – Control / Boundry Objects for Main
Figure 13 – Control / Boundry Objects for ViewFlashBoard
Figure 14 – Control / Boundry Objects for ViewFlashSegment
Figure 15 – Control / Boundry Objects for ViewHomePage
Figure 16 – State Chart Login
Figure 17 – State Chart File Browser
Figure 18 – State Chart MainControl
Figure 19 – State Chart Change Avatar
Figure 20 – State Chart FlashBoardControl
Figure 21 – State Chart FlashSegmentControl
Figure 22 – State Chart Filter FlashFeed
Figure 23 – State Chart Push Update
Figure 24 – Sequence Login Diagram
Figure 25 – Sequence Login Error Diagram
Figure 26 – Sequence FlashSegment Diagram
Figure 27 – Sequence FlashSegment Diagram with Errors
Figure 28 – Sequence FlashBoard Diagram
Figure 29 – Sequence UploadPicture Diagram
Figure 30 – Sequence UploadPicture Diagram with errors
Figure 30b – Sequence PostFlashSegment Diagram
Figure 31 – Sequence PostFlashSegment Diagram with Errors
Figure 32 – Sequence NewsFlash Diagram
Figure 33 – Sequence NewsFlash Diagram with Errors
Figure 34 – Sequence Filter FlashFeed Diagram
Figure 35 – Sequence UpdateAvatar Diagram
Figure 36 – Sequence UpdateAvatar Diagram with Errors
Figure 37 – Sequence DeleteFlashSegment Diagram
Figure 38 – Sequence DeleteFlashSegment Diagram with Errors
Figure 39 – Sequence DeleteNewsFlash Diagram
Figure 40 – Sequence DeleteNewsFlash Diagram with Errors

Tables

Table 1 – Functional Requirements
Table 2 – Non-Functional Requirements
Table 3 – High Level Use Case Descriptions
Table 4 – Detailed Use Case Descriptions
Table 5 – Entity Objects
Table 6 – Boundary Objects
Table 7 – Control Objects

1. Introduction

Throughout this document, we will be describing a social interaction program called CUBook. CUBOOK is a social networking tool aimed at promoting learning and social participation. The intended users are the students enrolled in the University of Carleton. CUBOOK allows students to share their thoughts and questions to all other users enrolled in the same course field. This social participation is what nurtures a sense of community among the users and enhances the productivity of the student's experience in the university institution.

1.1 Purpose of System

The system we are creating should be able to allow students to communicate to each other in their respective courses in a manner more convenient than currently available networking tools. This system should allow for professors and students to work together to create a better educational environment for students, and make the lives of professors easier.

The system at hand should be easy to navigate and use, so that it is accessible to all students and all professors, regardless of the program they are taking. This system should not require extensive training or technical knowledge to utilize. Errors should be obvious and usability (user friendliness) should be of the utmost importance.

1.2 Overview of Document

This document describes the CUBook system at a very high and very low level. This document should not leave the reader with any doubts as to the capabilities of the system, or how the

system will look once completed. Through this document we hope to find any errors or problems that may arise in the development of the actual system and stop them before they become serious design flaws in later builds.

In section 2, we go through an overview of the system and list off the functional and non-functional requirements of the system. Later in section 2, we go over the Use Case model detailing all the use case diagrams we present. Further on, we discuss the Class Diagrams, Entity, Boundary, and Control objects that are used in the creation of our system. We finish section 2 with screen shots of a prototype of the User Interface.

2. Proposed System

In this section, we will go over an overview of the various functionality that the system is capable of both a high and low level. We discuss the various states that the system can be in, the flow of events that leads from one action to another and the sequence of events with which those actions operate.

2.1 Overview

In this section we start by listing the function and non-functional requirements, then we go into the use case model, the object model, the dynamic model, and then finally the actual user interface itself.

2.2 Functional Requirements

In this section we discuss the functional requirements that are required of the system. Functional Requirements are in essence, things that the program has to be able to do, specifically features. They do not include restrictions on the system.

Table 1 – Functional Requirements

	Functional Requirements
	Client Side
F-01	Users must be able to launch CUBook
F-02	User must be able to connect to the server
F-02-1	Users must be able to disconnect from the CUBook server.
F-03	Users must be able to access their account off the server
	Users
F-04	Users must be able to view their profile photo on the main page
F-05	Users must be able to upload and replace their profile photo on the main page
F-06	Users must be able to view their Flashfeed on the main page, which is a chronological list of all newsflashes from flashboards that the user has access to. This should be the first thing that the user sees when they log in.

F-07	Items displayed on the flash feed must include the avatar of the person who posted it, the location of the post, and the contents of the post
F-08	The flash feed must be filterable based on the courses the user is enrolled in
F-09	Filtering settings of a Flash Feed must be saved at the end of each user session, and must be applied the next time the user logs in.
F-10	Users must have access to their personal information including their ID Number and Full Name
F-11	Users must have access to the flash boards associated with each individual class that the user is registered too.
F-12	Users must be able to post Flash Segments which are essentially folders that can contain news flashes.
F-13	Users must be able to post News Flashes on Flash Segments which can contain any combination of pictures, text notes, or web links
F-14	Users must be able to view Flash Segments on a Flash Board. Flash Segments must have a title.
F-15	Users must be able to view News Flashes in a Flash Segment
	Students
F-16	Must have access to their avatar from the main page.
F-17	Must have the ability to change their avatar from the main page.
F-18	Must have their avatar show up in any news flashes that they post.
	Instructors
F-19	Must have a real name that is visible on the main page.
F-20	Must have their real name show up on news flashes.
F-21	Must have a unique marking that makes their news flashes easily identifiable on flash boards
F-22	Must have the ability to delete flash segments made on the flashboard for the course they are teaching.
F-23	Must have the ability to delete news flashes made on a flash segment for the course they are teaching.
F-24	Professor User must be able to view the student name, student id, and summary of posts by the student in the course the Professor is teaching. Theses posting will be in chronological order from the beginning of the term to the present.

Table 2 – Non-Functional Requirements

2.3 Non Functional Requirements

In this section we discuss the non-functional requirements that are required of the system. Non-Functional requirements display restrictions on the system we are building. They are constraints on the system that have to be kept in mind when the system is being built.

Non-Functional Requirements

Dependability

NF-01	The user should not experience any disconnects due to server software failure
NF-02	The server should handle all forms of user input without causing corruption to storage, or causing the server to crash
	The server should keep a backup of the database in case of a critical system failure. This NF should be replaced with something more relevant. If it were an issue to maintain a backup of the data base, it would be more logical to have the backup handled on a hardware layer and not by the server it self.

Usability

NF-03	All error messages should be clear and understandable by students and professors.
NF-04	The login screen should display the Carleton logo.
NF-05	User settings should be saved on the server and not the client so that user settings will migrate from computer to computer. I.E. It should not matter where you log in.
NF-06	When a user updates something in their profile, that information should be pushed out to connected users
NF-07	If a user updates their profile information, it should change for all news flashes made before the update as well as after the update.
NF-08	User interface should be intuitive and not require extensive training to understand

Supportability

NF-09	When flash segments are deleted, all newsflashes inside the flash segments will be deleted
NF-10	User should never have the same avatar
NF-11	Avatar name must not exceed a certain length 24 characters
NF-12	Users must have default photo for the profile photo.

Performance

NF-13	The server should provide all connected users with updates within 3 seconds after they have been processed by the server
NF-14	The server should authenticate any logins within 3 seconds of being submitted.
NF-15	Updates should only occur when they are required or when the client requests them.
NF-16	Updates should be handled on a FIFO basis, and pushed out to clients with a response time of 3 seconds or less.
NF-17	CUBook system must support a minimum of six concurrent client processes

NF-18	User's must not be able to have more than one profile photo uploaded to the server at a time
	Implementation
NF-19	The system should be written in C++
NF-20	The system should be executable on the SCS Lambda Network
NF-21	Each client process must support a single user and execute on a different machine other than the host(server) machine.

2.4 System Models

System models are a way of modeling a large system without creating the system itself. Through this method, we are able to show the functionality of the system we are building at a very detailed level. This allows us to discern errors, problems, and changes that need to occur before the actual creation of the system occurs. In this section, we will be displaying the Use Case Model, Object Model, and Dynamic Model.

2.4.1 Use Case Model

The Use Case Model shows the basic functionality of the system at hand. In our case, it shows basic functionality like logging into the system, displaying various pages throughout the CUBook Interface. It also shows basic functionality for manipulating the system, including deleting and creating objects. This section starts with the actual Use Case Diagrams, and then goes into detail on how each use case within the diagram operates.

The highest level view of the system, users can be broken down into Professors or Students.
Any user can initiate the CUBook client, and then log in to the CUBook server.

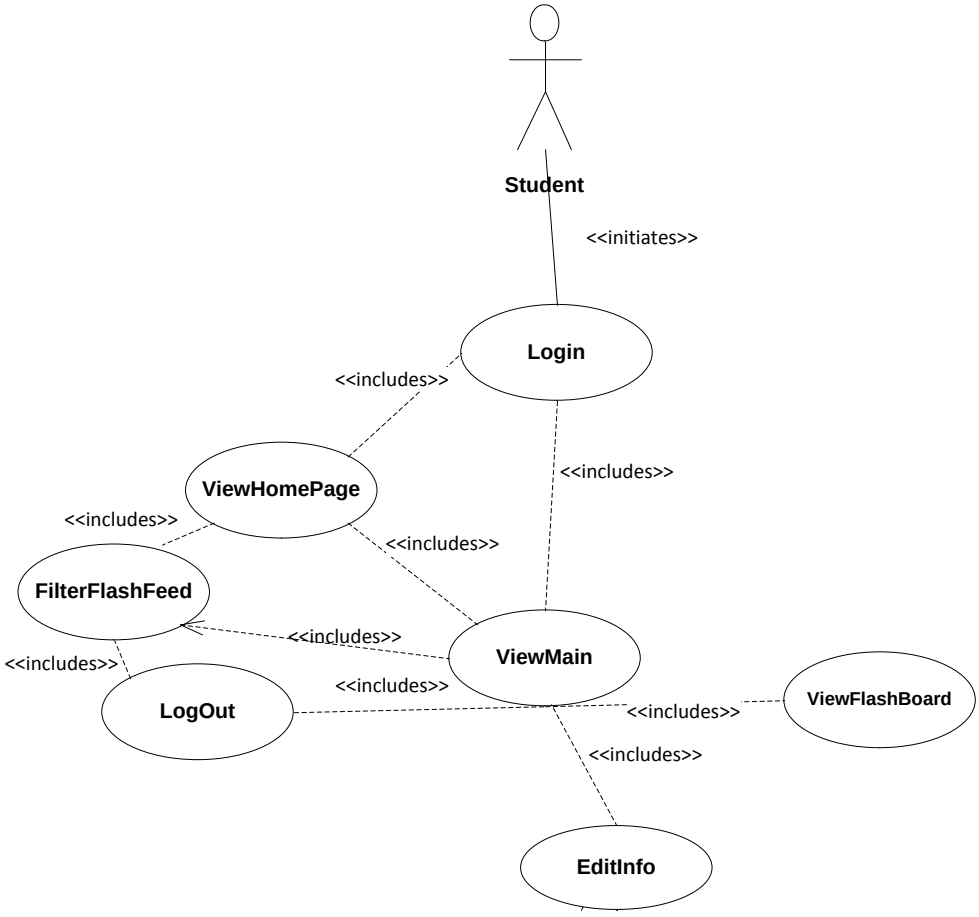


Figure 1 – High-Level Use Case Diagram

Table 3 – High Level Use Cases

UC-L-01	LaunchCUBook	The user starts up the CUBook client.
UC-L-02	Login	The user logs into the CUBook server using their student number and password.

Details the initiation of the CUBook client, and then logging into the CUBook server. Also dispalys the various errors that can occur while attempting to do so.

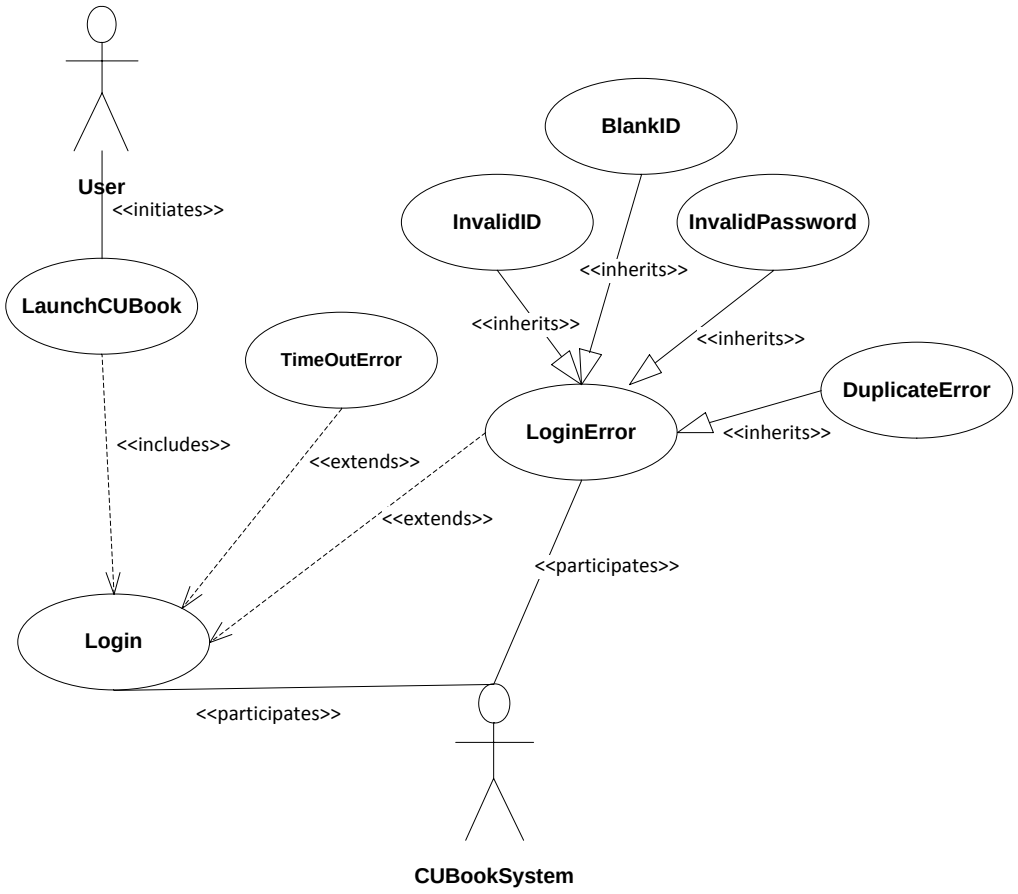


Figure 2 – Login Use Case Diagram

Details the process of changing the avatar and profile picture

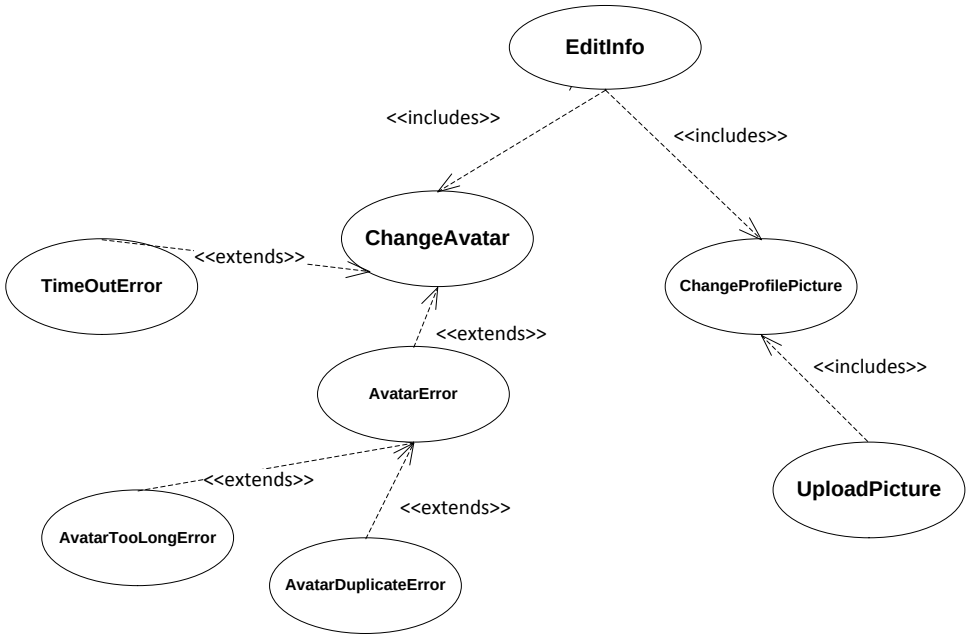


Figure 3 – Edit Info Diagram

Details the interactions between a student and a course Flash Board. Shows the various actions that can be done while on a Flash Board, or a Flash Segment Page within a Flash Board.

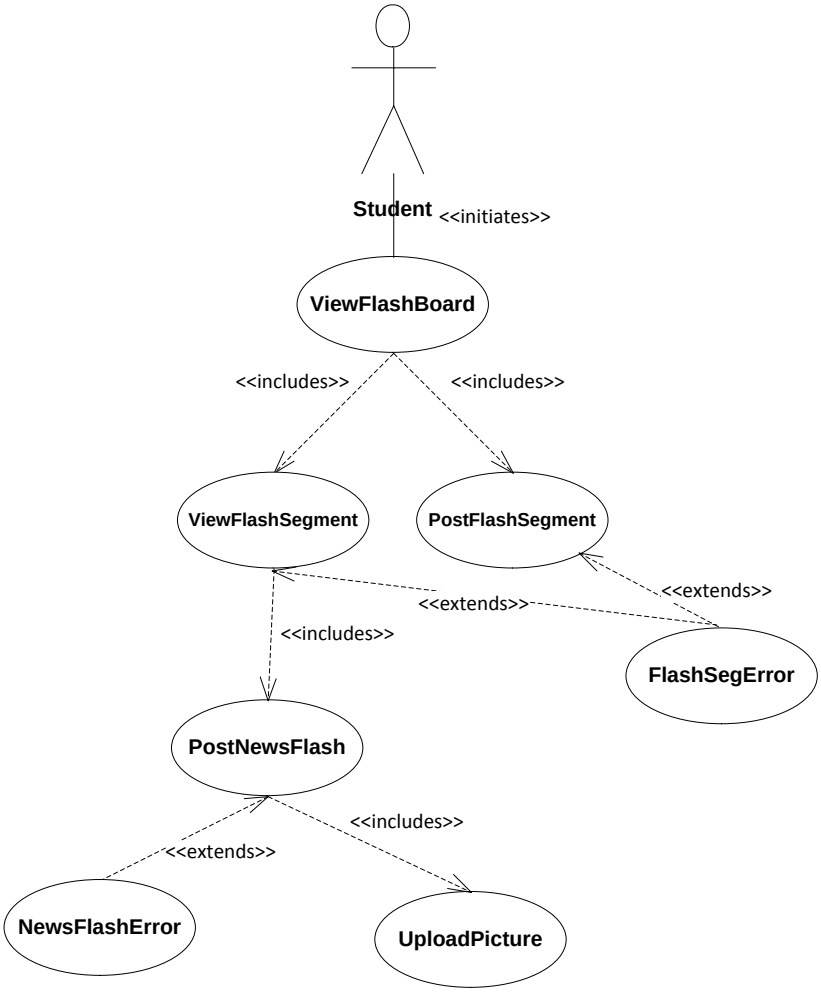


Figure 5 – Student Flash Board Diagram

Details the interactions between a professor and a course Flash Board. Shows the various actions that can be done while on a Flash Board, or a Flash Segment Page within a Flash Board. Different from students in that the Professor has the ability to delete flash segment and news flashes, as well as view student's profiles.

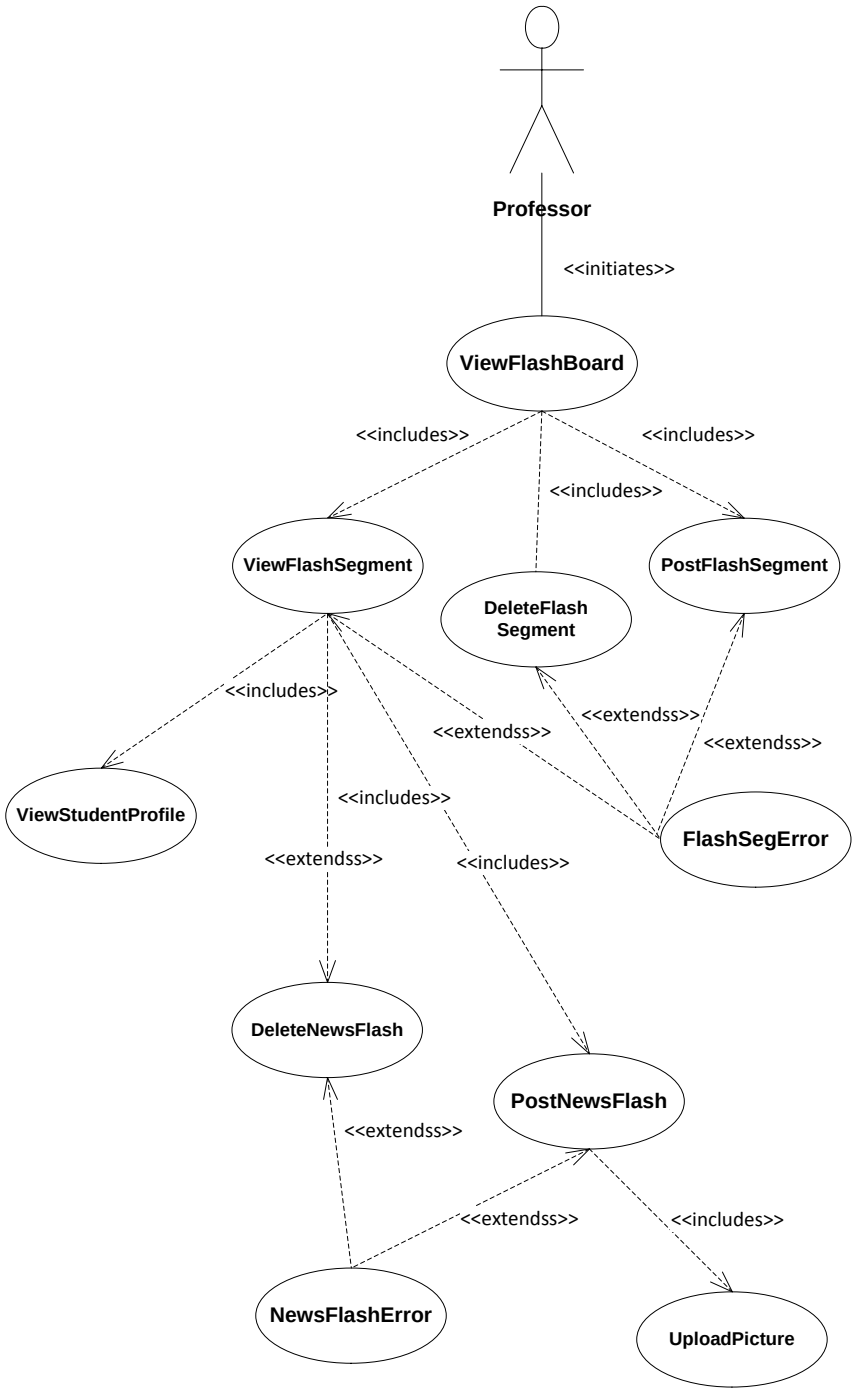


Figure 6 – Professor Flash Board Diagram

Details the process of uploading a picture to CUBook. UploadPicture Uses a file browser to let a user select a photo and upload it. Where the Photo goes will depend on the use case calling on it.

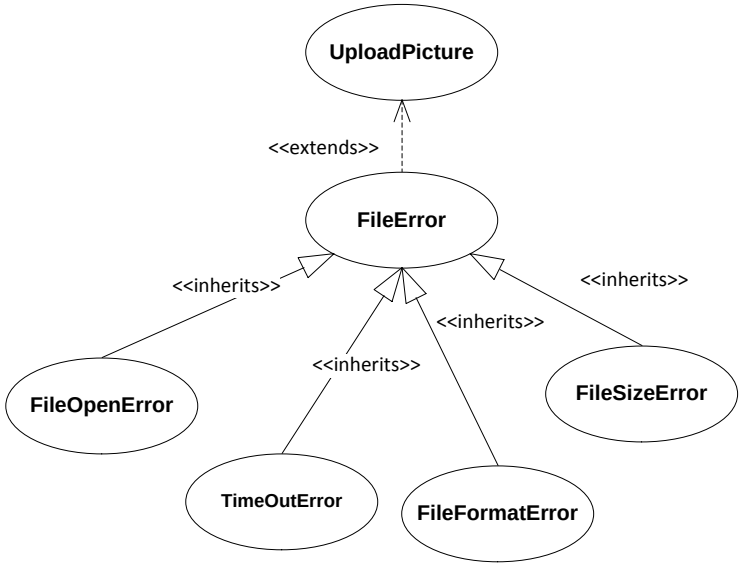


Figure 7 – Upload Picture Diagram

Details the various errors that can occur while the user is manipulationg Flash Segments and News Flashes.

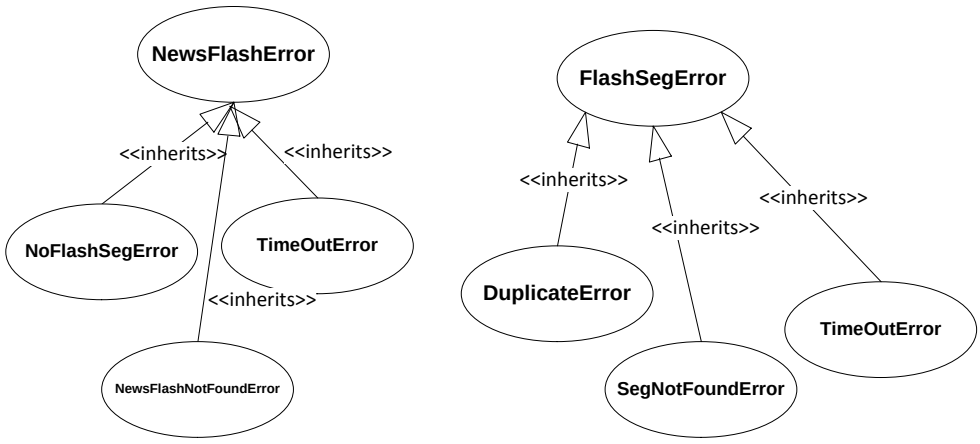


Figure 8 – NewsFlash / FlashSeg Error Diagram

Details the various interactions between actors and use cases. Shows who is participating and who is initiating any actions that can occur within the system.

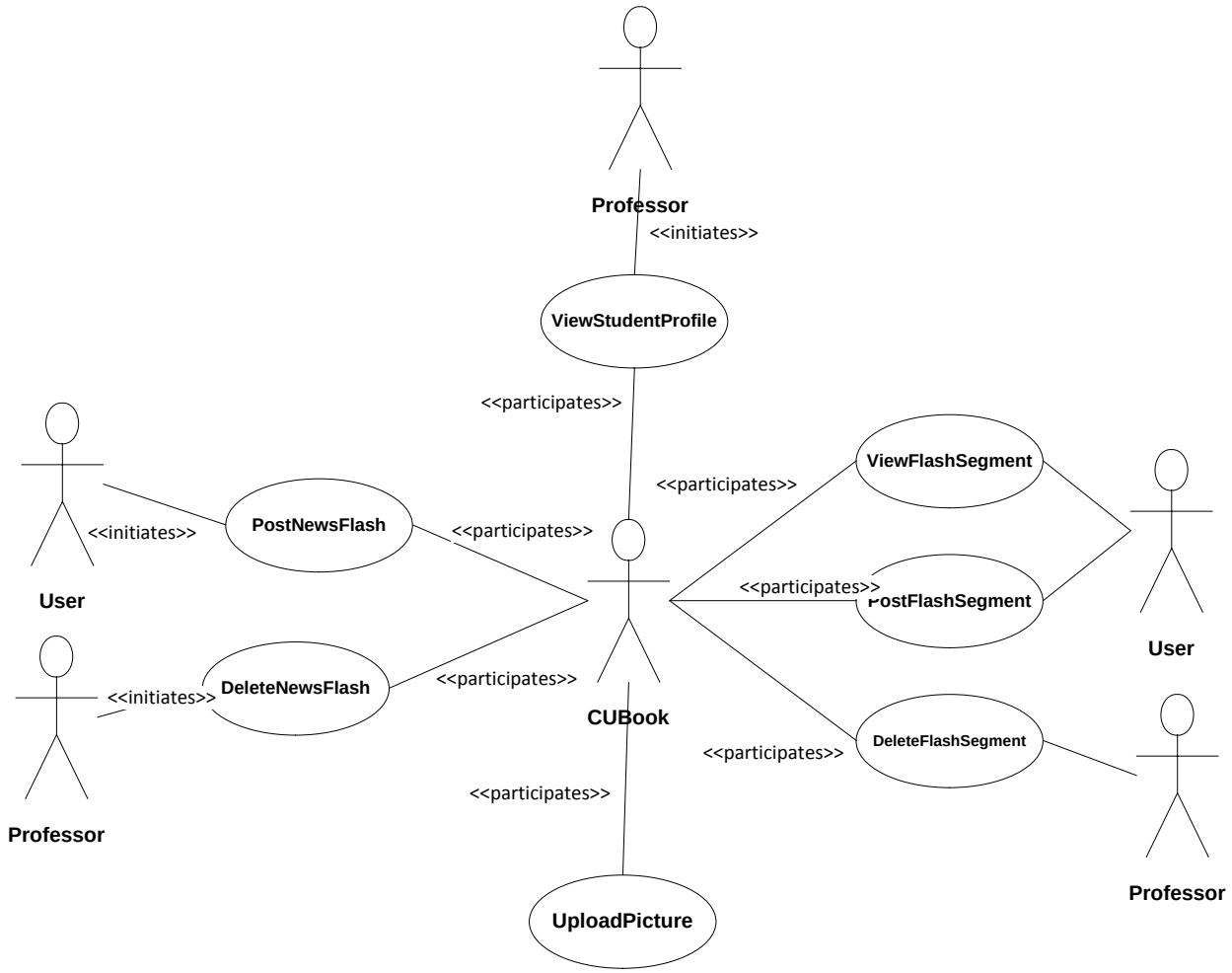


Figure 9 – Initiation Participation Diagram

Table 4 – Detailed Use Case Descriptions

UC-L-01	LaunchCUBook	The user starts up the CUBook client.
UC-L-02	Login	The user logs into the CUBook server using their student number and password.
UC-L-03	LoginError	Specifies situations where the user attempts to log in but is unsuccessful (generalizes into LoginError, InvalidId, BlankID, InvalidPassword, DuplicateError, TimeOutError)
UC-L-03-1	InvalidID	Error in which the user provides an ID that is not in the

		database.
UC-L-03-2	BlankID	Error in which the user provides no ID and attempts to login.
UC-L-03-3	InvalidPassword	Error in which the user provides a password that does not match up with the studentID/password combination in the data base.
UC-L-03-4	DuplicateError	Error in which the user attempts to log in but is already logged in in another location.
UC-L-04	TimeOutError	Error in which the server does not give a reply back to the client for any unspecified reason.
UC-M-01	ViewHomePage	Takes the user to their home page which contains their flash feed.
UC-M-02	ViewMain	Shows the main display which is a bar on the left hand side and top of the screen.
UC-M-03	ViewFlashBoard	Takes the user to the flashboard for the selected course.
UC-M-04	Logout	Disconnects from the server, after loging out the user will stop receiving updates and be brought back to the Login screen
UC-M-05	EditInfo	Allows the user to edit various information that is part of their profile.
UC-M-05-1	ChangeAvatar	Allows the user to change his avatar
UC-M-05-2	AvatarError	Specifies situations where the user's chosen avatar does not change successfully (generalizes into DuplicateError,and AvatarTooLongError)
UC-M-05-2-1	AvatarDuplicateError	Error in which the avatar entered is already in use by another user.
UC-M-05-2-2	AvatarTooLongError	Error in which the avatar entered is longer than the maximum size allowed.
UC-M-05-3	ChangeProfilePicture	Allows the user to change his profile picture
UC-M-06	FilterFlashFeed	Allows the user to filter their FlashFeed on the home page through a list of courses on the left hand side.
UC-U-1	UploadPicture	Allows the user to upload a picture using a file browser
UC-U-2	FileError	There was an error uploading or reading the selected picture (generalized by FileOpenError, TimeOutError, FileFormatError, FileSizeError). The correct error is displayed to the user and the operation aborts.
UC-U-2-1	FileOpenError	Error in opening the file on the client machine.
UC-U-2-2	FileFormatError	Error in the format of the picture that is being uploaded. I.E. trying to upload a document instead of a picture.

UC-U-2-3	FileSizeError	Error in which the submited picture is to large to be submited to the server
UC-F-1	ViewFlashSegment	The user selects a flash segment from the list and is shown the Flash Segment page with a list of news flashes
UC-F-2	PostFlashSegment	The user creates a flash segment on the flash board.
UC-F-3	DeleteFlashSegment	The professor chooses to delete the flash segment and all news flashes contained within.
UC-F-4	FlashSegError	Error that generalizes into DuplicateError, TimeOutError, and NotFoundError. Related to the viewing of, destruction of, and creation of flash segments
UC-F-4-1	DuplicateError	Error in which the flash segment title chosen already exists on the flash board.
UC-F-4-2	SegNotFoundError	Error in which the flash segment selected to view or delete no longer exists (was deleted between updates).
UC-F-5	PostNewsFlash	The user creates a news flash within a flash segment.
UC-F-6	DeleteNewsFlash	The professor deletes the specific news flash
UC-F-7	NewsFlashError	Generalizes into NoFlashSegError, and NewsFlashNotFoundError, relates to the creation of news flashes
UC-F-7-1	NoFlashSegError	Error in which the flash segment that the user is attempting to create a news flash on no longer exists.
UC-F-7-2	NewsFlashNotFoundError	Error in which the News Flash that the user is attempting to delete no longer exists.
UC-F-8	ViewStudentProfile	The professor views a students profile using their avatar on a Flash Segment.

Use Case Identifier	UC-L-01
Name	**LaunchCUBook**
Participating Actors	Initiated by: Student or Professor
Flow of Events	1. The user starts up the CUBook Client Interface 2. The client displays a log in screen which has: The Carleton Logo, a text field for the user's student id, a text field for the user's password, and a login button 3. If the user clicks log in, it sends a login request to the CUBook server (includes use case **Login**).
Entry Conditions	The user launches CUBook
Exit Conditions	The user successfully logs in or terminates the CUBook

Quality Requirements	The client should be executable on the SCS lambda servers.
Traceability	F-01, NF-20
Use Case Identifier	UC-L-02
Name	**Login**
Participating Actors	Initiated by: Student or Professor; CUBook Participates
Flow of Events	1. The user inputs their username and password and submits it to CUBook for authentication. 2.CUBook authenticates the username and password. If successful, CUBook gives the client their profile including all their saved settings, and directs them to their home page and the main window. 3. If this is the first time the user has logged in, a default photo is provided.
Entry Conditions	The user has entered in their information and selected login.
Exit Conditions	The user has logged in.
Quality Requirements	
Traceability	F-02, F-03, NF-12
Use Case Identifier	UC-L-03
Name	**LoginError**
Participating Actors	Student,Professor,CUBook
Flow of Events	1. The system notifies the user that an error occurred while attempting to login to CUBook
Entry Conditions	Authentication failed.
Exit Conditions	The login operation is aborted.
Quality Requirements	
Traceability	NF-03
Use Case Identifier	UC-L-03-1
Name	**InvalidID**
Participating Actors	Student,Professor,CUBook
Flow of Events	2. The system notifies the user that the ID they entered was not found in the database. (specializes use case **LoginError**)
Entry Conditions	Authentication fails due to improper ID
Exit Conditions	The login operation is aborted.
Quality Requirements	
Traceability	NF-03
Use Case Identifier	UC-L-03-2
Name	**BlankID**
Participating Actors	Student,Professor,CUBook
Flow of Events	2. The system notifies the user that the ID they entered was blank (specializes use case **LoginError**).
Entry Conditions	Authentication fails due to no ID being provided.
Exit Conditions	The login operation is aborted.

Quality Requirements	
Traceability	NF-03
Use Case Identifier	UC-L-03-3
Name	**InvalidPassword**
Participating Actors	Student,Professor,CUBook
Flow of Events	2. The system notifies the user that the password provided did not match the ID/Password combination in the system. (specializes use case **LoginError**).
Entry Conditions	Authentication fails due to an incorrect password being provided
Exit Conditions	The login operation is aborted.
Quality Requirements	
Traceability	NF-03
Use Case Identifier	UC-L-03-4
Name	**DuplicateError**
Participating Actors	Student,Professor,CUBook
Flow of Events	2. The system notifies the user that the ID entered is already logged in (specializes use case **LoginError**).
Entry Conditions	Authentication fails due to a duplicate user already being logged in.
Exit Conditions	The login operation is aborted.
Quality Requirements	
Traceability	NF-03
Use Case Identifier	UC-L-04
Name	**TimeOutError**
Participating Actors	Student,Professor,CUBook
Flow of Events	1.. The system notifies the user that the CUBook server is not responding to it's requests.
Entry Conditions	User makes a request on the system and does not get a reply in 20 seconds or less.
Exit Conditions	The current operation is aborted
Quality Requirements	
Traceability	NF-03
Use Case Identifier	UC-M-01
Name	**ViewHomePage**
Participating Actors	Initiated by: Student or Professor; CUBook Participates
Flow of Events	1. The system generates a flashfeed based on a chronological listing of all recent activity, relevant to the user. Each post will show the location of the post, the contents of the post, and the avatar or name of the person who posted it. 2. The panel displays the user's flashfeed to the user. 3. The FlashFeed gets filtered based on the filters selected from the filter list (include Use Case **FilterFlashFeed**).

Entry Conditions	A: The user has just logged into the CUBook server B: The user has clicked on his photo in the top left corner, bringing him back to his home page from any other page.
Exit Conditions	The user decides to go to another page
Quality Requirements	
Traceability	F-06, F-07

Use Case Identifier	UC-M-02
Name	**ViewMain**
Participating Actors	Initiated by: Student or Professor; CUBook Participates
Flow of Events	1. The left bar of the screen is populated with the user's profile, user's photo, and the user's list of registered courses. These courses can be selected to filter the user's FlashFeed (include use case **FilterFlashFeed**) 2. If the user is a student, the left hand bar is also populated with the user's avatar. 3. If the user is a professor, the left hand bar is populated with the user's real name. 4. If the user is a professor, their name is highlighted green to make them more apparent. 4. The top bar of the screen is populated with a title bar showing what page the user is currently looking at, a drop down menu of links to courses, and a log out button (include use cases **LogOut, ViewFlashBoard**). 5. A button is included on the side called Edit Profile allowing the user to change aspects of their profile (include use case **EditInfo)** 6. The remainder of the page is populated with the current page the user is viewing. If the user just logged in, the page shown is the user's home page (include use case **ViewHomePage**)
Entry Conditions	Displayed upon logging into CUBook
Exit Conditions	Once the user chooses to log out.
Quality Requirements	The main page includes the left hand bar and top bar. These bars should never change or move, and provide the user with a constant method of getting to any page in CUBook in as few clicks as possible.
Traceability	F-04, F-06, F-10, F-16, F-19, F-21

Use Case Identifier	UC-M-03
Name	**ViewFlashBoard**
Participating Actors	Initiated by: Student or Professor; CUBook Participates
Flow of Events	1. The page panel is populated with FlashSegments for the selected course. 2. A "Post Flash Segment" button is added to the page panel, allowing for users to post Flash Segments (include use case **PostFlashSegment**). 3. If the user is a professor, a delete link is placed next to every single flash segment listed in the page panel (include use case **DeleteFlashSegment**).
Entry Conditions	The user selects their desired course from the main window.
Exit Conditions	The user moves to another page, enters a flash segment, or attempts to post a flash

	segment.
Quality Requirements	
Traceability	F-11, F-14
Use Case Identifier	UC-M-04
Name	**Logout**
Participating Actors	Initiated by: Student or Professor; CUBook Participates
Flow of Events	1. The client tells the CUBook server that it is finished. 2. The client uploads it's filter list for the FlashFeed to the server (include use case **FilterFlashFeed**). 3. The server cleans up it's list of active users. 4. The client returns to the initial Launch screen with log in information.
Entry Conditions	The user selects the log out button from the top bar.
Exit Conditions	
Quality Requirements	The user's Filter should be updated to the server before log out is completed.
Traceability	F-02-1
Use Case Identifier	UC-M-05
Name	**EditInfo**
Participating Actors	Initiated by: Student or Professor; CUBook Participates
Flow of Events	1. A button appears next to the profile picture allowing the user to begin browsing for a photo. 2. If the user is a student, the user's Avatar becomes a text field already filled in with their old Avatar's name. The user may then choose to change that field. 3. The edit profile button changes to a "save changes" button.
Entry Conditions	The user selects the Edit Profile button from the left hand bar
Exit Conditions	The user selects the Save Changes button from the left hand bar.
Quality Requirements	
Traceability	F-05, F-16
Use Case Identifier	UC-M-05-1
Name	**ChangeAvatar**
Participating Actors	Initiated by: Student; CUBook Participates
Flow of Events	1. The text field gets sent to the CUBook server for authentication. 2. Once the chosen Avatar has been authenticated, it is pushed out to all active clients.
Entry Conditions	The user fills out the Avatar text field and presses the save changes button.
Exit Conditions	The user changes his avatar.
Quality Requirements	The chosen avatar must be under 24 characters and not be a duplicate of another avatar on the server.
Traceability	F-16, F-17
Use Case Identifier	UC-M-05-2

Name	**AvatarError**
Participating Actors	Student, CUBook
Flow of Events	1. The system notifies the student that the Avatar was not able to be changed.
Entry Conditions	The change avatar method has failed
Exit Conditions	The operation of editting information is aborted.
Quality Requirements	
Traceability	NF-03
Use Case Identifier	UC-M-05-2-1
Name	**AvatarDuplicateError**
Participating Actors	Student, CUBook
Flow of Events	2. The system notifies the student that the Avatar chosen is already in use and can not be used (generalizes into **AvatarError**).
Entry Conditions	The user attempts to enter an avatar that is already in use by another user.
Exit Conditions	The operation of editting information is aborted.
Quality Requirements	
Traceability	NF-03
Use Case Identifier	UC-M-05-2-2
Name	**AvatarTooLongError**
Participating Actors	Student, CUBook
Flow of Events	2. The system notifies the student that the Avatar chosen is too long and can not be used (generalizes into **AvatarError**).
Entry Conditions	The user attempts to enter an avatar that is greater than 24 characters.
Exit Conditions	The operation of editting information is aborted.
Quality Requirements	
Traceability	NF-11, NF-03
Use Case Identifier	UC-M-05-3
Name	**ChangeProfilePicture**
Participating Actors	Initiated by: Student or Professor; CUBook Participates
Flow of Events	1. The user selects the Change Profile Picture button next to their profile picture. 2. A file browser opens up allowing them to upload a photo (includes use case **UploadPicture**) 3. Once the photo they want to use is selected and uploaded, the server verifies the photo and pushes it out to all connected clients.
Entry Conditions	The user elects to change their profile picture while editing their profile.
Exit Conditions	The photo is uploaded and pushed out to all clients, the user sees their new photo.
Quality Requirements	
Traceability	F-16
Use Case Identifier	UC-M-06

Name	**FilterFlashFeed**
Participating Actors	Initiated by: Student or Professor; CUBook Participates
Flow of Events	1. The user checks or unchecks any courses they want to filter / unfilter on the left hand panel. 2. The Flash Feed's list of displayed News Flashes changes to reflect the desired courses. 3. Any courses selected in the Filter will be displayed when the User changes pages to the Home Page. 4. If the user is already on the home page, the FlashFeed is updated immediately. (include use case **ViewHomePage**). 5. When the user logs out, all filter settings get uploaded to the server and saved to the user's profile (include use case **LogOut**).
Entry Conditions	The user elects to upload a photo to a newflash or to their profile.
Exit Conditions	The user's photo is selected and uploaded.
Quality Requirements	UploadPicture should not be aware of any other functions that are utilizing it. It just serves as a file browser that sends the photo to the server. Any extra information on how the photo should be used or where it is going, is handled by the use case utilizing it.
Traceability	F-06, F-07, F-08
Use Case Identifier	UC-U-1
Name	**UploadPicture**
Participating Actors	Initiated by: Student or Professor; CUBook Participates
Flow of Events	1. A file browser opens up allowing the user to search for photos on his/her computer. 2. Once a photo is found that the user wishes to upload, they click the upload button. 3. The photo is sent to the CUBook server where it gets verified that it is a photo. 4. Once verification is complete, the photo gets submitted to the user's chosen destination.
Entry Conditions	The user elects to upload a photo to a newflash or to their profile.
Exit Conditions	The user's photo is selected and uploaded.
Quality Requirements	UploadPicture should not be aware of any other functions that are utilizing it. It just serves as a file browser that sends the photo to the server. Any extra information on how the photo should be used or where it is going, is handled by the use case utilizing it.
Traceability	F-05
Use Case Identifier	UC-U-2
Name	**FileError**
Participating Actors	Student,Professor,CUBook
Flow of Events	1. The system notifies the user that an error occurred while trying to upload the photo.
Entry Conditions	The upload failed.
Exit Conditions	The upload operation was aborted.
Quality Requirements	

Traceability	NF-03
Use Case Identifier	UC-U-2-1
Name	**FileOpenError**
Participating Actors	Student,Professor,CUBook
Flow of Events	2. The system notifies the user that the file that was trying to be uploaded could not be read, upload failed before the file was sent (specifies use case **FileError**).
Entry Conditions	The upload failed due to the file being unreadable.
Exit Conditions	The upload operation was aborted.
Quality Requirements	
Traceability	NF-03
Use Case Identifier	UC-U-2-2
Name	**FileFormatError**
Participating Actors	Student,Professor,CUBook
Flow of Events	2. The system notifies the user that the photo was uploaded but the format of the file was not a recognizable picture format (specifies use case **FileError**).
Entry Conditions	The upload failed because the file format was not a recognizable format.
Exit Conditions	The upload operation was aborted.
Quality Requirements	
Traceability	NF-03
Use Case Identifier	UC-U-2-3
Name	**FileSizeError**
Participating Actors	Student,Professor,CUBook
Flow of Events	2. The system notifies the user that the picture was partially uploaded but the server rejected it when the size of the file was either not given or too large. (specifies use case **FileError**)
Entry Conditions	The upload failed because the file was too large or not given.
Exit Conditions	The upload operation was aborted.
Quality Requirements	
Traceability	NF-03
Use Case Identifier	UC-F-1
Name	**ViewFlashSegment**
Participating Actors	Initiated by: Student or Professor; CUBook Participates

Flow of Events	1. The user selects a desired flash segment from the list of flash segments. 2. The user is given a list of news flashes from the CUBook server 3. A button called "Post News Flash" is displayed allowing the user to post a news flash on this flash segment (include use case **PostNewsFlash**). 4. If the user is a professor, each news flash will have a delete link next to it (include use case **DeleteNewsFlash**). 5. If the user is a professor, he/she may select any given Avatar who has posted on the flash segment to view that student's profile (include use case **ViewStudentProfile**).
Entry Conditions	The user selects a flash segment from the flashboard
Exit Conditions	The user changes pages, attempts to post a news flash, or views a student's profile
Quality Requirements	
Traceability	F-15, F-18, F-20
Use Case Identifier	UC-F-2
Name	**PostFlashSegment**
Participating Actors	Initiated by: Student or Professor; CUBook Participates
Flow of Events	1. The user selects the Post Flash Segment button on the flash board. 2. A text field is displayed allowing the user to enter a name, and the "Post Flash Segment" button is replaced by a "Submit" button. 3. Once the user has given the flash segment a name and pressed the Submit button, the flash segment name is submitted to the CUBook server for authentication. 4. Once the name is authenticated, it gets pushed out to all clients who are connected to the server.
Entry Conditions	The user is on a flashboard and elects to post a new flash segment.
Exit Conditions	The flash segment is pushed out to the client and he/she sees it updated on the page.
Quality Requirements	
Traceability	F-12
Use Case Identifier	UC-F-3
Name	**DeleteFlashSegment**
Participating Actors	Initiated by: Professor; CUBook Participates
Flow of Events	1. The user selects the delete button next to a flash segment on a flash board. 2. The client sends a message to the server asking for the flash segment to be deleted. 3. The server authenticates the request and then pushes out the new Flash Segment list to all connected clients.
Entry Conditions	A professor is on a flashboard and elects to delete a flash segment.
Exit Conditions	The flash segment is deleted and the new list is pushed to the client from the server.
Quality Requirements	
Traceability	F-22
Use Case Identifier	UC-F-4
Name	**FlashSegError**
Participating Actors	Student,Professor,CUBook

Flow of Events	1. The system notifies the user that an error occurred while attempting to interact with a Flash Segment
Entry Conditions	Attempting to post, view, or delete a flash segment failed.
Exit Conditions	The operation is aborted
Quality Requirements	
Traceability	NF-03
Use Case Identifier	UC-F-4-1
Name	**DuplicateError**
Participating Actors	Student,Professor,CUBook
Flow of Events	2. The system notifies the user that the flash segment could not be posted because another flash segment with the same name is already on that flash board. (specifies use case **FlashSegError**).
Entry Conditions	The user attempted to post a Flash Segment that already exists on the Flash Board.
Exit Conditions	The Post Flash Segment operation is aborted.
Quality Requirements	
Traceability	NF-03
Use Case Identifier	UC-F-4-2
Name	**NotFoundError**
Participating Actors	Student,Professor,CUBook
Flow of Events	2. The system notifies the user that the Flash Segment he/she is attempting to view or delete no longer exists (specifies use case **FlashSegError**).
Entry Conditions	The user attempted to view or
Exit Conditions	The user attempts to view or delete a Flash Segment that no longer exists
Quality Requirements	
Traceability	NF-03
Use Case Identifier	UC-F-5
Name	**PostNewsFlash**
Participating Actors	Initiated by: Student or Professor; CUBook Participates
Flow of Events	1. The user selects the "Post News Flash" button located on the Flash Segment. 2. A text field appears as well as an upload photo button, and the "Post News Flash" button changes to a "Submit" button. 3. The user enters any text or weblinks they would like into the text field 4. The user attatches any photos that they would like to be submitted (include use case **UploadPicture**). 5. The user selects the Submit button and sends the News Flash to the server to be authenticated. 4. Once the news flash is authenticated, it gets pushed out to all clients connected to the server.
Entry Conditions	The user is on a flash segment and selects the "Post News Flash" button.
Exit Conditions	The news flash gets pushed back to the user after being authenticated by the server.

Quality Requirements	
Traceability	F-13, F-18
Use Case Identifier	UC-F-6
Name	**DeleteNewsFlash**
Participating Actors	Initiated by: Professor; CUBook Participates
Flow of Events	1. The user selects the delete link inside the News Flash on the Flash Segment 2. The client sends a message to the server asking for the News Flash to be deleted. 3. The server authenticates the request and then pushes out the new News Flash list to all connected clients.
Entry Conditions	A professor is on a flashboard and elects to delete a News Flash.
Exit Conditions	The News Flash is deleted and the new list is pushed to the client from the server.
Quality Requirements	
Traceability	F-23
Use Case Identifier	UC-F-7
Name	**NewsFlashError**
Participating Actors	Student,Professor,CUBook
Flow of Events	1. The system notifies the user that there was an error in interacting with the news flash.
Entry Conditions	The user attempts to post or delete a news flash.
Exit Conditions	The operation is aborted.
Quality Requirements	
Traceability	NF-03
Use Case Identifier	UC-F-7-1
Name	**NoFlashSegError**
Participating Actors	Student,Professor,CUBook
Flow of Events	2. The system notifies the user that the flash seg that the user is attempting to delete or post a news flash on no longer exists (specializes use case **NewsFlashError**).
Entry Conditions	The user attempts to post or delete a News Flash.
Exit Conditions	The operation is aborted.
Quality Requirements	
Traceability	NF-03
Use Case Identifier	UC-F-7-2
Name	**NewsFlashNotFoundError**
Participating Actors	Student,Professor,CUBook
Flow of Events	2. The system notifies the user that the News Flash they are attempting to delete no longer exists (specializes use case **NewsFlashError**).
Entry Conditions	The user attempts to delete a News Flash.
Exit Conditions	The operation is aborted.
Quality Requirements	
Traceability	NF-03

Use Case Identifier	UC-F-8
Name	**ViewStudentProfile**
Participating Actors	Initiated by: Professor, CUBook participates
Flow of Events	1. The professor selects an avatar off a post on a flash segment. 2. The main page panel changes to the student's personal page. 2a. This shows an external flash feed of the student's activities in the course being taught by the professor who is viewing it. 2b. The professor also sees the student's real name and student ID.
Entry Conditions	The professor selects a student's avatar off of a flash segment
Exit Conditions	The operation is aborted.
Quality Requirements	
Traceability	F-24

2.4.2 Object Model

The Object Model is the model of all "things" in the system. This includes the virtual representation of actual things, or entities, as well as the components of the user interface that actors interact with, and the control objects that bridge the two of them together. For our system, this will include things like the main window panels, the flash boards, the flashboard controllers, and the users that interact with them.

2.4.2.1 Data Dictionary

The Data Dictionary is a listing of all objects that are included in the system. This includes the actual model data, the UI "Boundary Object" components that the users interact with, and the control objects that manage the two of them.

Table 5 – Entity Objects

Entity Object	Attributes / Associations	Definition
User	1. User ID 2. ProfilePicture	The user is a generic data type, and extends into Professors and Students.

Professor	3. Real Name	Professor is a type of user. They have additional capabilities beyond what students can do, including deleting News Flashes and Flash Segments, as well as viewing Student Profiles. They also have a unique identifier that makes them extremely visible when they post a News Flash. Otherwise they have all the same capabilities as a Student except that they can not have an Avatar, or change an Avatar
Student	3. Avatar	Students are types of Users. They have, and can change, an avatar (nick name). Students can also post on Flash Segments, and create new Flash Segments.
Course	1. course code 2. course name	Courses contain flash boards, and otherwise have no unique capabilities of their own.
FlashBoard		FlashBoards contain Flash Segments. There is 1 Flash Board per course, and 1 course per Flash Board.
FlashSegment	1. Title 2. Location	Flash Segments can contain News Flashes. This is where you will find conversations between Students and Professors.
NewsFlash	1. Text 2. Weblinks 3. Pictures 4. Location	This is the basic element of a Flash Segment. Flash Segments will be full of News Flashes, which each maintain text, links to websites, and pictures. These are the building blocks for conversations on CUBook
FlashFeed		The Flash Feed is a chronological listing of posts. Expands into Internal and External FlashFeeds
InternalFlashFeed		An Internal Flash Feed is the flash feed that you see as a user when you first log in on the home page. It shows a list of posts relevant to you, in any courses that you are registered in, in chronological order.
ExternalFlashFeed		External Flash Feeds are only viewable by Professors. When a professor views a student's profile, it will display their External Flash Feed, which is a chronological list of posts made by that student in the professor's course, since the beginning of the term.

Table 6– Boundary Objects

Boundary Object	Definition
MainPanel UI	This object is responsible for the visual output of the left and top panel of all windows. The profile photo and avatar(for Student) or real name(for professor) is display. The course drop down menu is located on the top right for navigating to flashboard. The middle left shows the courses that the flashfeed will show.
Message Protocol	This object sends required data to the Cubook Server from the Control object
Avatar Update Notification	This object update the status of the Avatar to confirm correctness
Flashboard UI	This object is responsible for the visual output of the Flash segments
Delete Segment Notification	This object notify that the segment has been deleted
Flash Segment UI	This object is responsible for the visual output of the newsflashes
Delete Notification	This object notify that the newsflash has been deleted
Post Segment Notification	This object notify that the flash segment has been add to the Flashboard
Post Newsflash Notification	This object notify that the newsflash has been add to the Flash Segment UI
File Browser	This object opens up a file browser to upload a profile photo
Save Photo Notification	This object notify that the photo has been uploaded
Login UI	This object is responsible for the Login screen output
Homepage UI	This object is responsible for the output of the flashfeed

Table 7– Control Objects

Control Object	Definition
MainControl	Manage the interactions between the user and the Main UI
FlashboardControl	Manage the interactions between the user and the Flashboard UI
Flash SegmentControl	Manage the interactions between the user and the Flash Segment UI

HomepageControl	Manage the interactions between the user and the Homepage UI
LoginControl	Manage the interactions between the user and the Client UI

2.4.2.2 Class Diagrams

This section details the class diagrams, which are diagrams showing how entities relate to each other, how boundary objects and control objects interact with each other, and how control objects interact with entity objects.

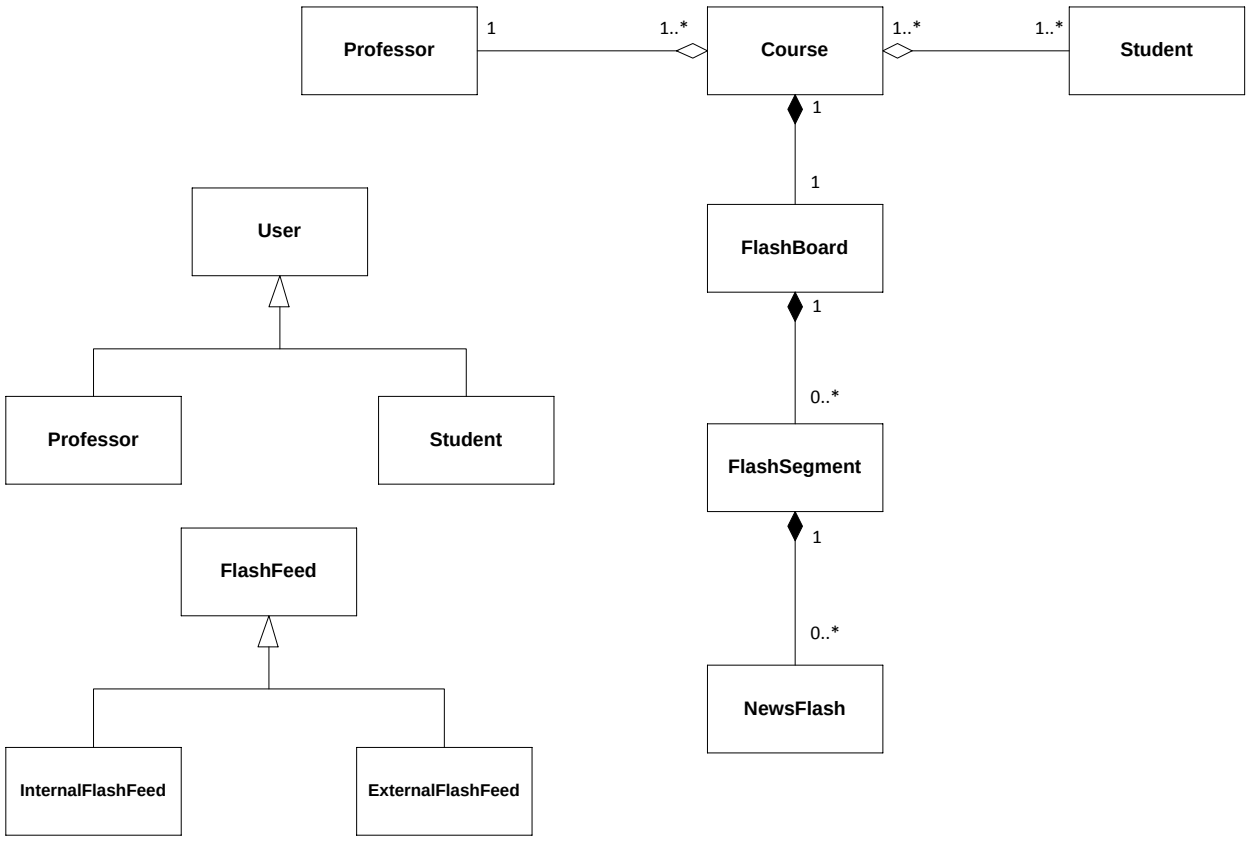


Figure 10 – Entity Objects

Figure 11 – Control / Boundry Objects for Login

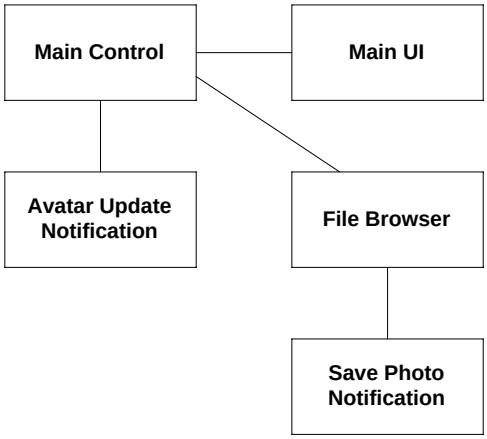


Figure 12 – Control / Boundry Objects for ViewMain

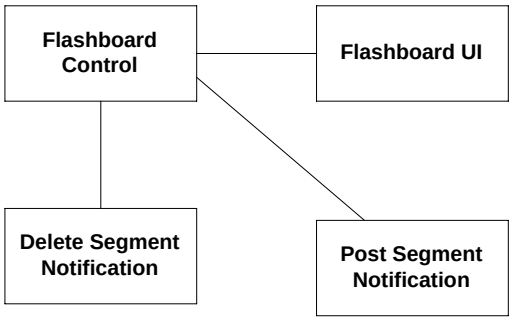


Figure 13 – Control / Boundry Objects for ViewFlashBoard

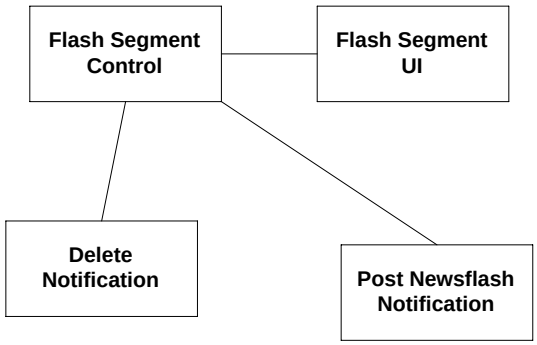


Figure 14 – Control / Boundry Objects for ViewFlashSegment

Figure 15 – Control / Boundry Objects for ViewHomePage

2.4.3 Dynamic Model

This section begins with a list of the various state machines that represent our system, and also shows the various sequences of events that components of the system undergo.

State Machine diagrams are diagrams that represent various control and boundary objects and the different states that they can be in at any given time. They also detail how the objects change from one state to another.

Sequence Diagrams show how Entity, Boundary, and Control objects interact with each other in a specific flow of events.

State Machine for LoginControl object (UC-02)

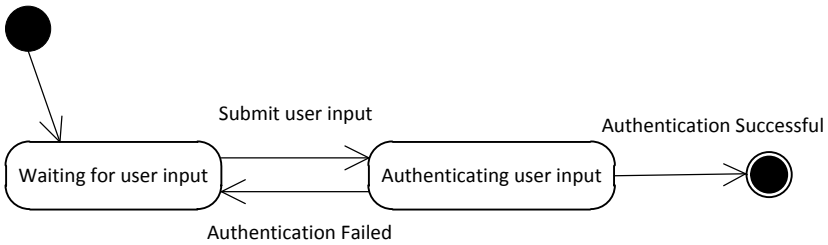


Figure 16 – State Chart Login

State Machine for MainControl's FileBrowser (UC-U-1)

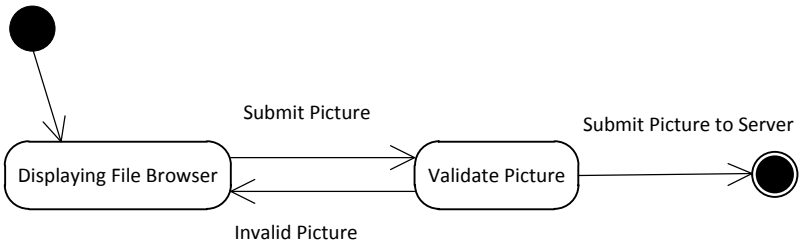


Figure 17 – State Chart FileBrowser

State Machine for MainControl object (UC-M-02)

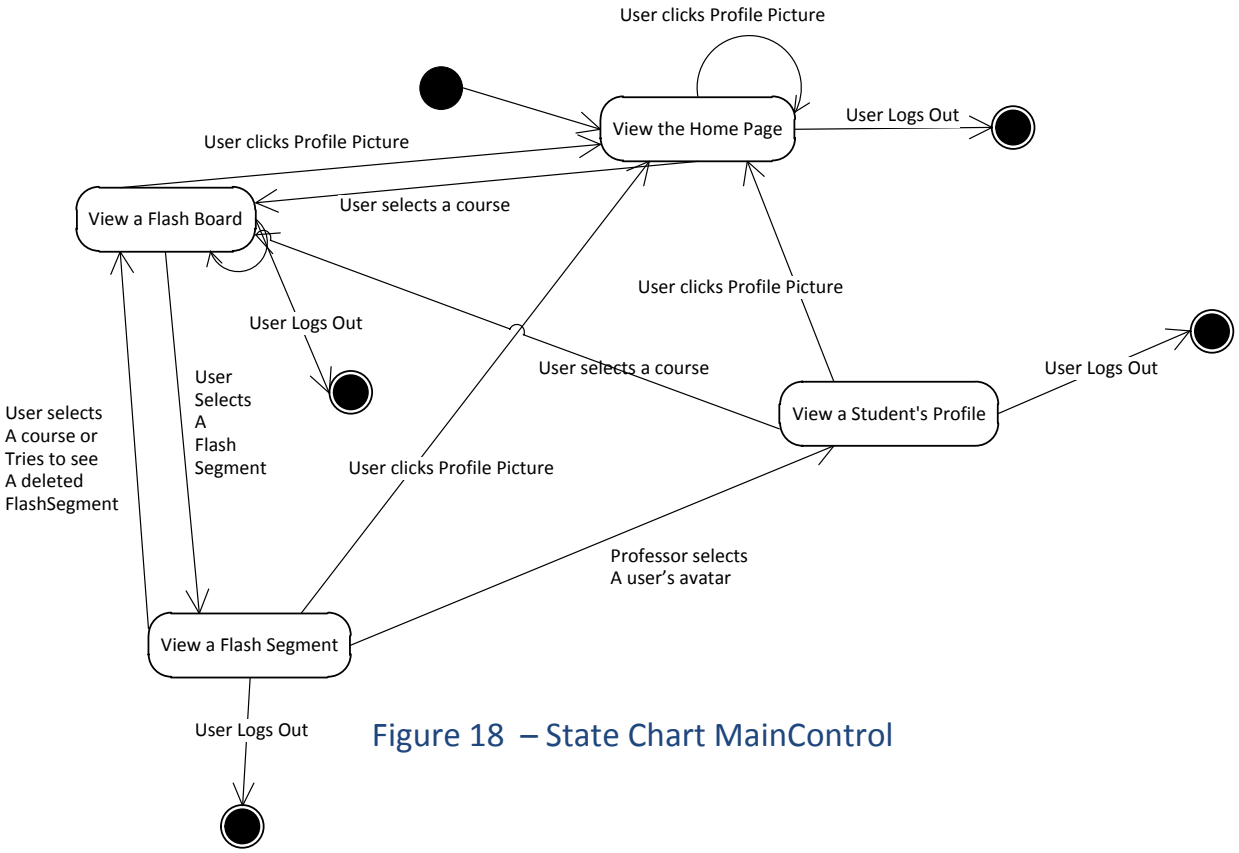


Figure 18 – State Chart MainControl

State Machine for MainControl's EditProfile (UC-M-05)

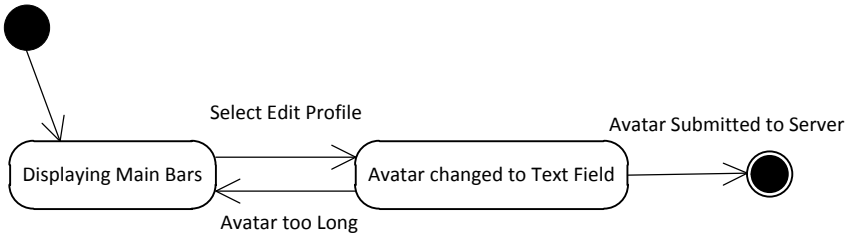


Figure 19 – State Chart ChangeAvatar

State Machine for FlashBoardControl object (UC-M-03)

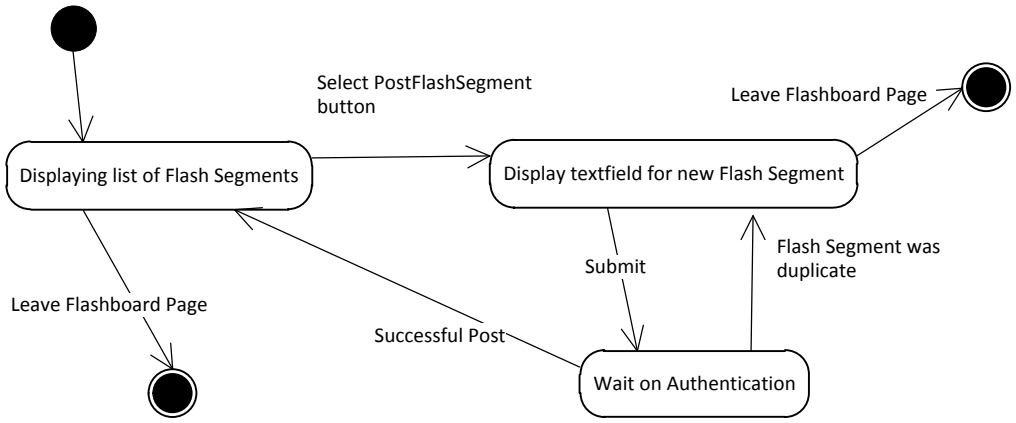


Figure 20 – State Chart FlashBoardControl

State Machine for FlashSegmentControl object (UC-F-1)

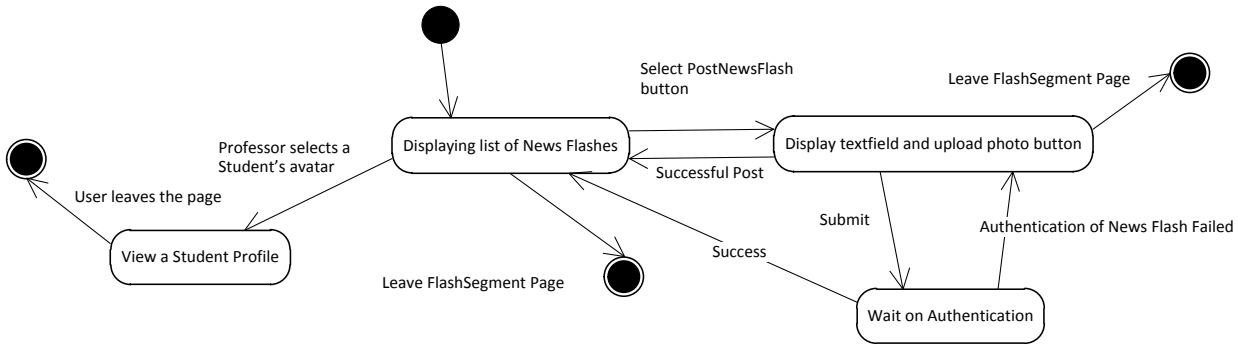


Figure 21 – State Chart FlashSegmentControl

State Machine for FlashFeed object (UC-M-06)

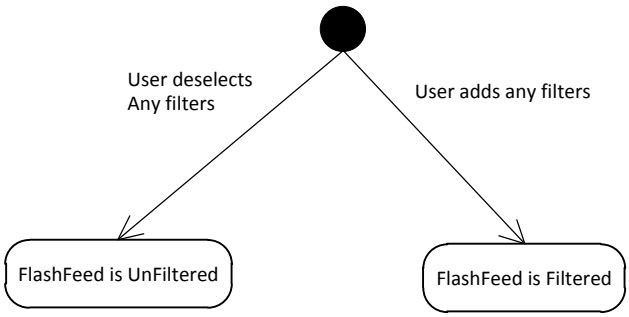


Figure 22 – State Chart Filter FlashFeed

State Machine for the System's Push Update feature (UC-S-1)

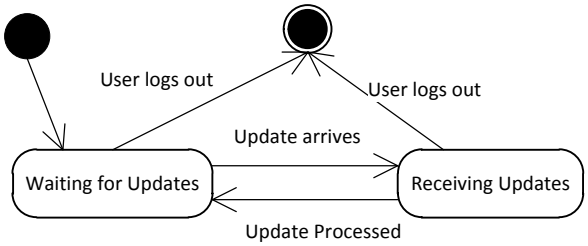


Figure 23 – State Chart Push Update

Sequence Diagram for use case LaunchCUBook and Login (UC-L-01, UC-L-02)

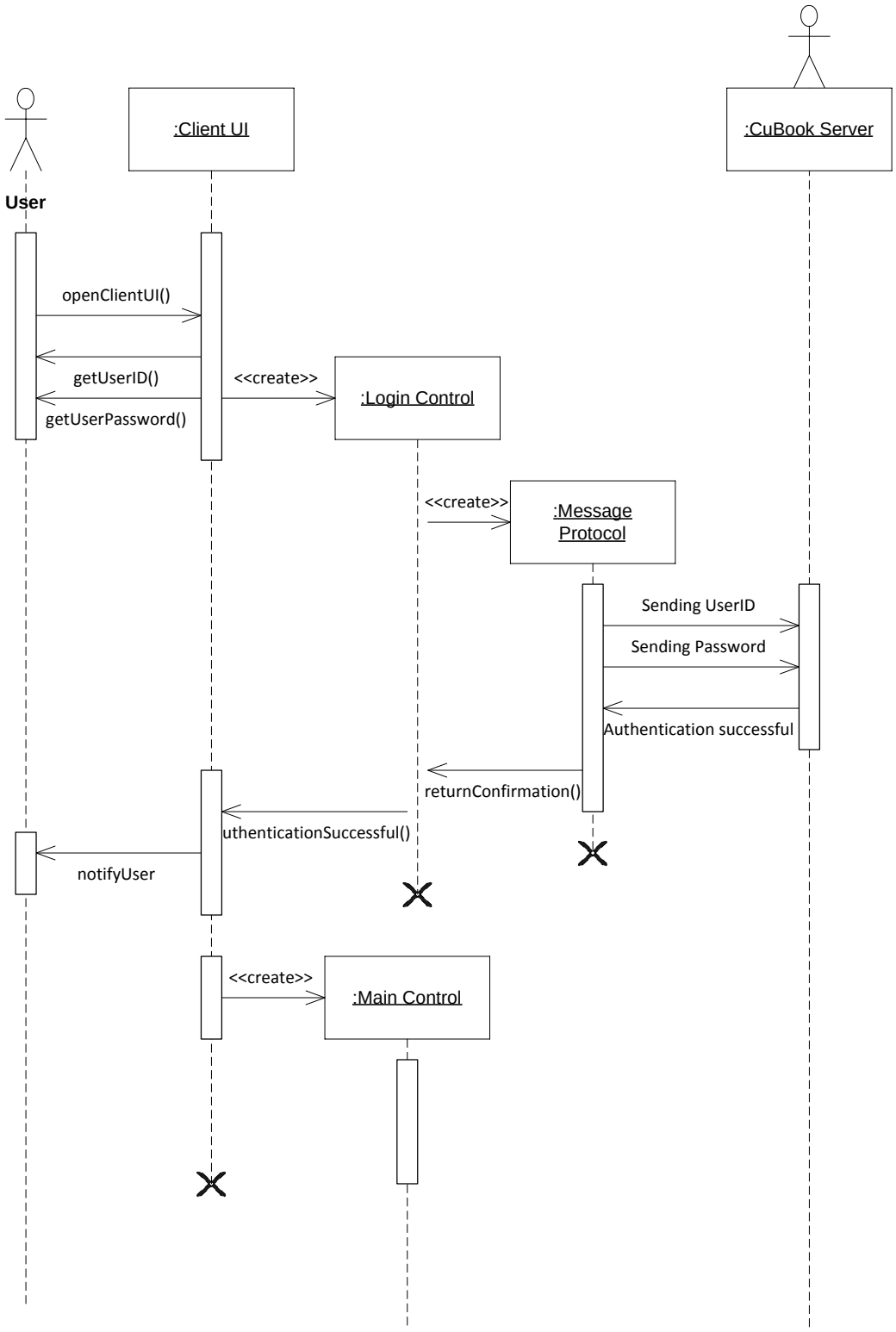


Figure 24 – Sequence Login Diagram

Sequence Diagram for use case LaunchCUBook and Login (UC-L-01, UC-L-02)
Demonstrates functionality when error occurs in authentication.

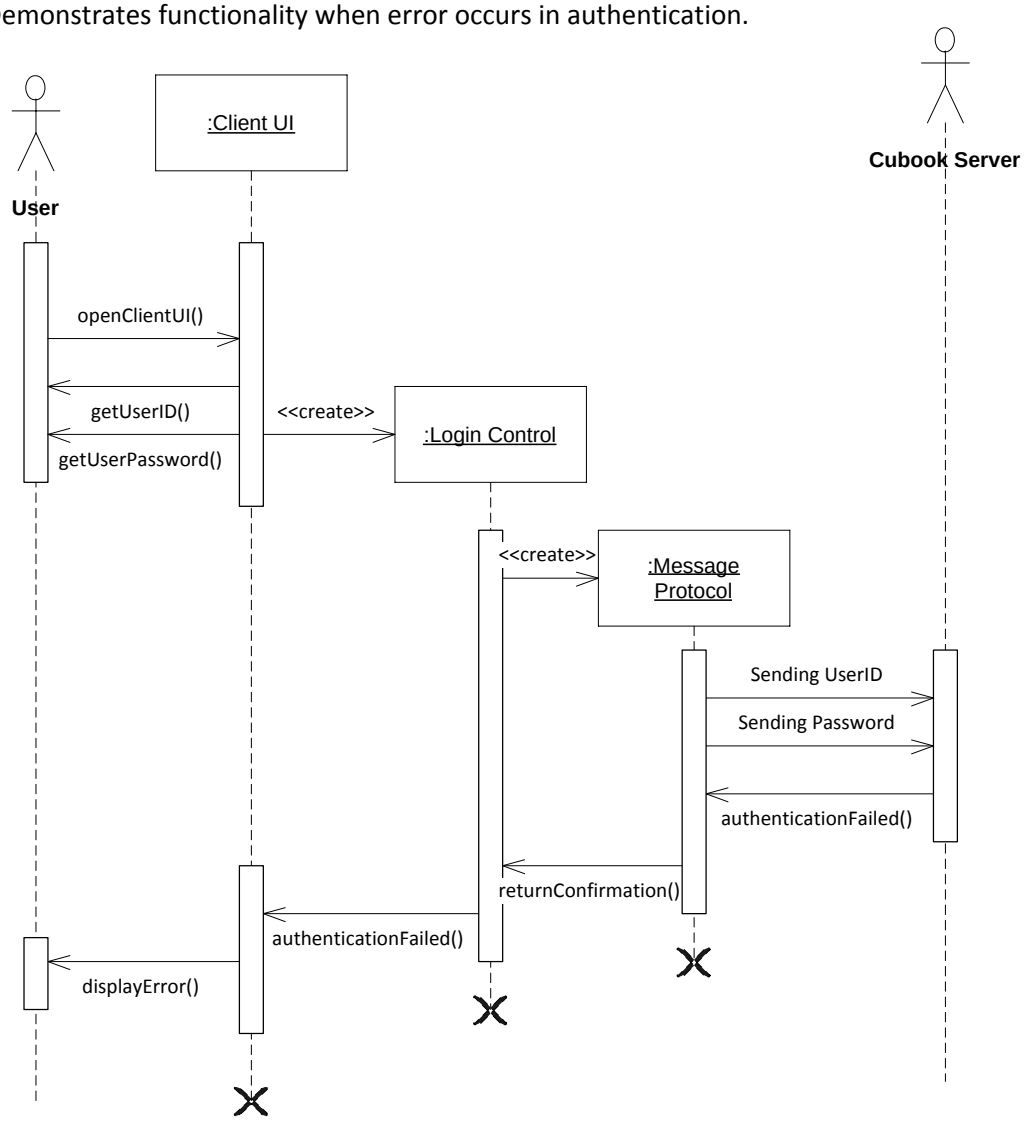


Figure 25 – Sequence Log In Error Diagram

Sequence Diagram for Viewing a flash segment page (UC-F-01)

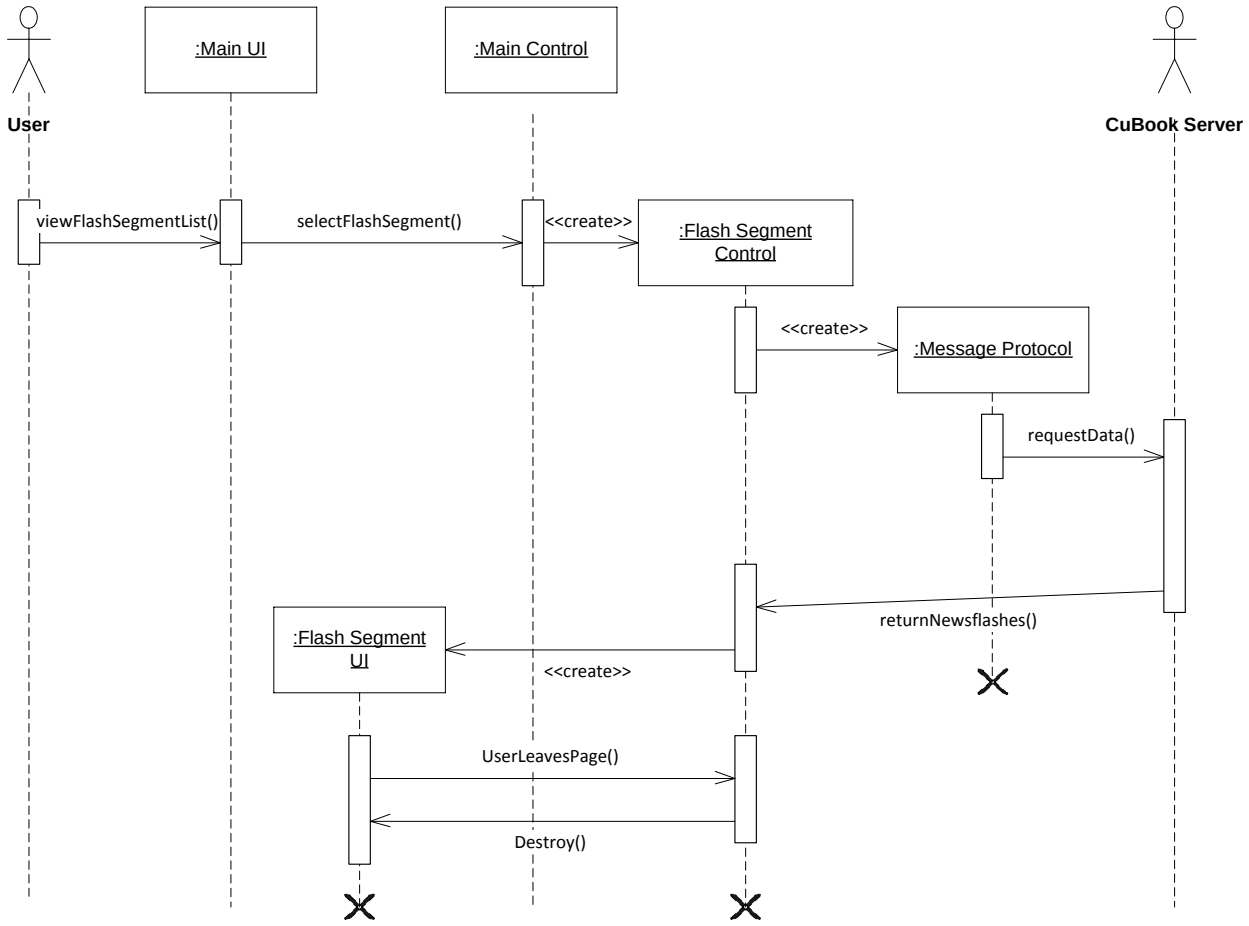


Figure 26 – Sequence Flash Segment Diagram

Sequence Diagram for Viewing a flash segment page in which an error occurs while Trying to post a flash segment (UC-F-01)

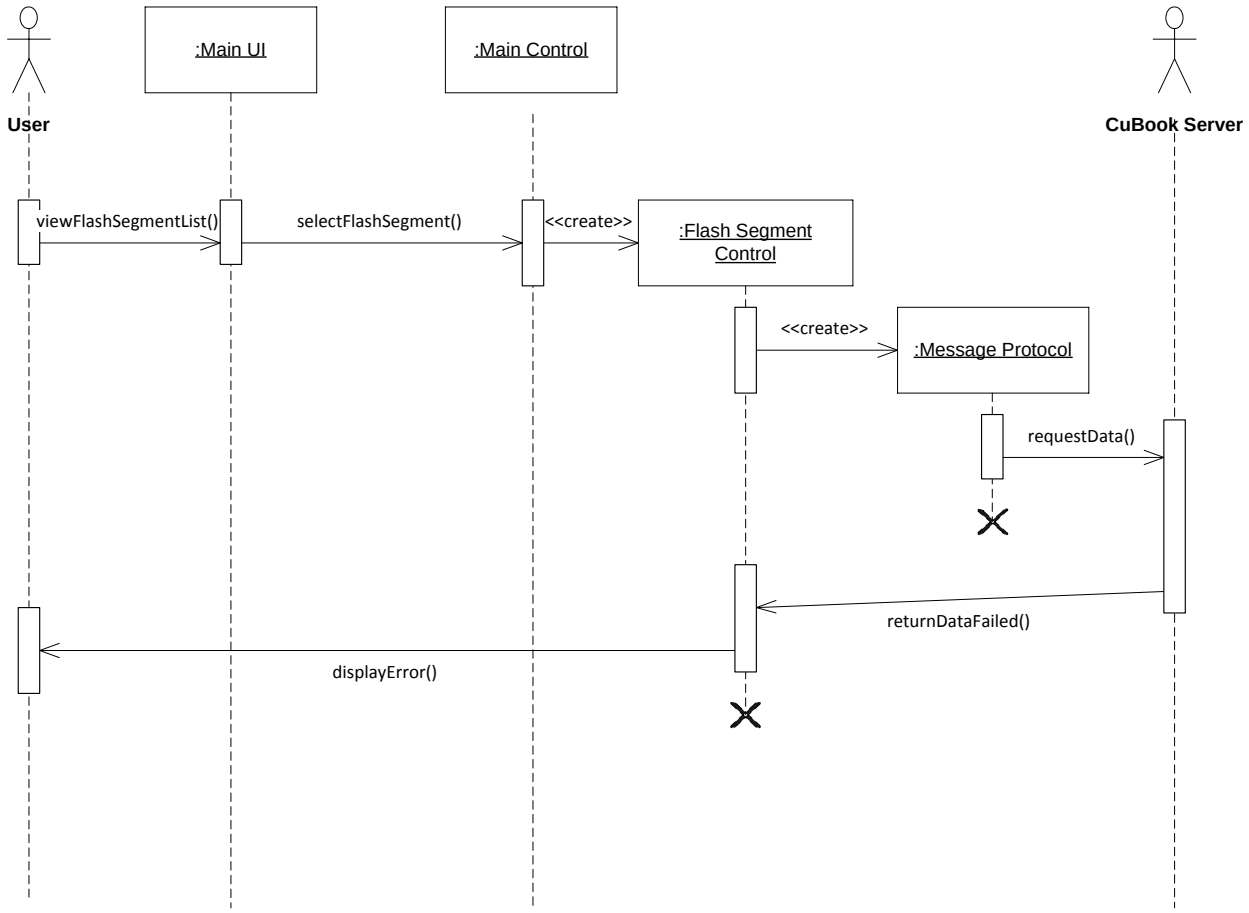


Figure 27 – Sequence Flash Segment Diagram with Errors

Sequence Diagram for Switching over to the flashboard page (UC-M-02)

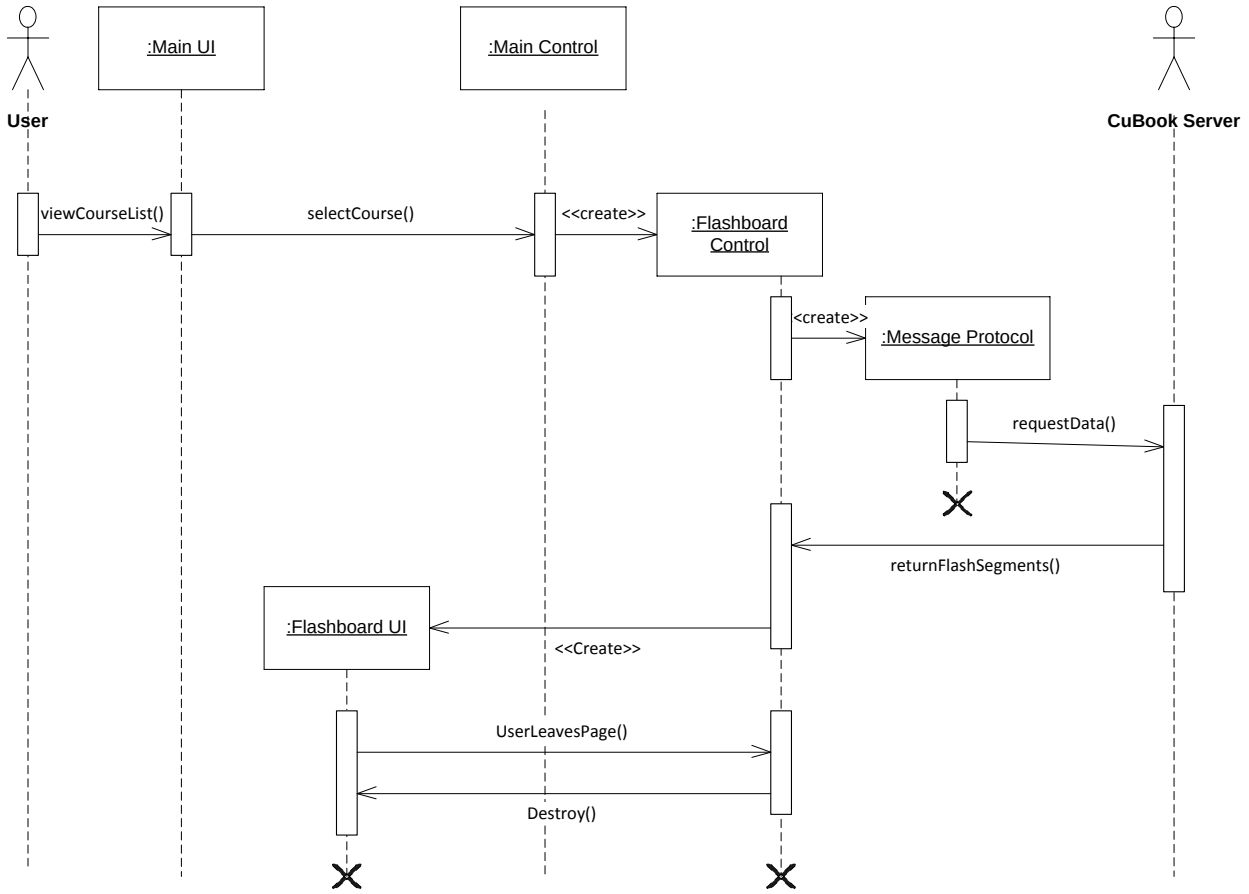


Figure 28 – Sequence flashboard diagram

Sequence Diagram for Attempting to upload a photo to CUBook (UC-U-01)

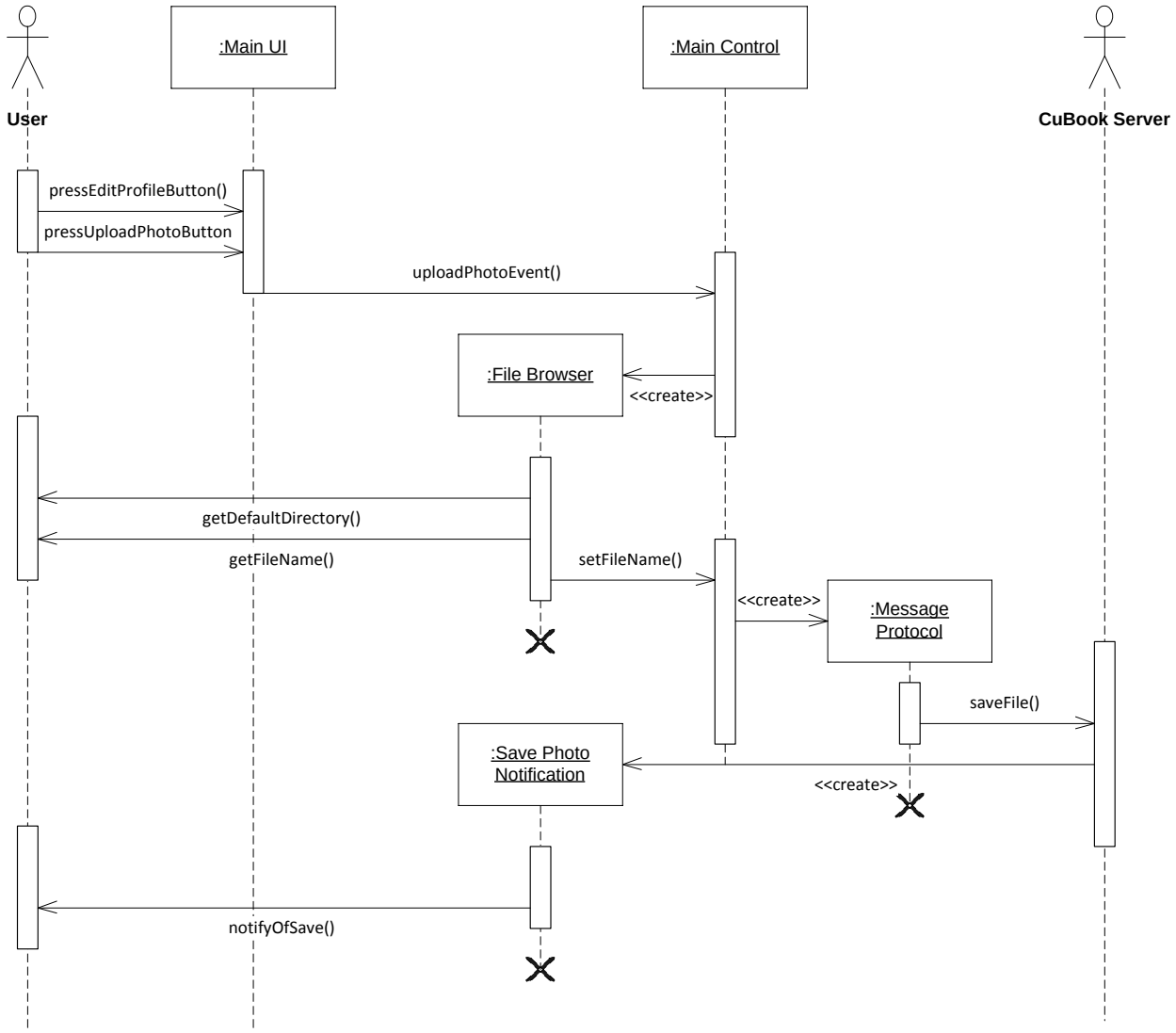


Figure 29 – Sequence Upload Picture Diagram

Sequence Diagram for Attempting to upload a photo to CUBook with errors (UC-U-01)

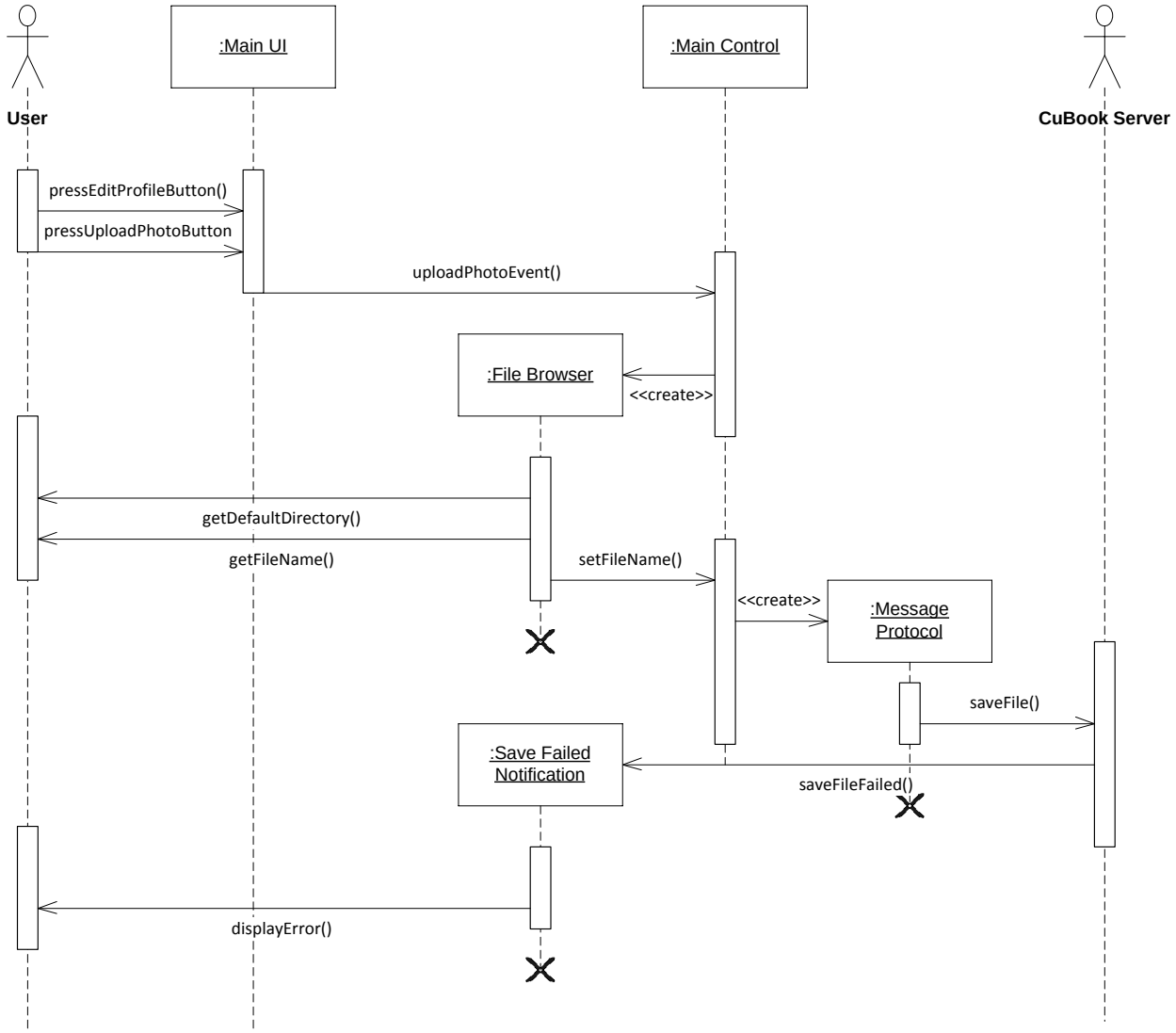


Figure 30 – Sequence UploadPicture Diagram with Errors

Sequence Diagram for Uploading a FlashSegment to CuBook (UC-F-02)

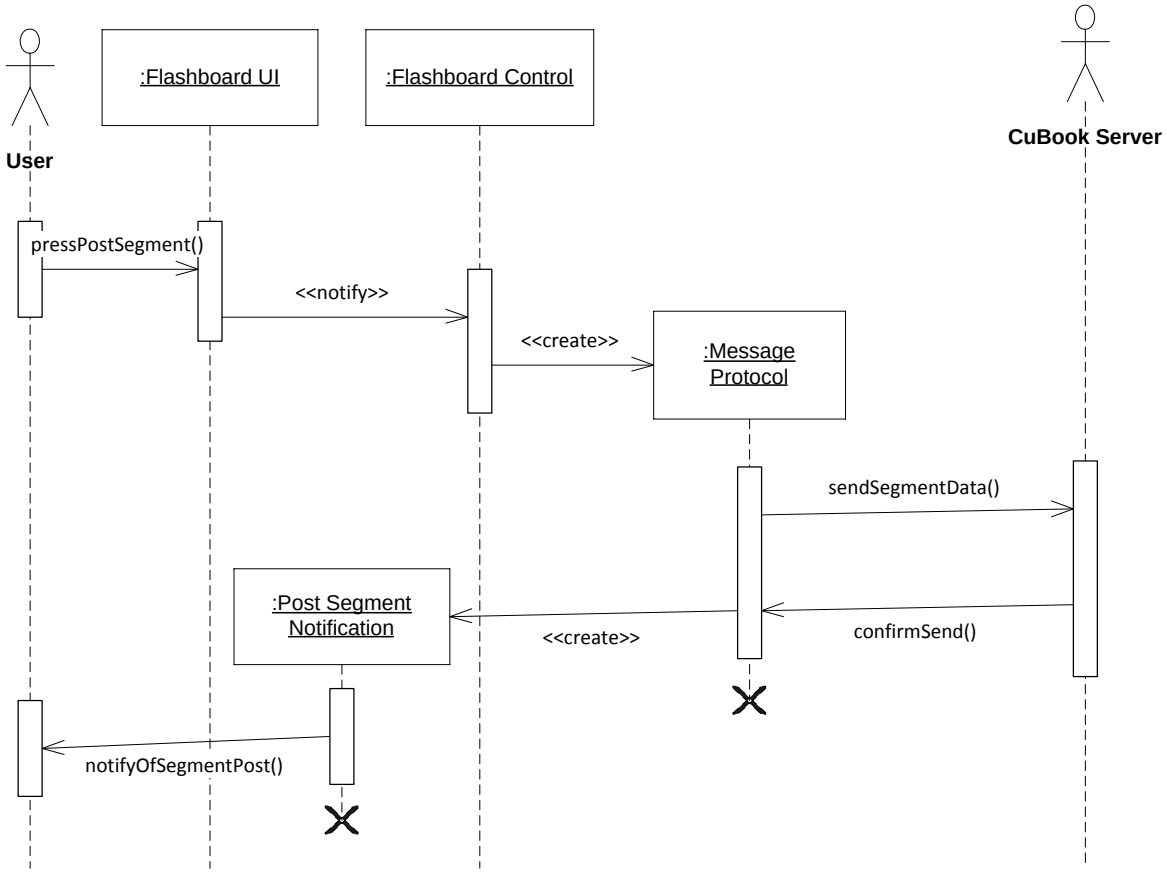


Figure 30b – Sequence Post-flashsegment Diagram

Sequence Diagram To upload a flashsegment to CUBook with errors (UC-F-2)

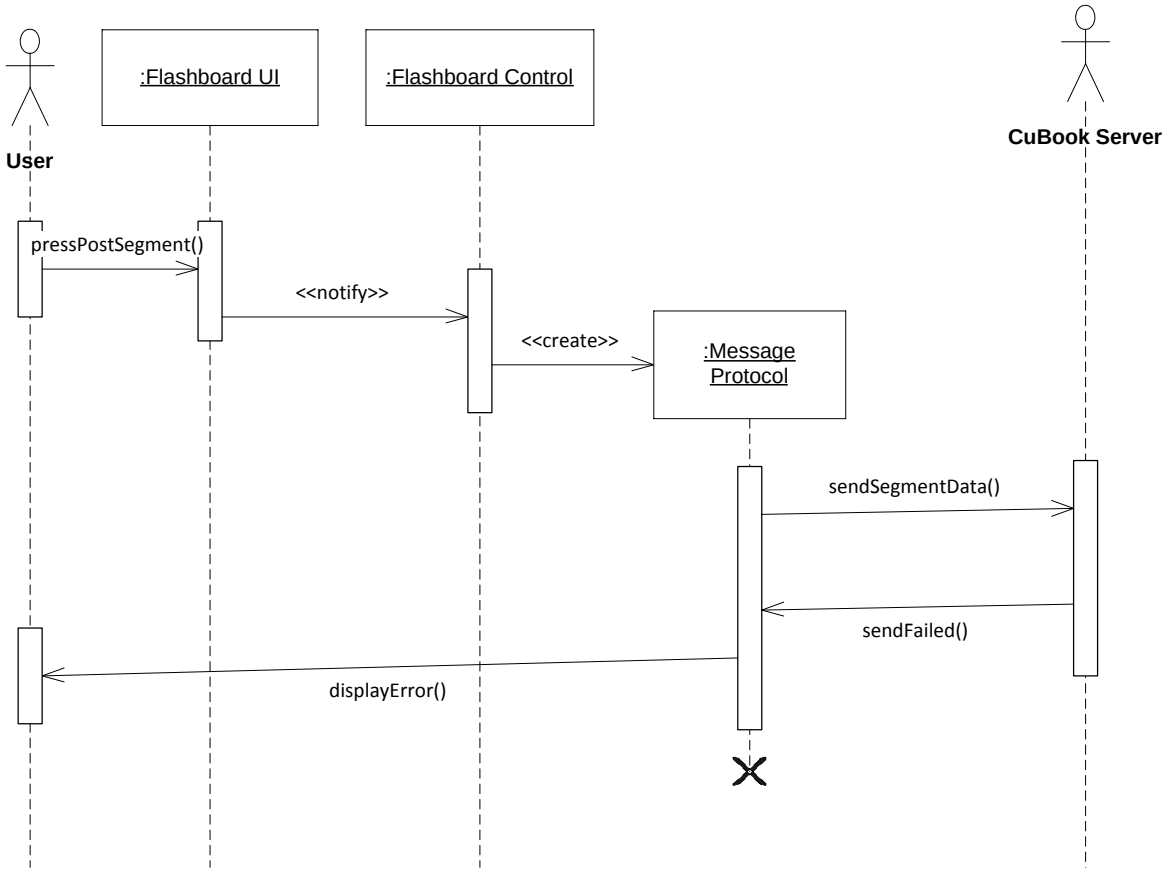


Figure 31 – Sequence Post Flashsegment diagram with errors

Sequence Diagram to upload a newsflash to CUBook (UC-F-5)

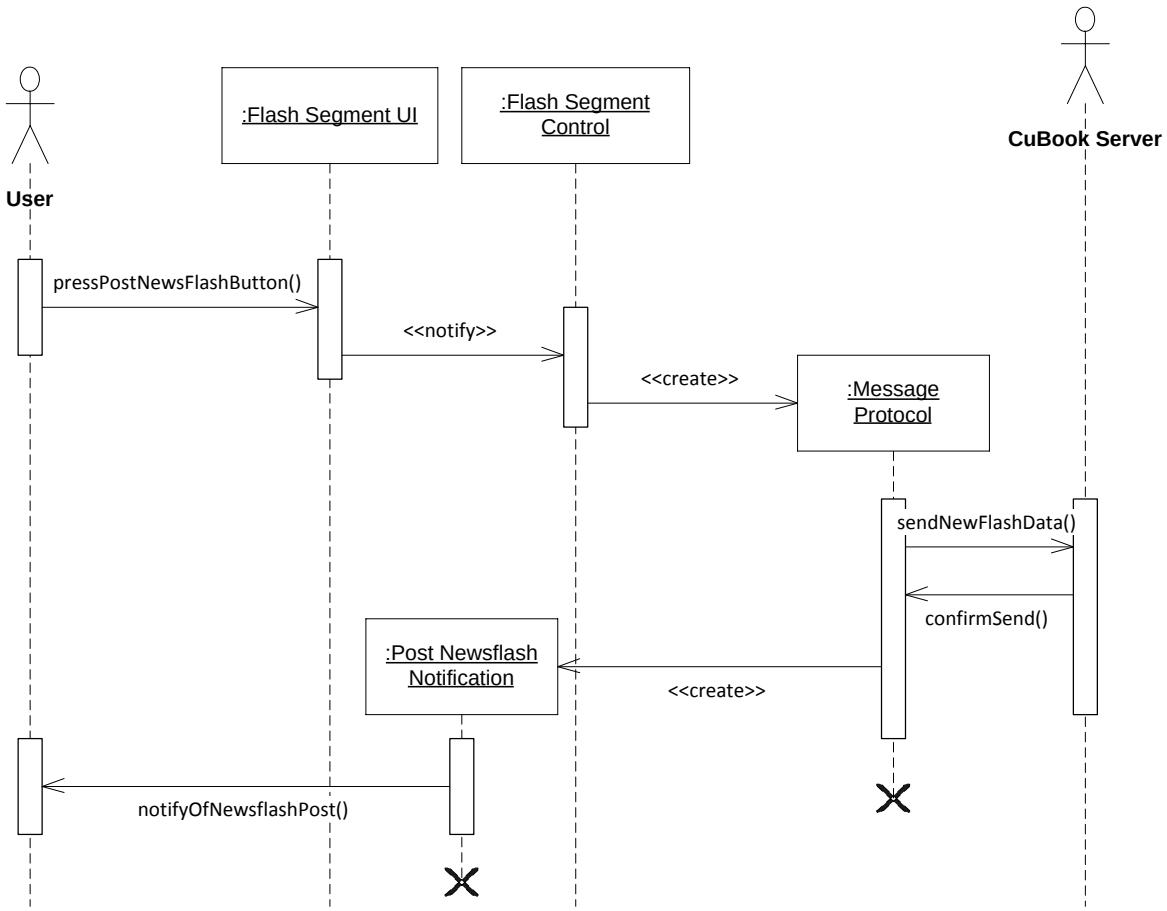


Figure 32 – Sequence Newslfash Diagram

Sequence Diagram To upload a NewsFlash to CUBook with errors (UC-F-5)

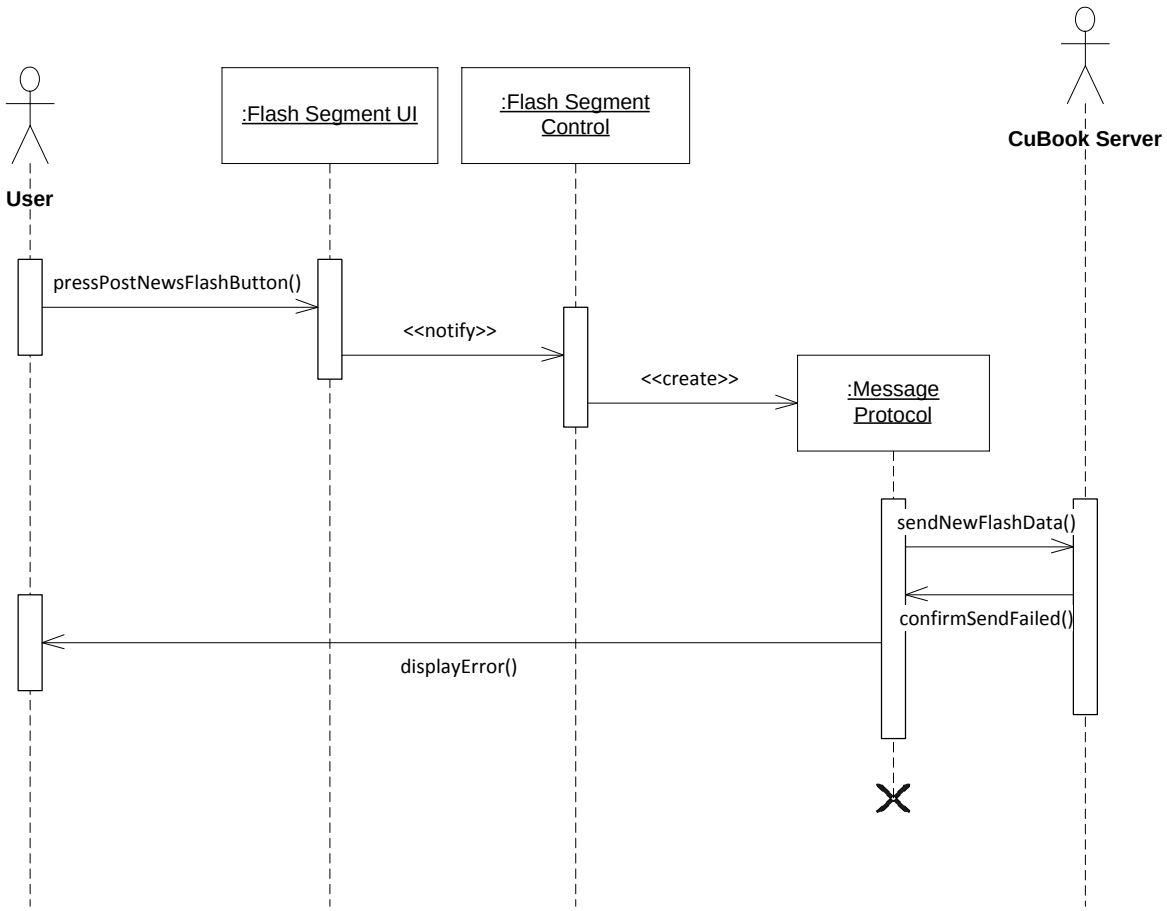


Figure 33 – Sequence NewsFlash Diagram with Errors

Sequence Diagram To filter a flashfeed (UC-M-6)

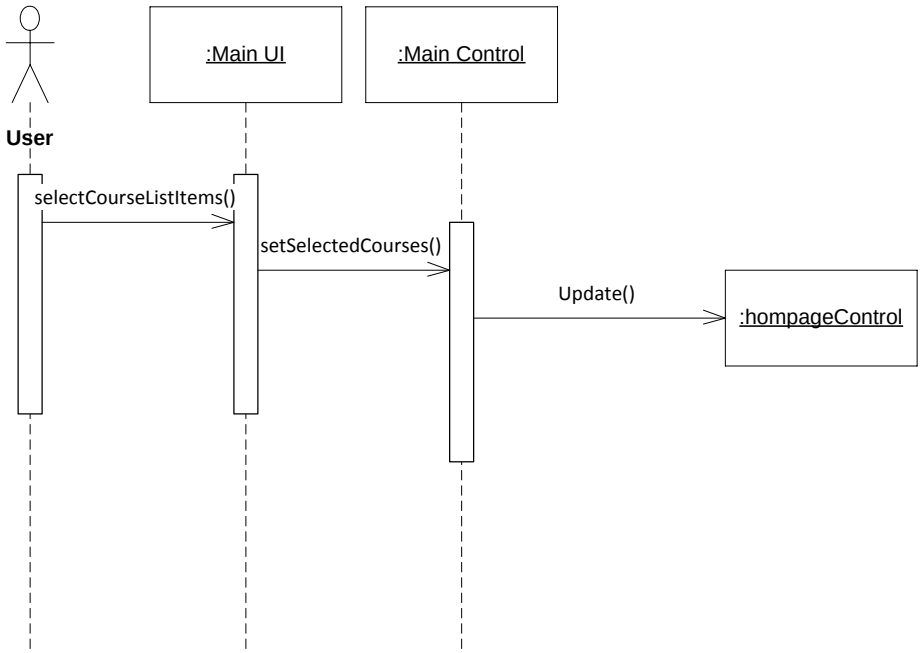


Figure 34 – Sequence Filter Flashfeed Diagram

Sequence Diagram To update an avatar (UC-M-05-1)

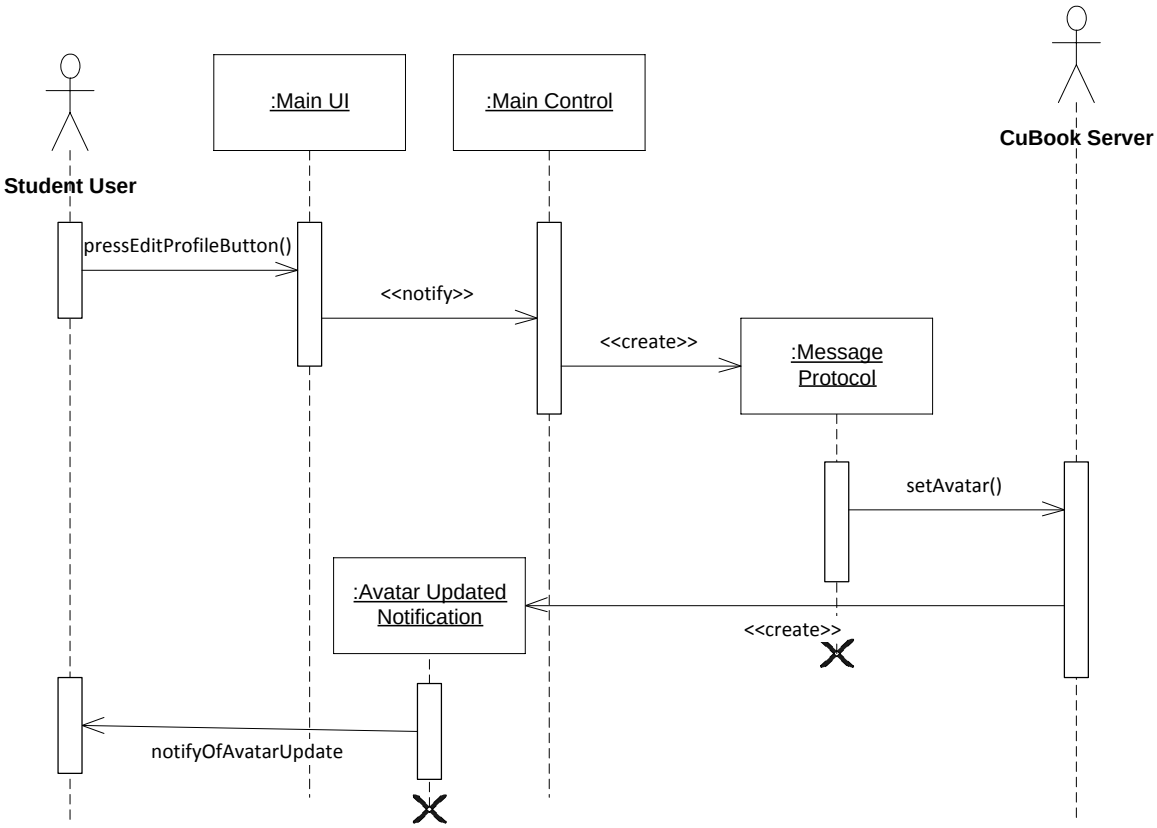


Figure 35 – Sequence UpdateAvatar Diagram

Sequence Diagram To update an avatar with error (UC-M-05-1)

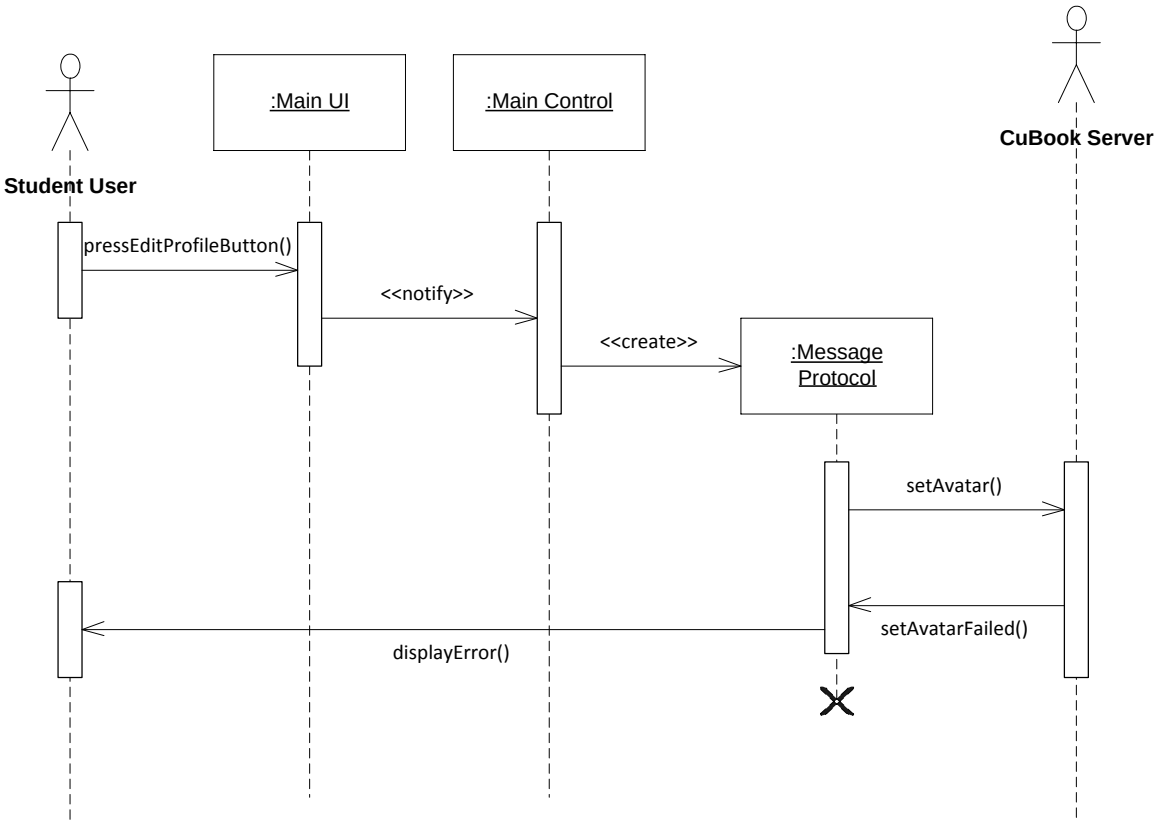


Figure 36 – Sequence UpdateAvatar Diagram with Errors

Sequence Diagram To delete a flash segment (UC-F-3)

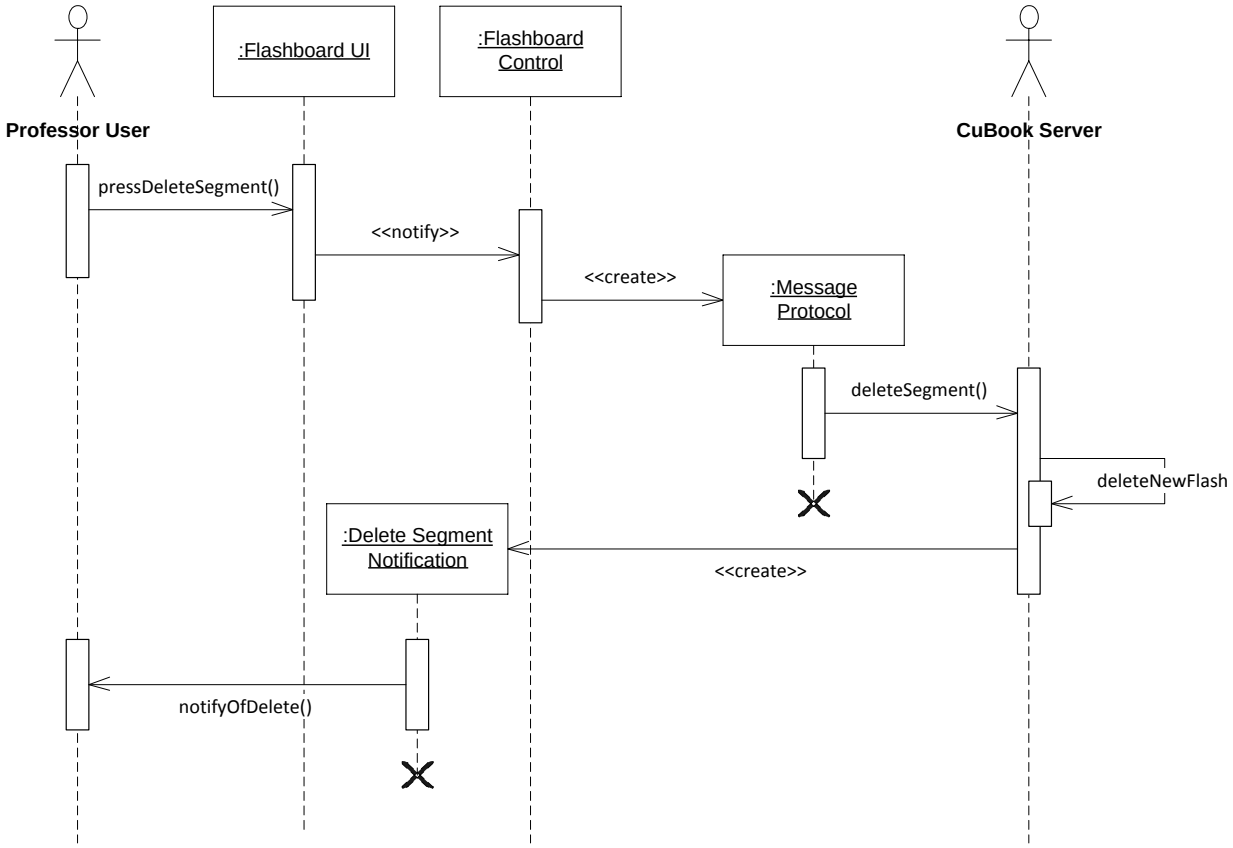


Figure 37 – Sequence DeleteFlashSegment Diagram

Sequence Diagram to delete a flashsegment with errors (UC-F-3)

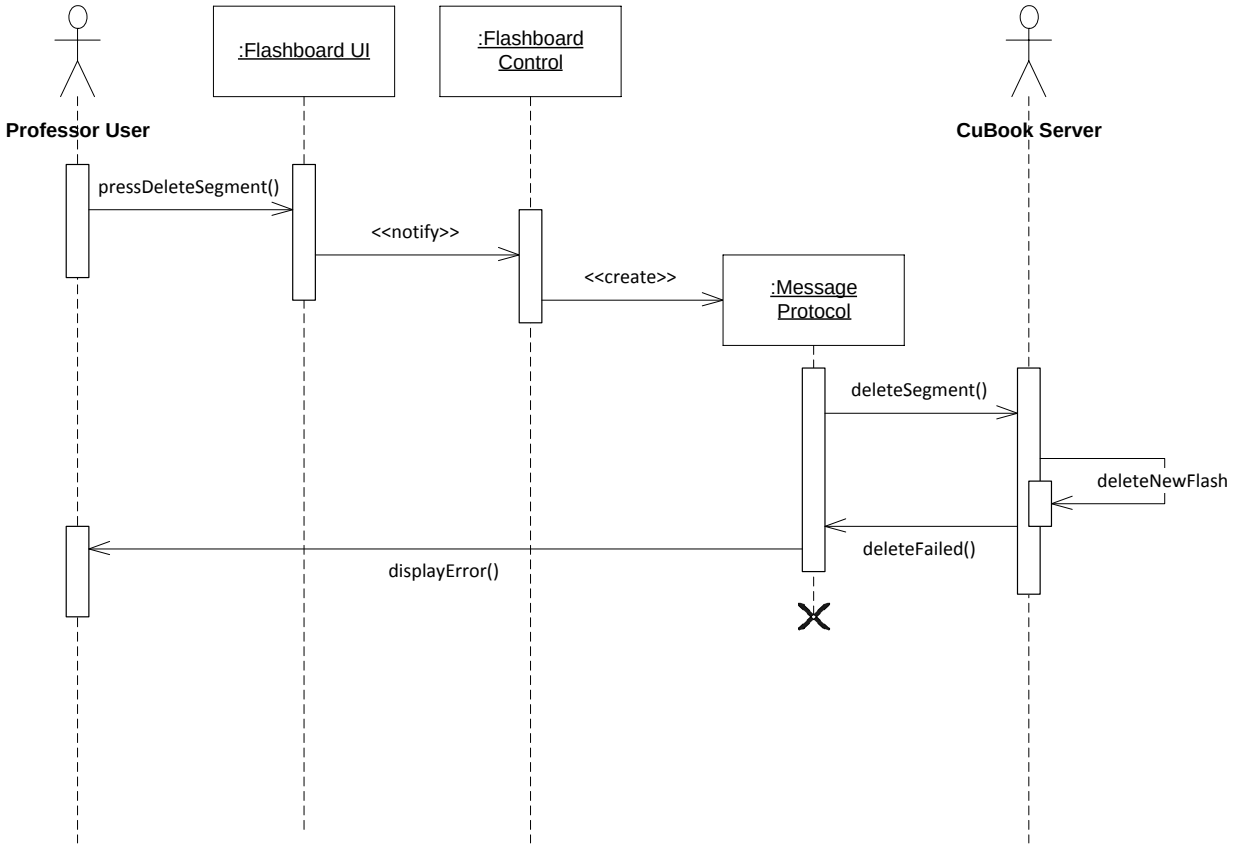


Figure 38 – Sequence DeleteFlashSegment Diagram with errors

Sequence Diagram To delete a newsflahs (UC-F-6)

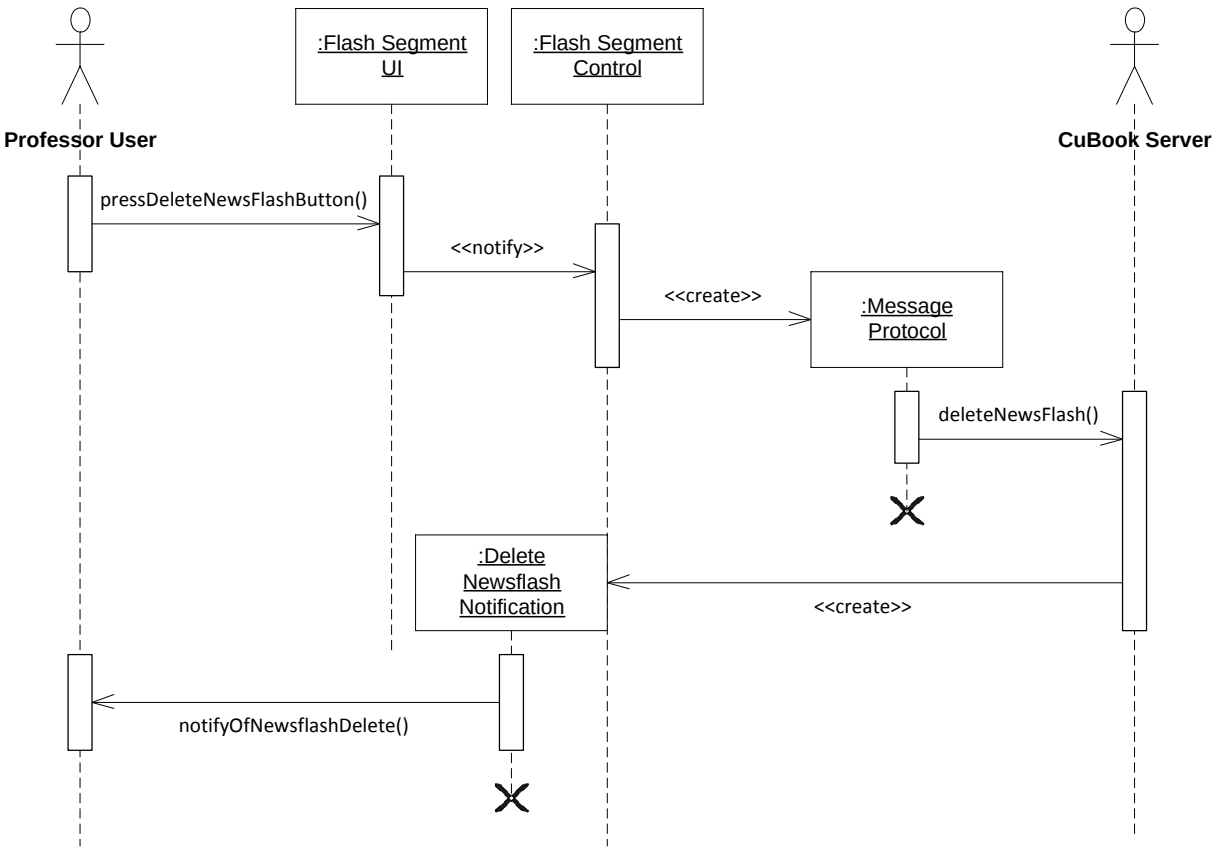


Figure 39 – Sequence Delete NewsFlash Diagram

Sequence diagram to delete a newsflash with errors (UC-F-6)

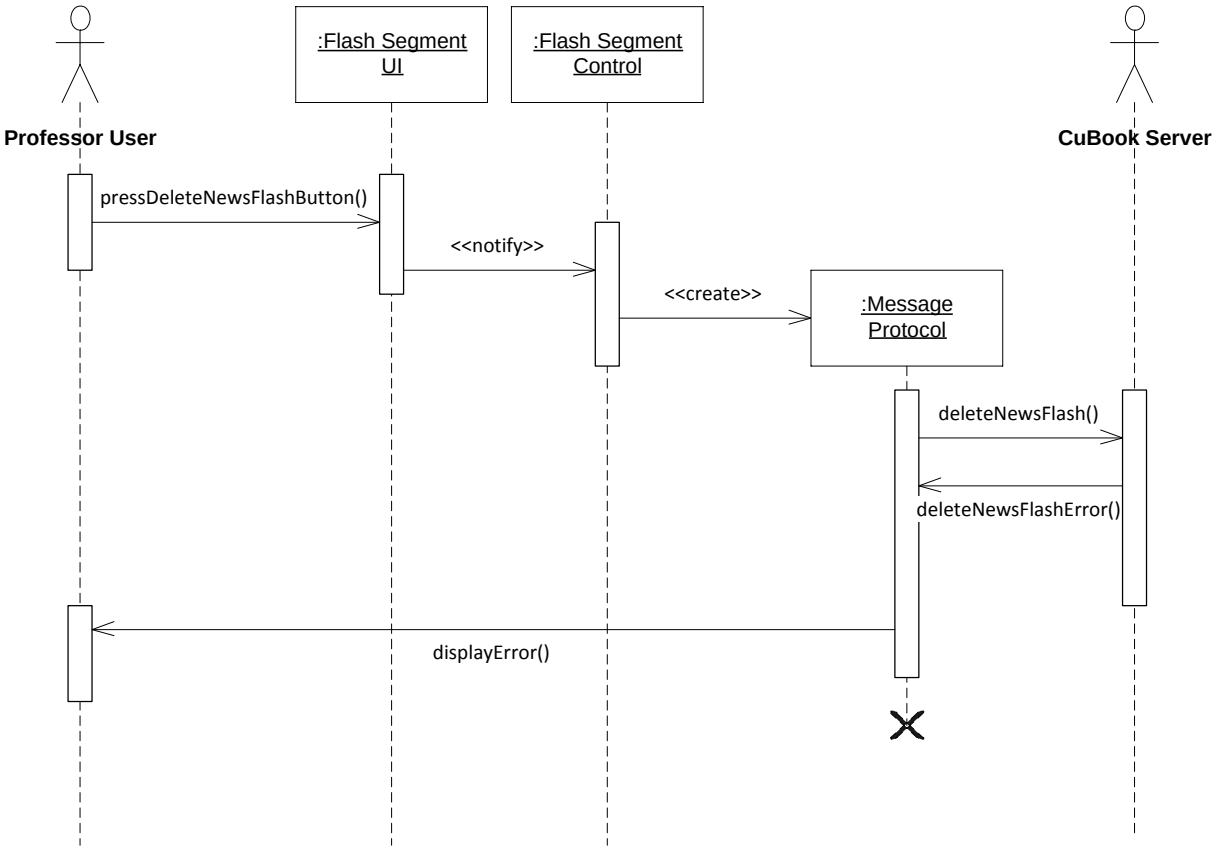
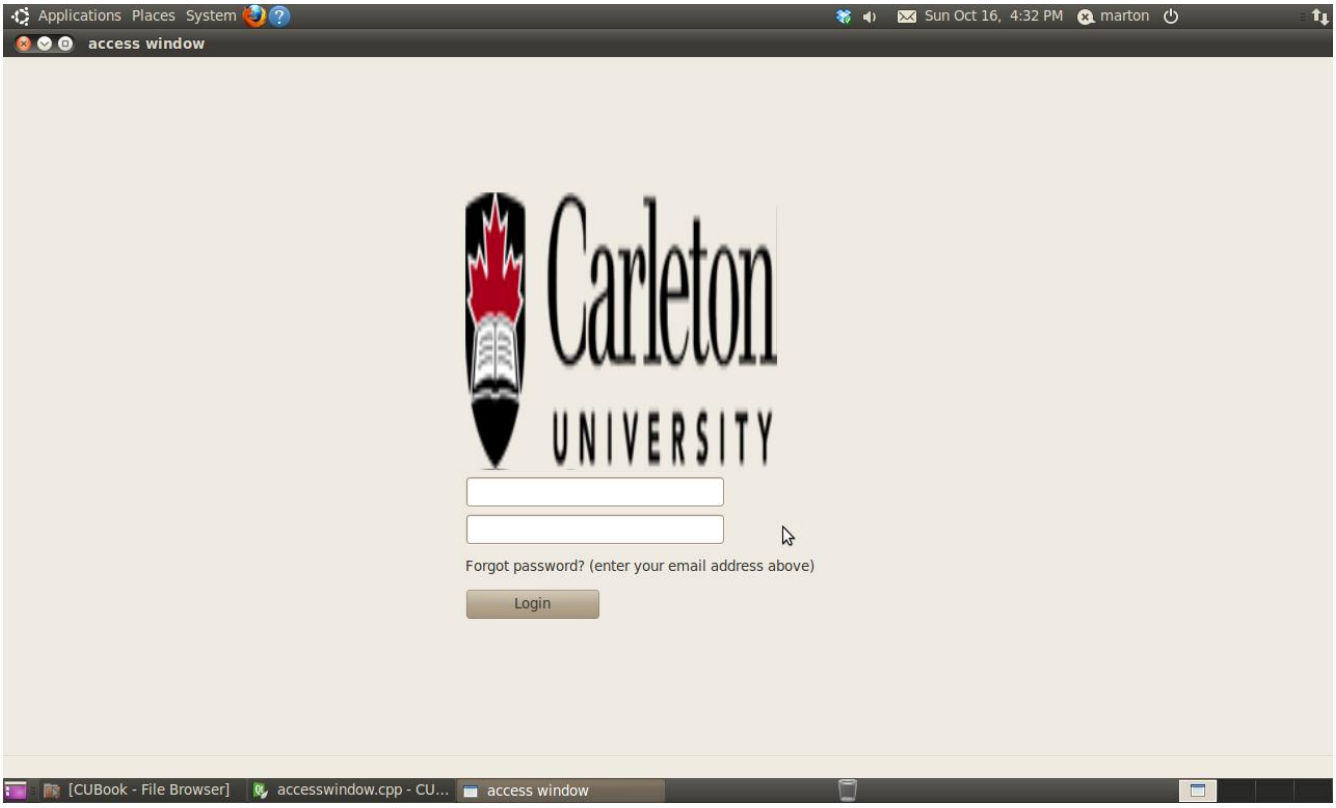
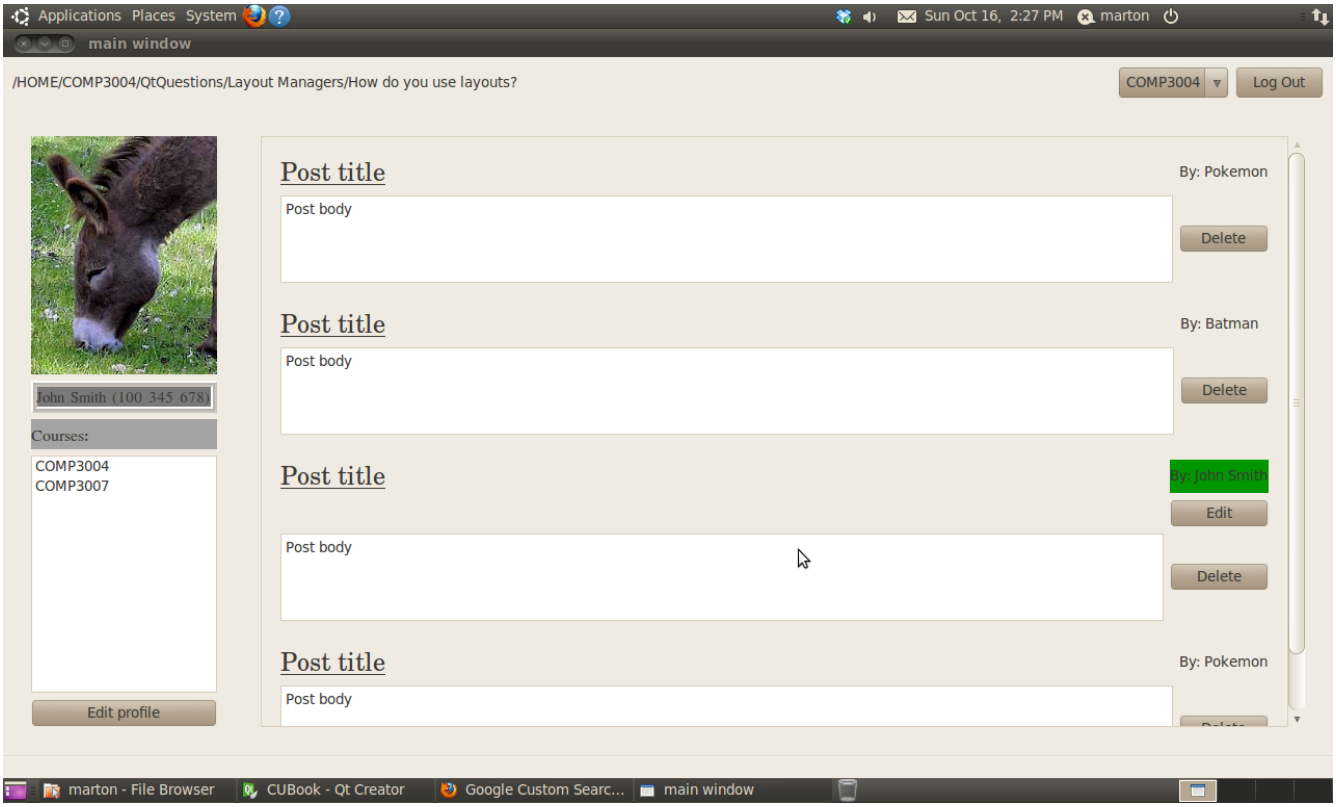


Figure 40 – Sequence DeleteNewsflash diagram with errors

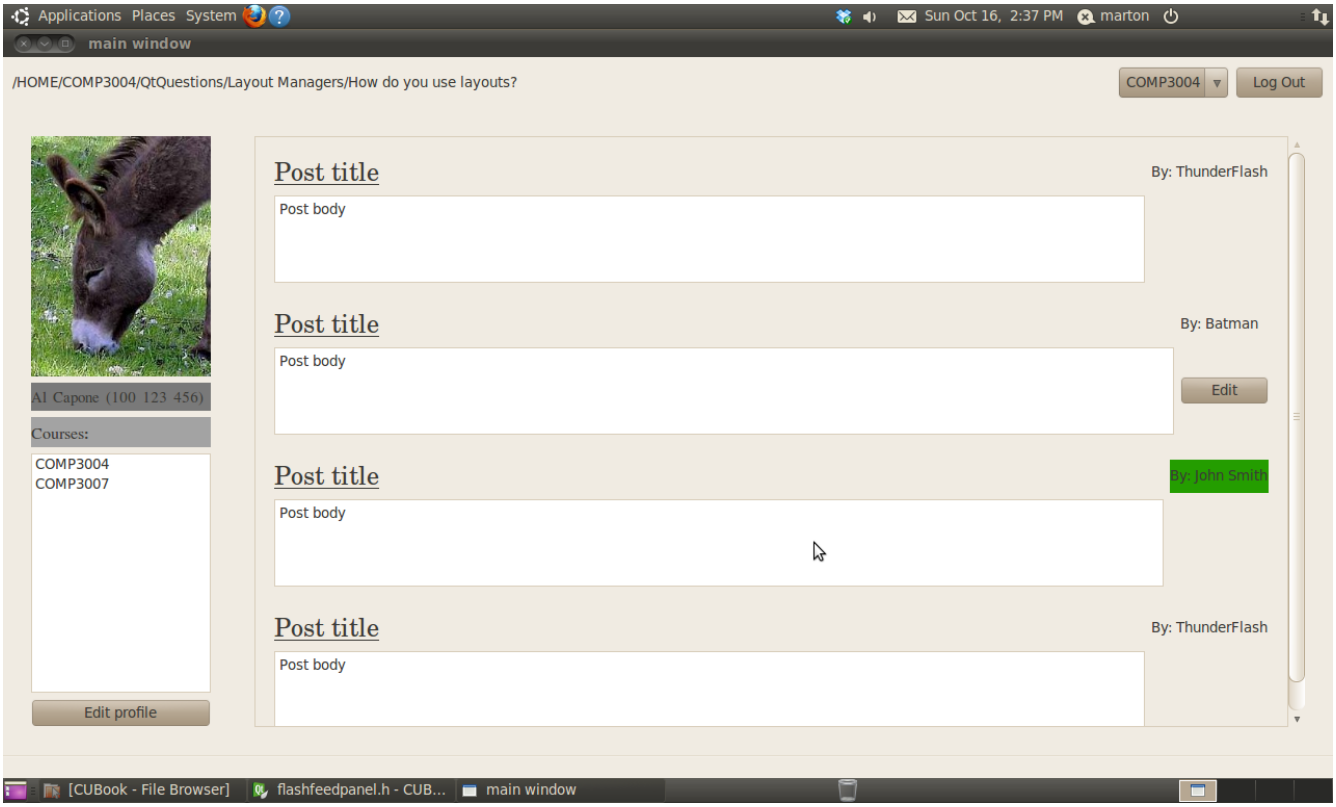
2.4.4 User Interface



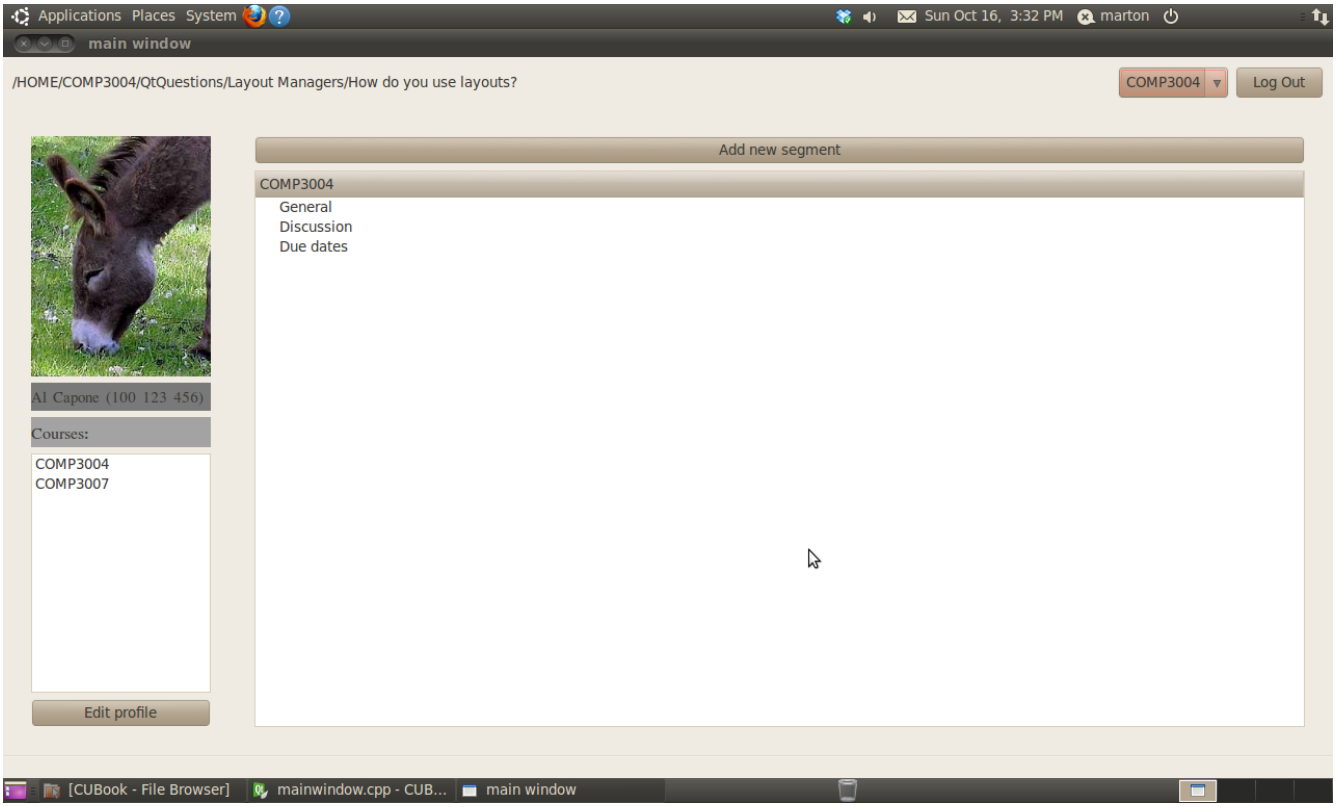
Login page – this is the page that is first displayed when the program is run. Has basic functionality to take in a user name and password, and then submit it to CUBook for authentication.



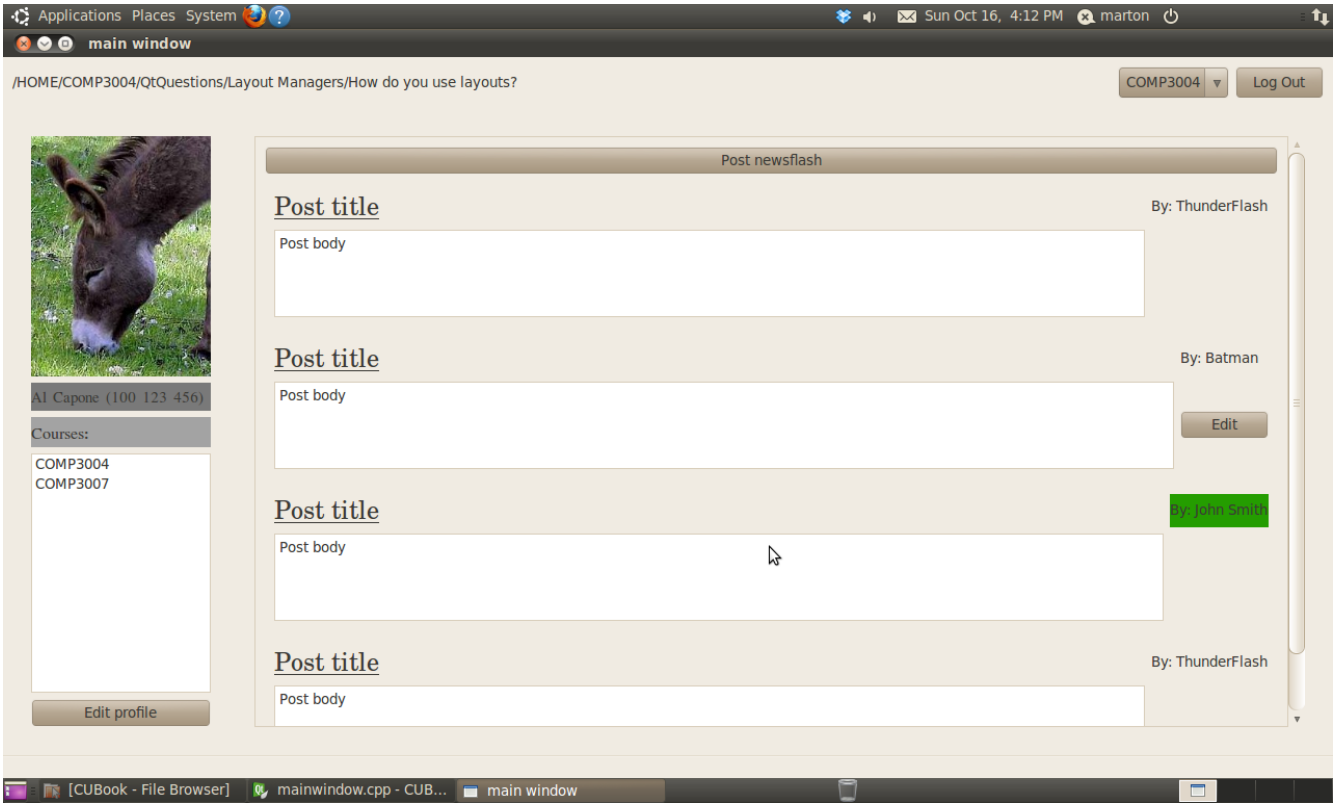
Flashfeed (professor) –this is the home page for a professor after they log in. It shows their flash feed, list of courses to filter on the left, drop down menu to select a course's flash board at the top, log out button, profile picture, real name, edit profile button, and professor information.



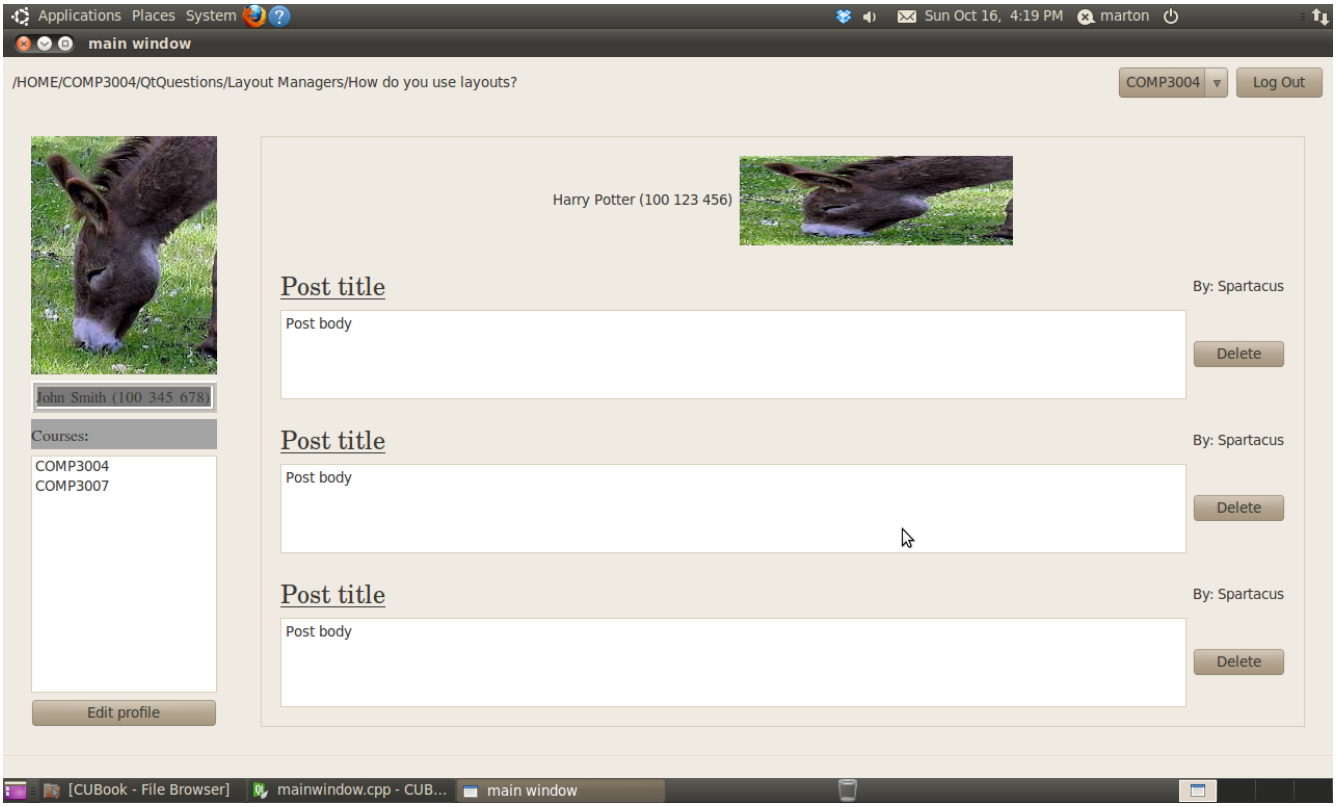
Flashfeed (student) – this is the home page for a student after they log in. Like the professor page, it shows their flash feed, list of courses to filter on the left, drop down menu to select a course's flash board at the top, log out button, profile picture, avatar, edit profile button, and student information.



Flashboard – When a user visits a flashboard page, the page panel changes to a list of flash segments.



Flashsegment - When a user visits a Flash Segment page, it shows a list of News Flashes. If a News Flash is made by a professor, it will display their name with a highlighted green background.



Studentinfo – When a Professor views a student's profile, he will see this page. This page shows the student's recent list of posts in their course, their real name, and their student id number.

flashfeed (main) (professor) -Flashfeed (from all courses) for professors

flashfeed (main) (student)-Flashfeed (from all courses) for students

*Note: the above two screens are also meant to highlight the distinction *between the side panels of students and profs

flashboard (student) - Shows all the flash segments within a course

Flashsegment (student) - the flashfeed for individual flashsegments

Studentinfo (only for profs) - the student info that a prof can view

includes a feed of all of the students postings on all boards

Loginpage

3. Glossary

Flashfeed – The page listing newsflashes in chronological order. Newsflashes are filtered according to class, using the class listings on the left of the page. When a course is displayed in a newsfeed, all of the newsflashes on its flashsegments are displayed chronologically. Synonyms: home page

Homepage – Synonym for flashfeed.

Newsflash – A single post in a flash segment , which are inside flashboards.

Flashboard – A listing of categories (flash segments) into which newsflashes are organized. There is one flashboard for each course and flash segments can be added here.

Flash Segment – A category into which newsflashes are organized. Newsflashes can be added here and no nested flash segments can be added. Flash segments are similar to a flashfeed in that the posts are listed chronologically.

InternalFlashFeed - An Internal Flash Feed is the flash feed that you see as a user when you first log in on the home page. It shows a list of posts relevant to you, in any courses that you are registered in, in chonological order.

ExternalFlashFeed - External Flash Feeds are only viewable by Professors. When a professor views a student's profile, it will display their External Flash Feed, which is a chronological list of posts made by that student in the professor's course, since the beginning of the term. See also: student info page

Student info page – This page shows the student's picture and real name, and shows the external flash feed below that. See also: external flash feed.

Real name – Each CUBook user has a real name associated with them, but this is not displayed in the case of students, unless their student info page is viewed by a professor. Professors' real name is always displayed and they do not have an avatar. See also: avatar

Avatar – A nickname that is visible for student posts. A student's real name is visible when the prof navigates to their student info page.