

Software Design Description

Spartan Course Analysis & Matching

Version 2.0 - 5/16/17

San José State University Charles W. Davidson College of Engineering

**Mark Casapao
Christian Timbol
Preyrna Yadav
Hiep Nguyen**

Revisions

Version	Primary Authors	Description of Version	Date Completed
v1.0 draft	Mark Casapao Christian Timbol Preyrna Yadav Hiep Nguyen	An outline or foundation of the document	4/1/17
v1.0 final	Mark Casapao Christian Timbol Preyrna Yadav Hiep Nguyen	A finalized and confirmed iteration of the document	4/9/17
v2.0 final	Mark Casapao Christian Timbol Preyrna Yadav Hiep Nguyen	A refined iteration of the document. All docs are now interconnected.	5/16/17

Table of Contents

1.	INTRODUCTION.....	4
1.1.	Design Overview.....	4
1.2.	Requirements Traceability Matrix.....	4
2.	SYSTEM ARCHITECTURAL DESIGN.....	5
2.1.	Chosen Design Architecture (MVC Model).....	6
2.2.	Discussion of Alternative Designs (Class, Data Flow).....	8
2.3.	System Interface Description (Context Model).....	9
3.	DETAILED DESCRIPTION OF COMPONENTS (Sequence Diagram).....	11
3.1.	Login.....	11
3.2.	Profile.....	12
3.3.	Current Schedule.....	13
3.4.	Social.....	14
4.	USER INTERFACE DESIGN.....	15
4.1.	Description of User Interface.....	15
4.1.1.	Screen Display (UI Transition)	15
4.1.2.	Objects and Actions (Process Model, Activity Diagram, State Transition).....	17
5.	ADDITIONAL MATERIAL.....	21
5.1.	Architectural View of Deployment.....	21

1.0 Introduction

This section will provide an overlaying description of the product's design, and will provide a clear idea on how the system intends to be implemented.

1.1 Design Overview

The purpose of this document is to inform everybody who is involved with the project a general idea on the design of the product. The following models will be included: Context Model, MVC Architecture, Process Model, Activity Diagrams, State Transition Diagram, Sequence Diagram, Data Flow Diagram, Class Diagram, Architectural View of Deployment, and UI Transition Diagram. Each design model will provide a different aspect or point of view on the interpretation of the product, and its many different fields.

1.2 Requirements Traceability Matrix

Functional Requirements	Design Specifications	Test Cases	Implementation
2.2.1.1 Registration - User must be able to register SCAM account.	1.1 Users register via login page	3.1.1 Registration is tested via BBT - working	1.1 Executed as the default screen
2.2.1.2 Logging In - User must be able to log in using the newly generated account.	1.2 User is capable of logging in by entering fields via LoginPage	3.1.2 Login is tested via BBT and unit tests - working	1.2 Executed as the default screen
2.2.2.1 Edit Profile - Fill in blanks with necessary information.	2.1 ProfilePage provides an option to edit profiles	3.2.1 Profile is tested via BBT and unit tests - working	2.1 Executed at the Profile Page
2.2.2.2 Delete Info - Privacy options	2.2 ProfilePage's options provides a feature to hide all information necessary	3.2.2 Hiding is tested via BBT and unit tests - working	2.2 Executed at the Profile Page
2.2.2.3 Delete	2.3 Users can delete	3.2.3 Deletion is	2.3 Executed at the

Account - Option to delete account	accounts via ProfilePage	tested via unit tests - working	Profile Page
2.2.2.4 View Profile - User must be able to view their own profile.	2.4 Users are able to see their profile via ProfilePage	3.2.4 Viewing is tested via BBT - working	2.4 Executed at the Profile Page
2.2.3.1 View Schedule - Visual schedule	3.1 A schedule is displayed through SchedulePage	3.3.1 Viewing is tested via BBT - working	3.1 Executed in the Wall Page
2.2.3.2 Edit Schedule - Add/drop/modify class	3.2 SchedulePage provides options to edit class schedules	3.3.2 Editing is tested via unit tests and BBT - working	3.2 Executed in the Wall Page
2.2.3.3 Feedback - User must be able to contribute information.	3.3 Users are able to rate and review classes	3.3.3 Feedback is tested via unit tests - working	3.3 Executed in the Wall Page
2.2.4.1 View Inbox - access inbox page from the menu	4.1 Inbox is provided in SocialPage	3.4.1 Inbox is tested via unit tests and BBT - working	4.1 Executed using the menu in the corner of the screen
2.2.4.2 Send/Delete Messages - send their own messages, delete old messages.	4.2 Messages are sent back and forth between inboxes	3.4.2 Msg is tested via unit tests - working	4.2 Executed using the menu in the corner of the screen
2.2.4.3 Search Users - find users via search bar.	4.3 A search bar is provided within the SocialPage	3.4.3 Searching is tested via BBT - working	4.3 Executed using the menu in the corner of the screen
2.2.4.4 Add User to Friends - add user to friends	4.4 Every user has a friends list paired with him/her	3.4.4 Adding is tested via BBT and unit tests - working	4.4 Executed using the menu in the corner of the screen
2.2.4.5 View User Profile/Schedule -view contacts' profile and schedules	4.5 Users can view the profiles of other users they search up in the search bar	3.4.5 Viewing is tested via BBT - working	4.5 Executed using the menu in the corner of the screen
2.2.4.6 View Notifications - view any recent events around the user's network	4.6 Users can view their notifications via the notifications list	3.4.6 Notifications were tested via BBT - working	4.6 Executed using the menu in the corner of the screen

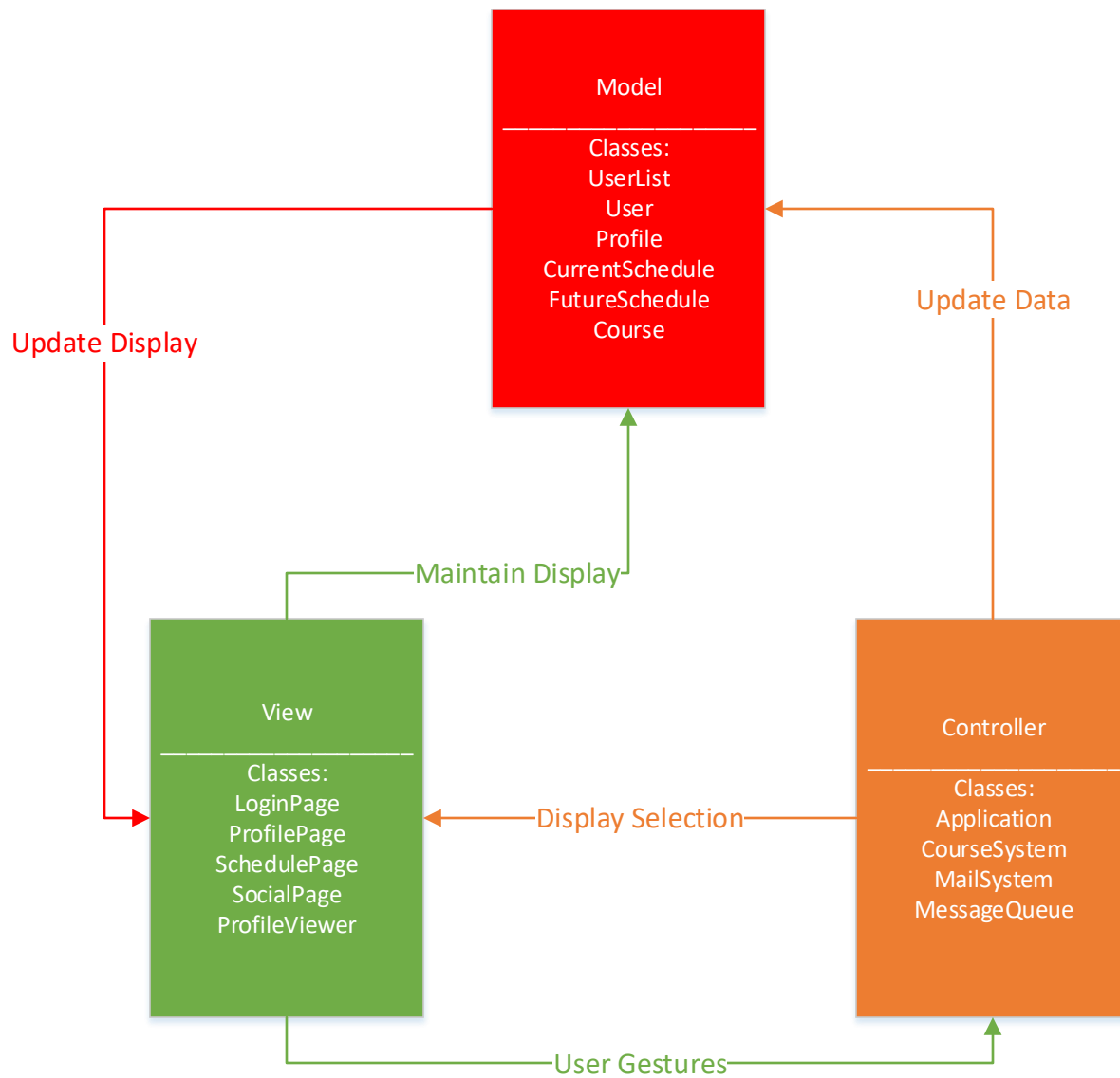
2.0 System Architectural Design

This section will follow up on the general portion of the design, which will then be put into components that will be described in later detail down the document. It opens up with the MVC Pattern, and builds up on that with a class diagram and a class diagram followed by a data flow model. Then, the context model will sum up the interaction of the system with other systems.

2.1 Chosen Design Architecture

The following model shows the MVC pattern that SCAM will take the form of. Each of the three components of MVC, model, view, and controller holds a list of classes that are generally characterized by that corresponding role of the whole architecture design.

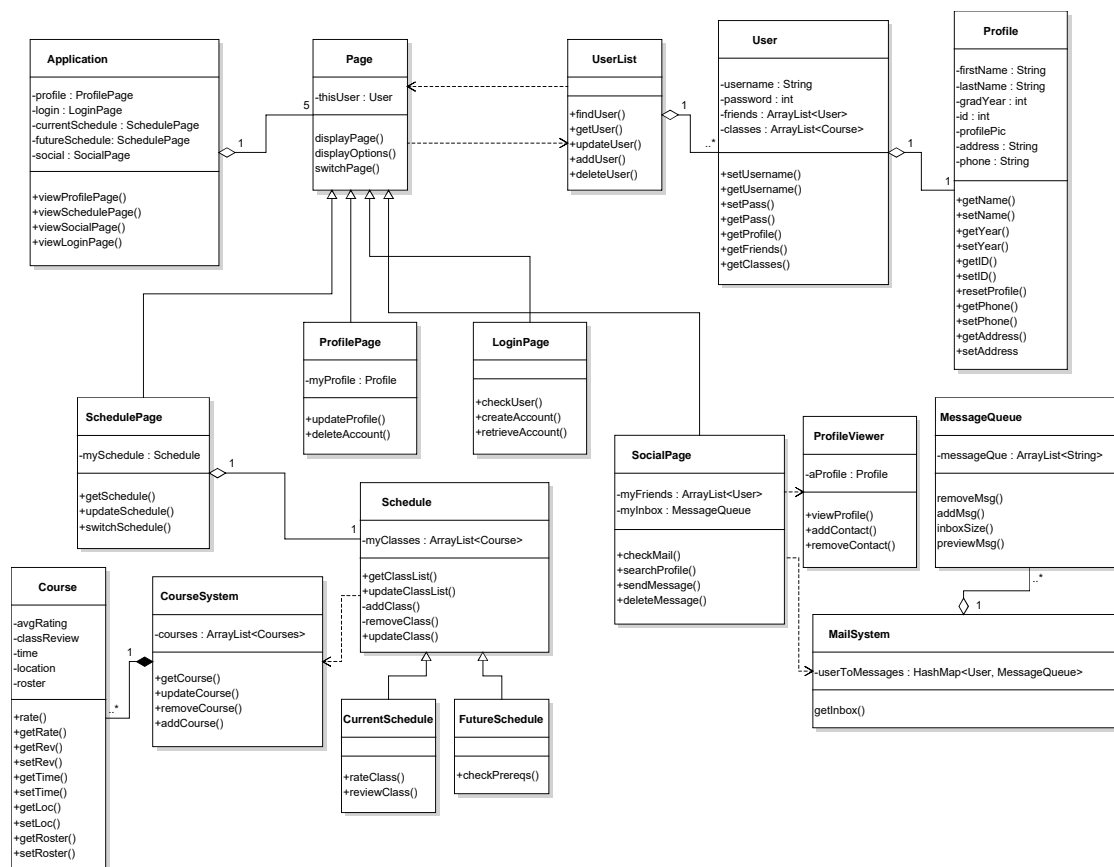
MVC Model



2.2 Discussion of Alternative Designs

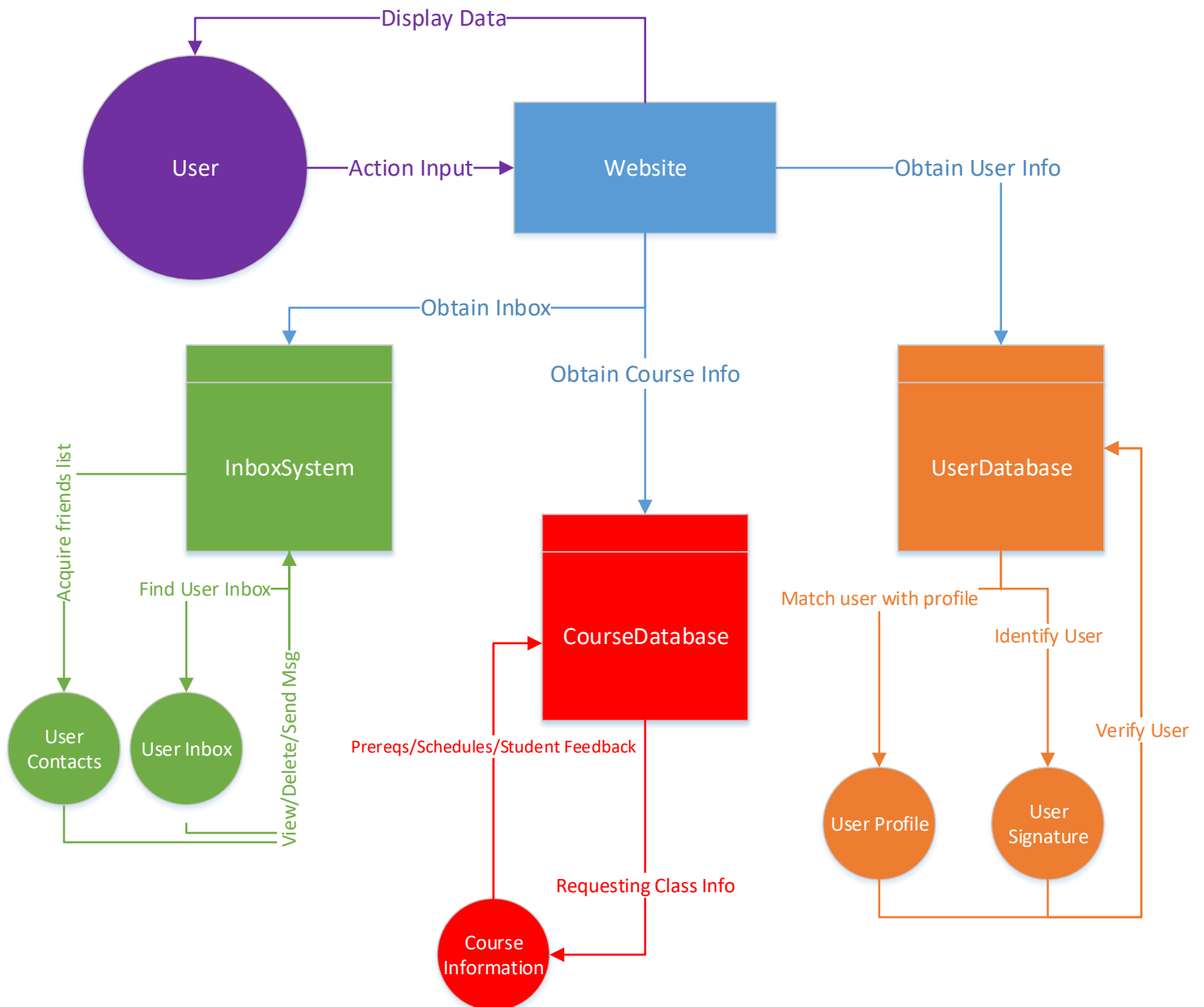
SCAM will be using an Object Oriented Design approach, hence a class diagram is very much necessary to show the relationships between all the classes (as in program terminology, not educational institutional terminology) and incite a general idea of how the code will be implemented.

SCAM Class Diagram



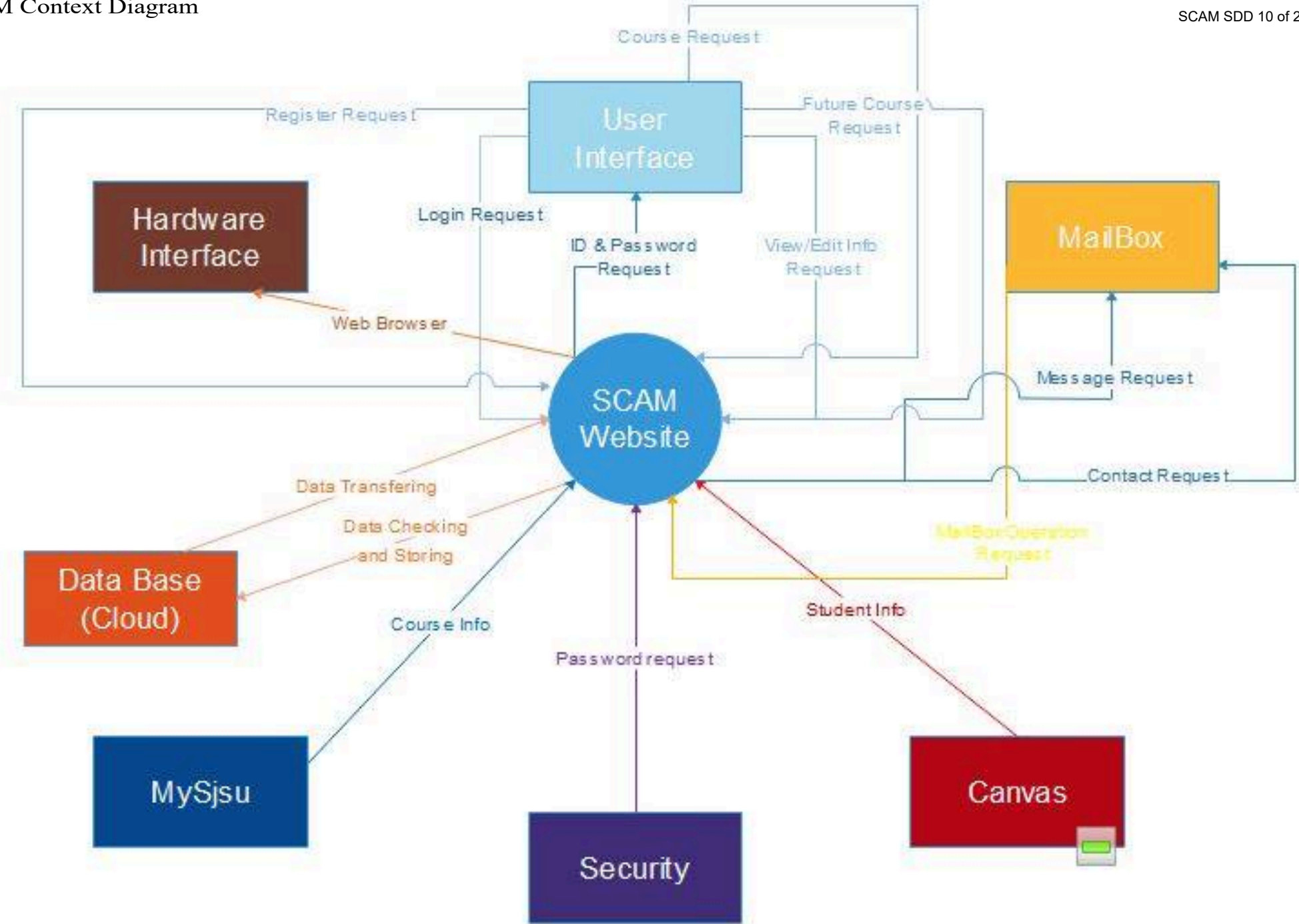
Another point of view that will come in handy is to perceive the flow of data throughout the program and how it takes input and how it produces an output. A data flow model is then used to display how the data is found, stored, exchanged, and manipulated.

Data Flow Diagram



2.3 System Interface Description

This portion of the document will describe how the system's interface is derived. The following context model depicts the relationship SCAM follows with other systems and where it lies in the software interface model.

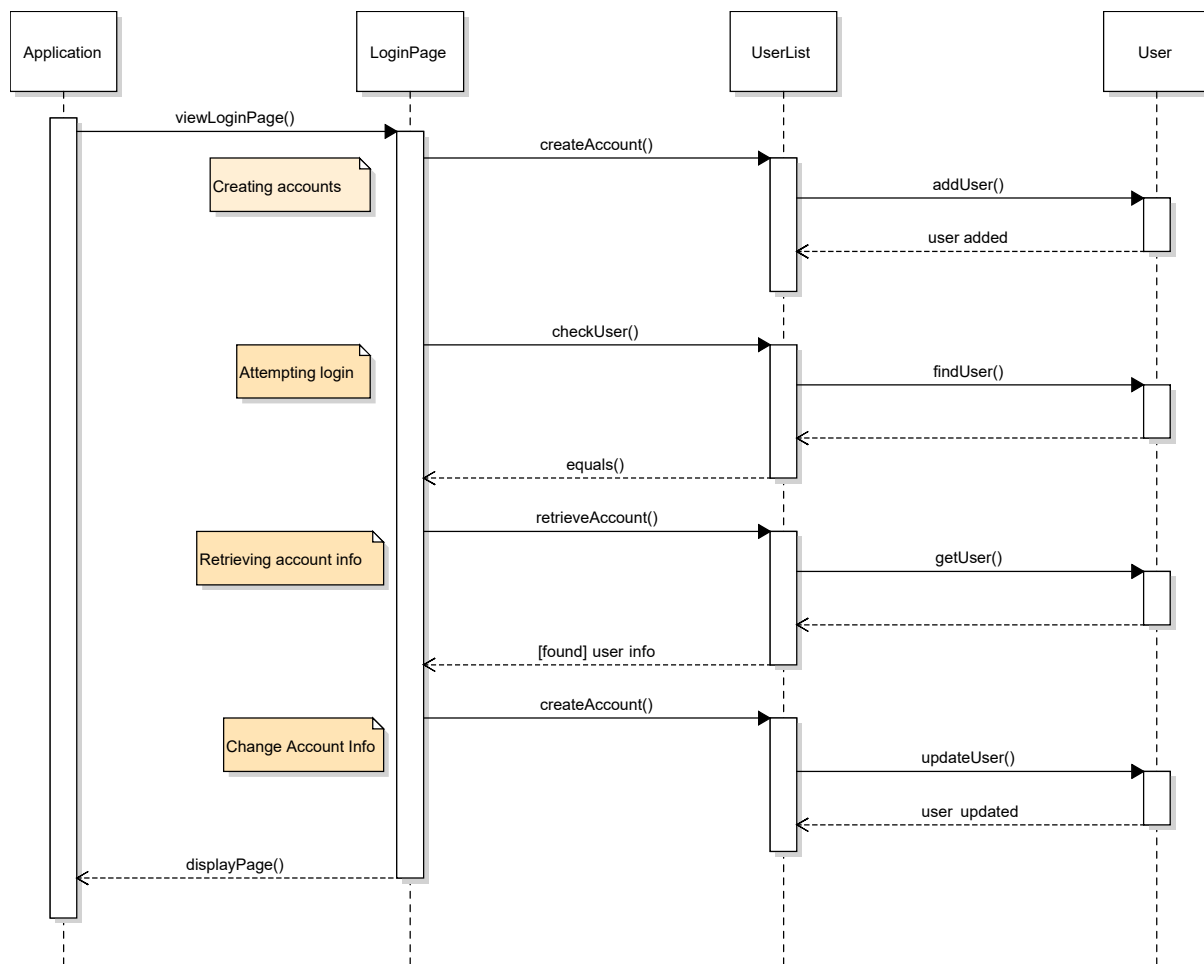


3.0 Detailed Description of Components

This section will primarily focus on the different components of SCAM and their corresponding functionalities. With that, a sequence diagram mirroring each general use case is used to show how the classes interact with one another during run time. The following components are addressed: Login page, Profile page, Current Schedule, Future Schedule, and Social page.

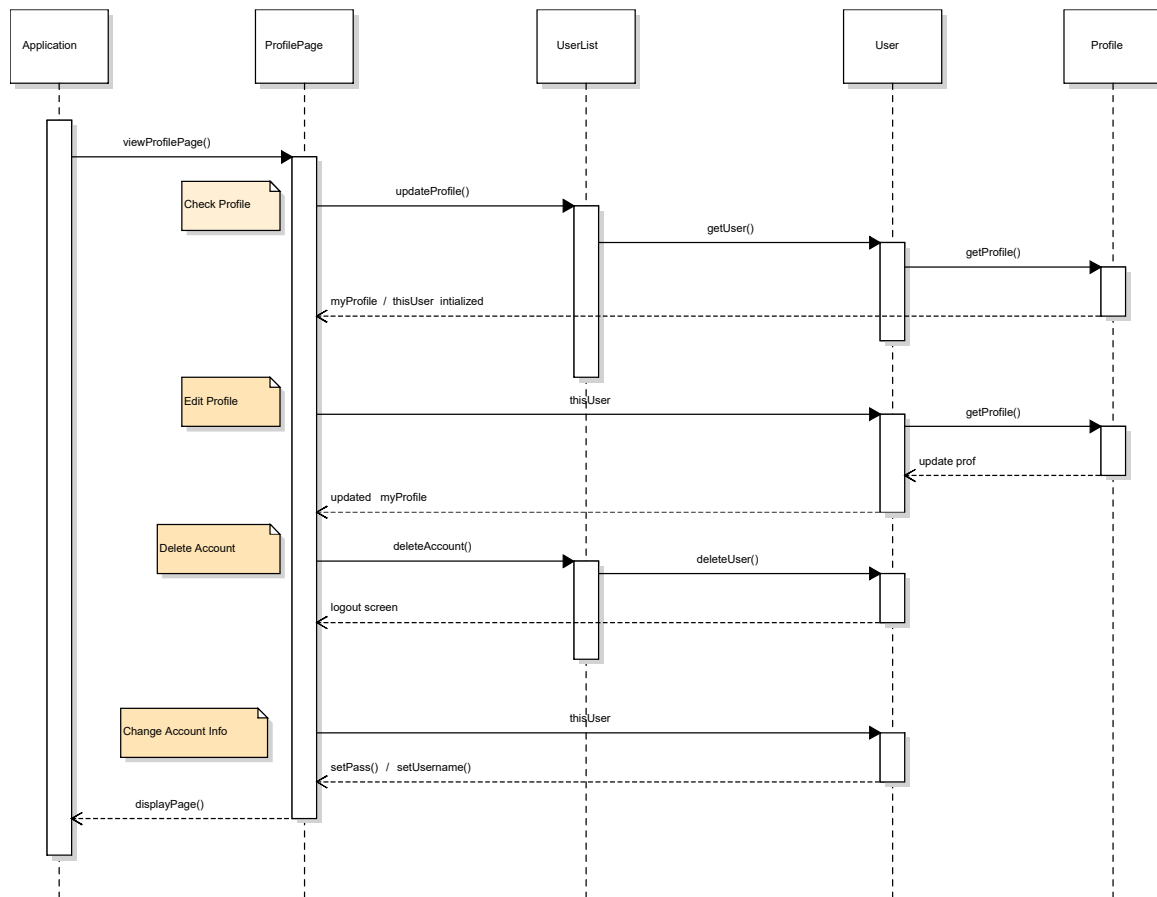
3.1 Login

The login page addresses many of the common functionalities with accounts. Logging in, making an account, retrieving accounts, and such.



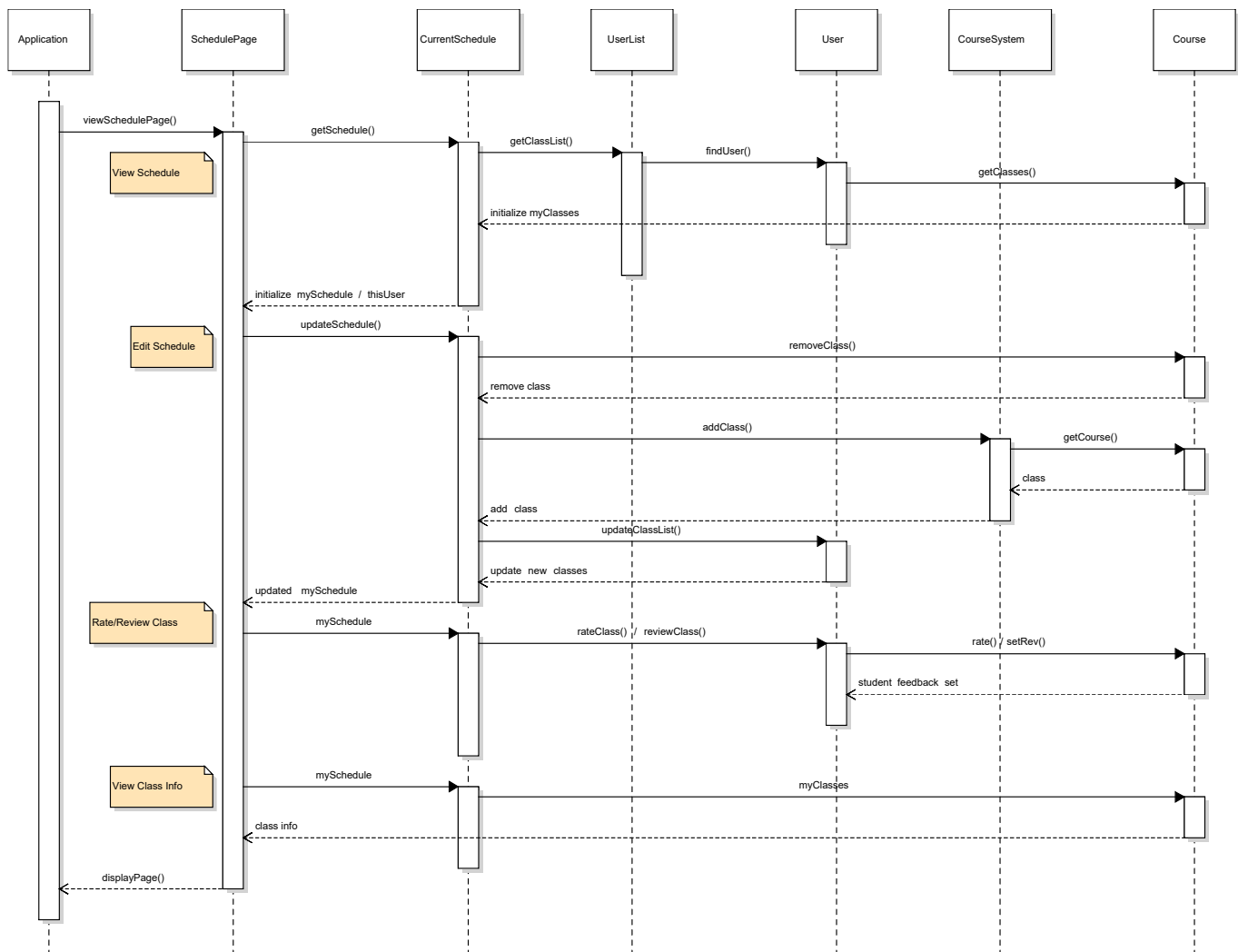
3.2 Profile

The profile page works on providing every user with a profile that contains all their information. This will hold functions such as deleting accounts, manipulating profile, and actually seeing the page itself.



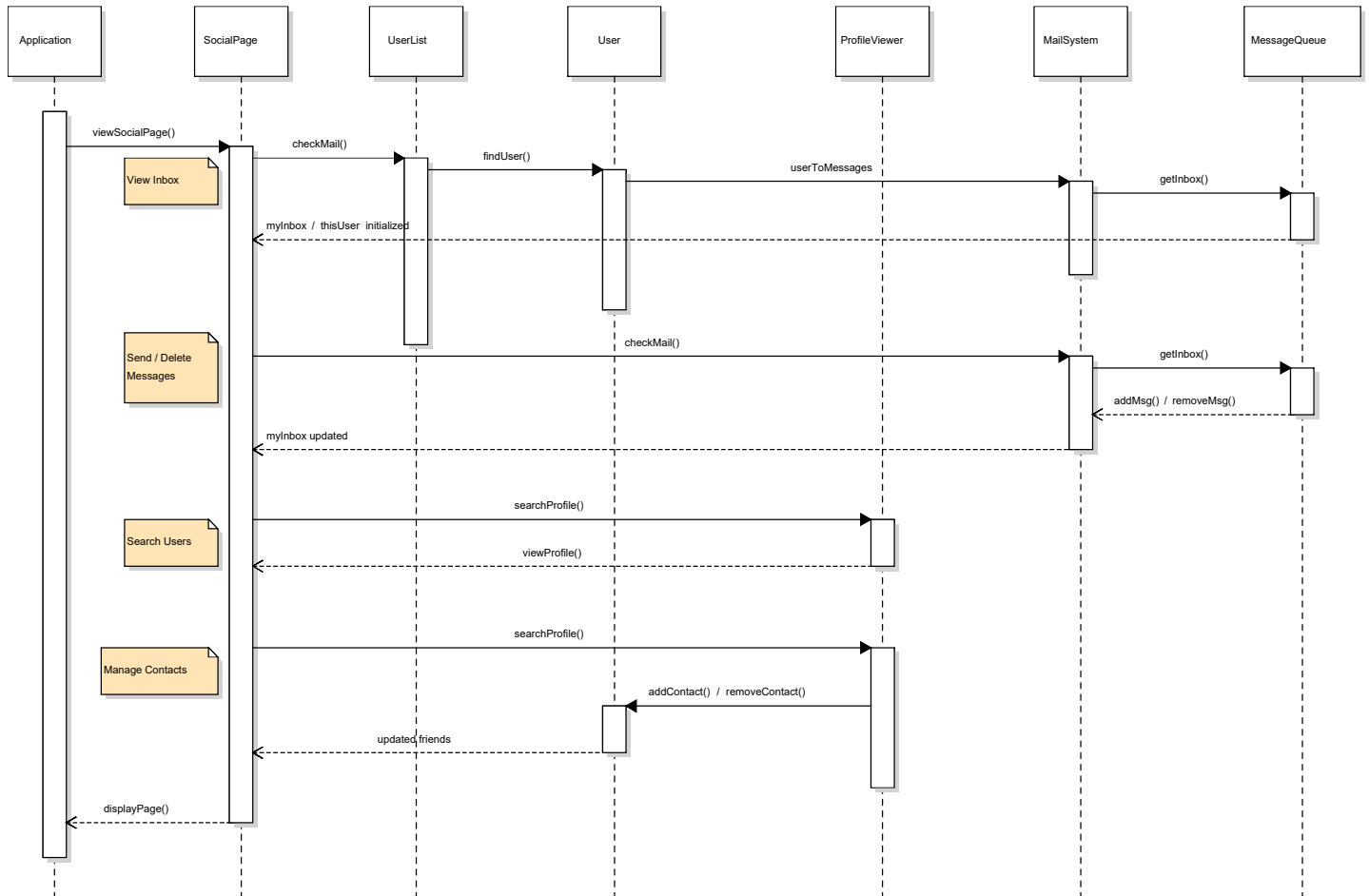
3.3 Current Schedule

The current schedule mirrors the user's current semester course. This will compose a visual representation of the user's classes and allow the user to edit it, look into a class, and rate classes.



3.4 Social Page

The social page manages all of the user's friends list and inbox. This involves sending and receiving messages, as well as adding and removing contacts. The search bar is also utilized for user to preview other user profiles.



4.0 User Interface Design

This portion of the document shall describe the user interface and the context followed by the user's point of view. This is all represented by flowcharts and models of all sorts of processes from the system and the user.

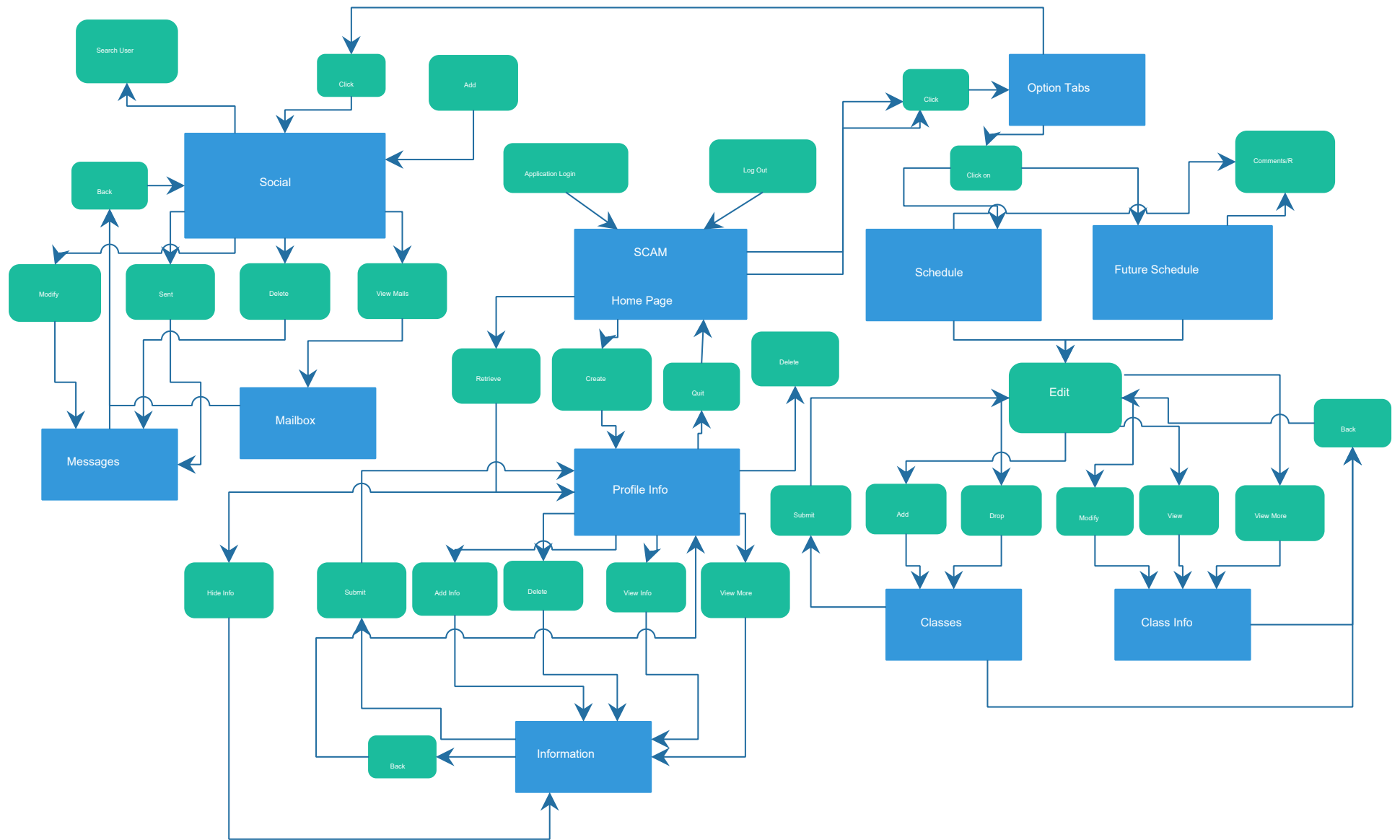
4.1 Description of User Interface

The user interface will be described for what the user will see based on which menu they're at and which options they take. This is shown in the later following UI Transition diagram. Then, three other diagrams (process, activity, and state transition) will be used to show the objects and their actions.

4.1.1 Screen Display

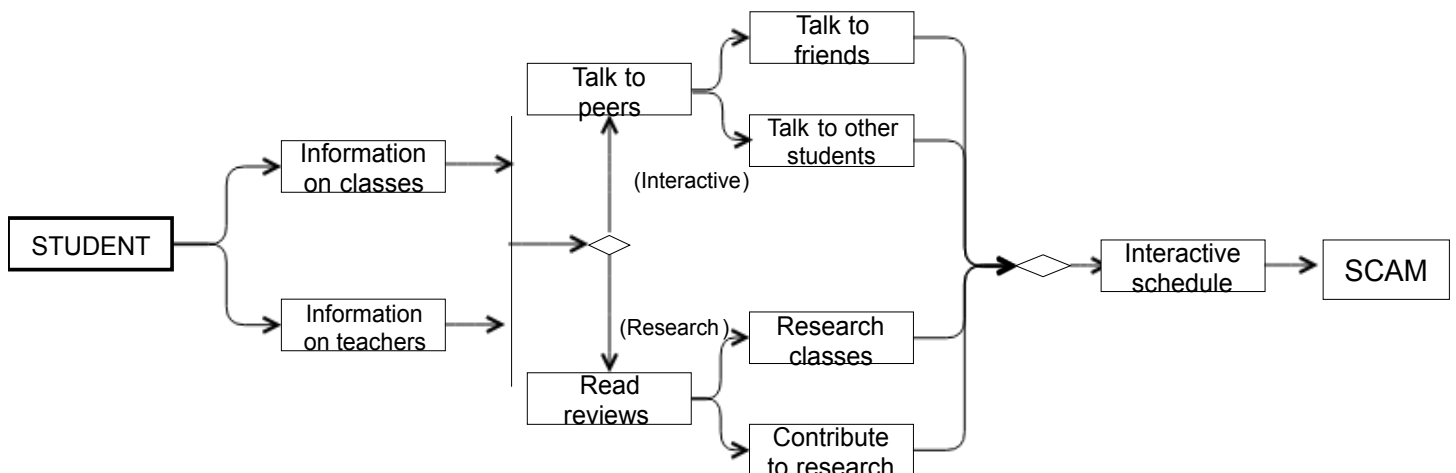
The following UI Transition diagram models the process that the user will experience depending on how they navigate through all the menus. There are five main screen displays to look for, the Login page, the Profile page, the two Schedules, and the Social page.

UI Transition Diagram

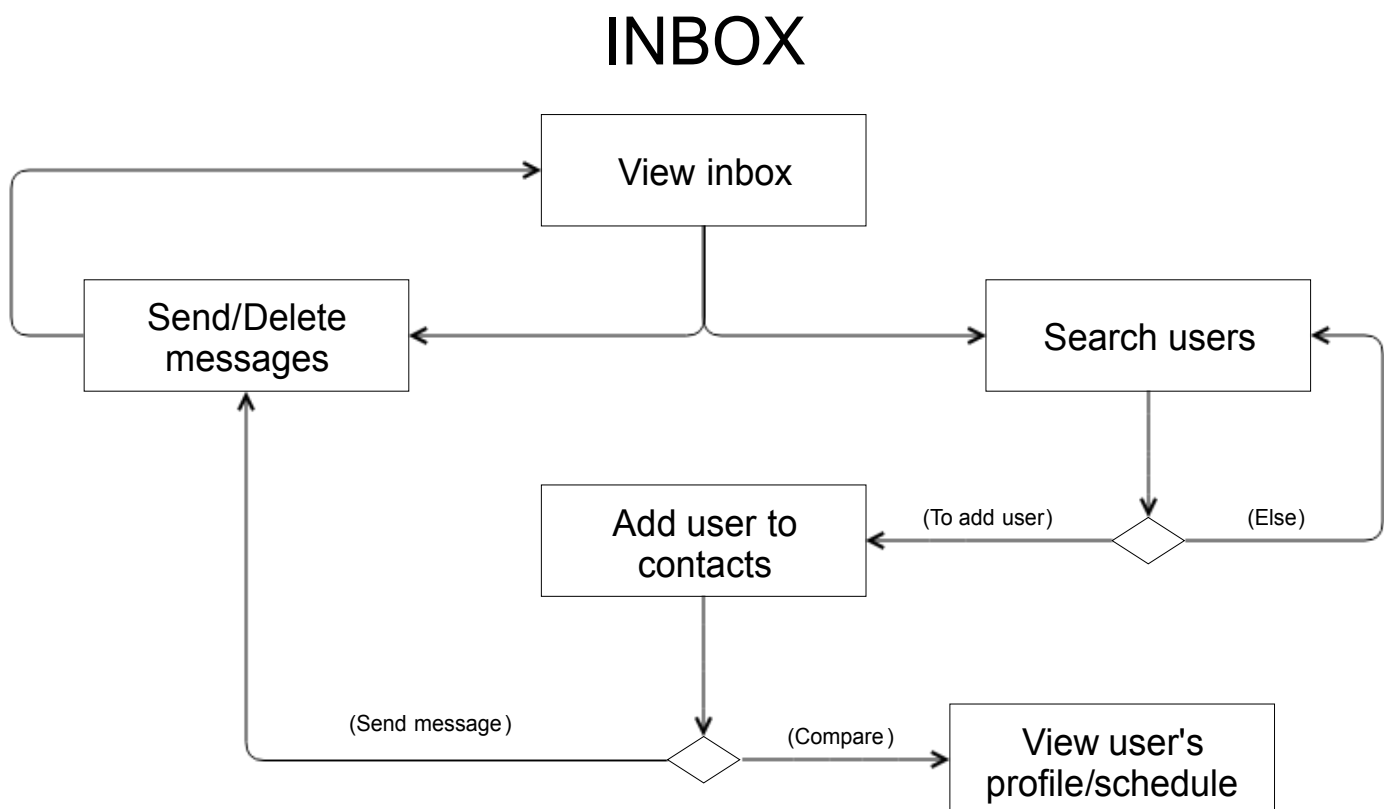
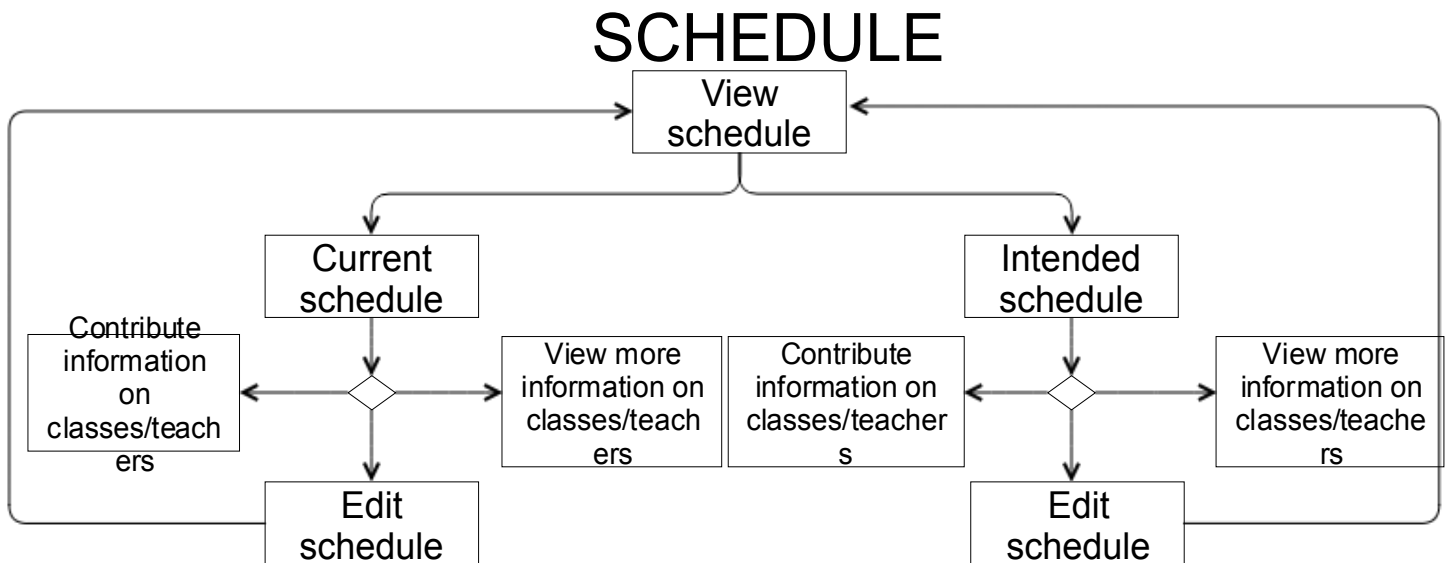


4.1.2 Objects and Actions

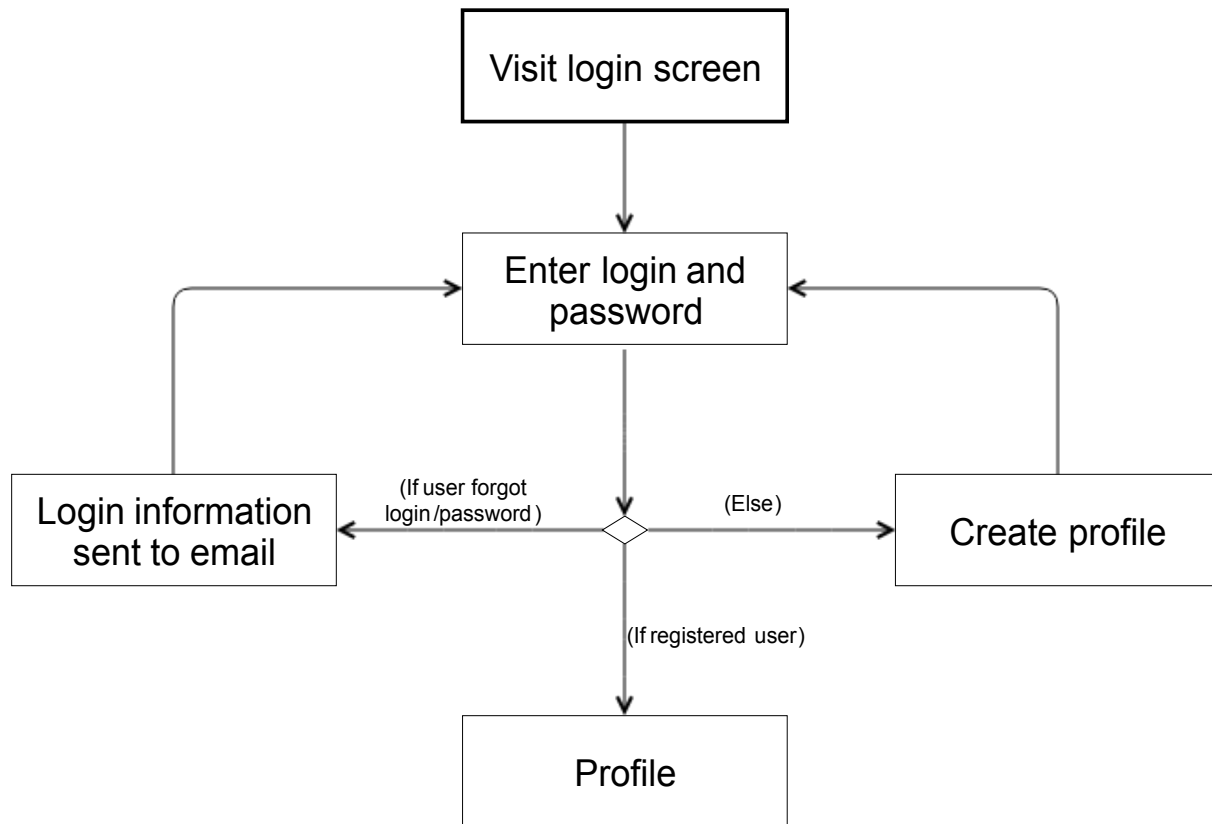
The first thing to take note of is the user's thought processes and how he may interact with SCAM through certain contexts. In a sense, there is a need for a flow chart that signifies how the typical quality of life with a user is like when they utilize SCAM. The flow chart will represent the user's actions in a daily basis and where the program fits into such a context. This is modeled by the following Process Model.



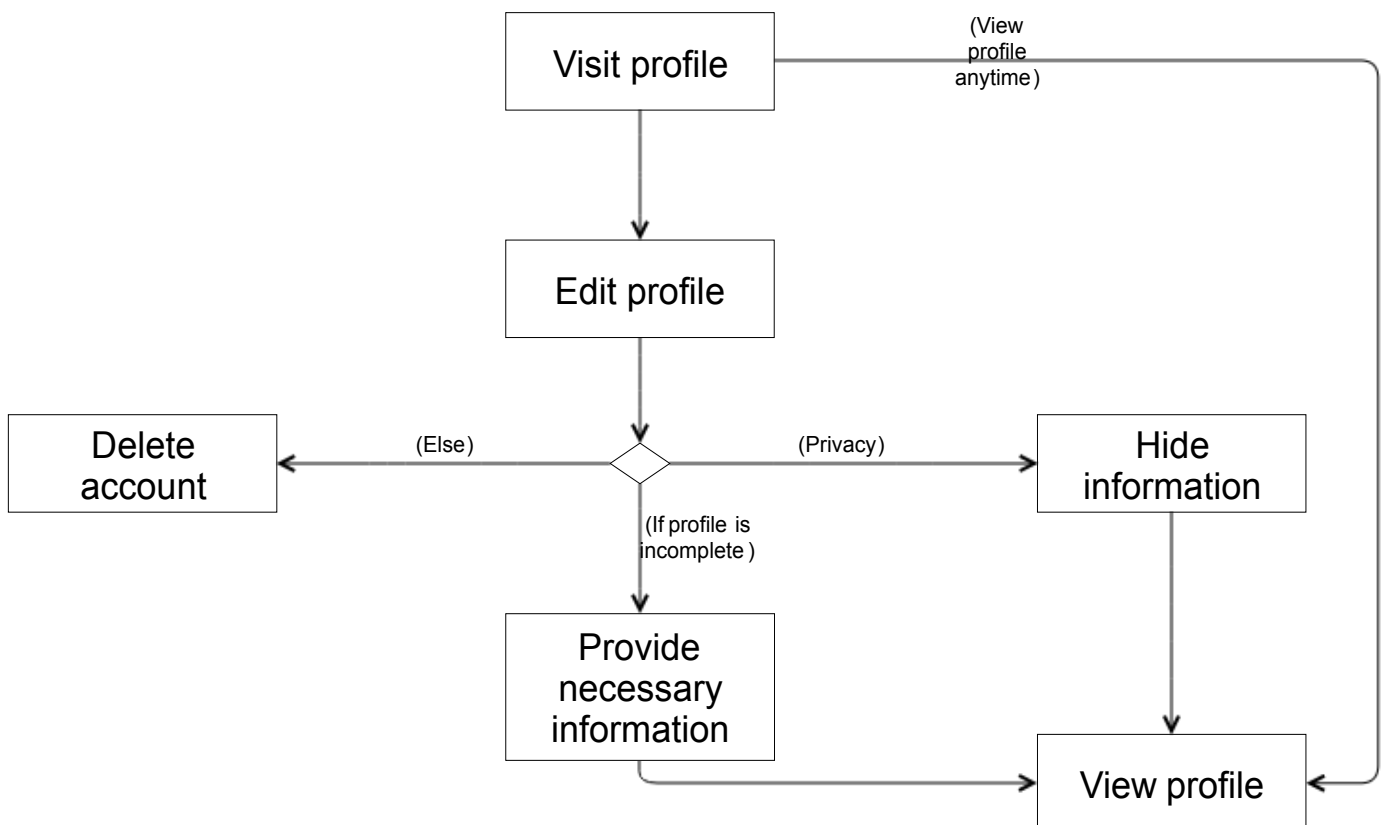
The next thing to note is how the user can fully manipulate each feature in SCAM without a vague generalized approach, but rather for specific user stories. This is modeled by the Activity Diagram. The following four activity diagrams are modeled over the four main categories of uses for SCAM.



LOGIN



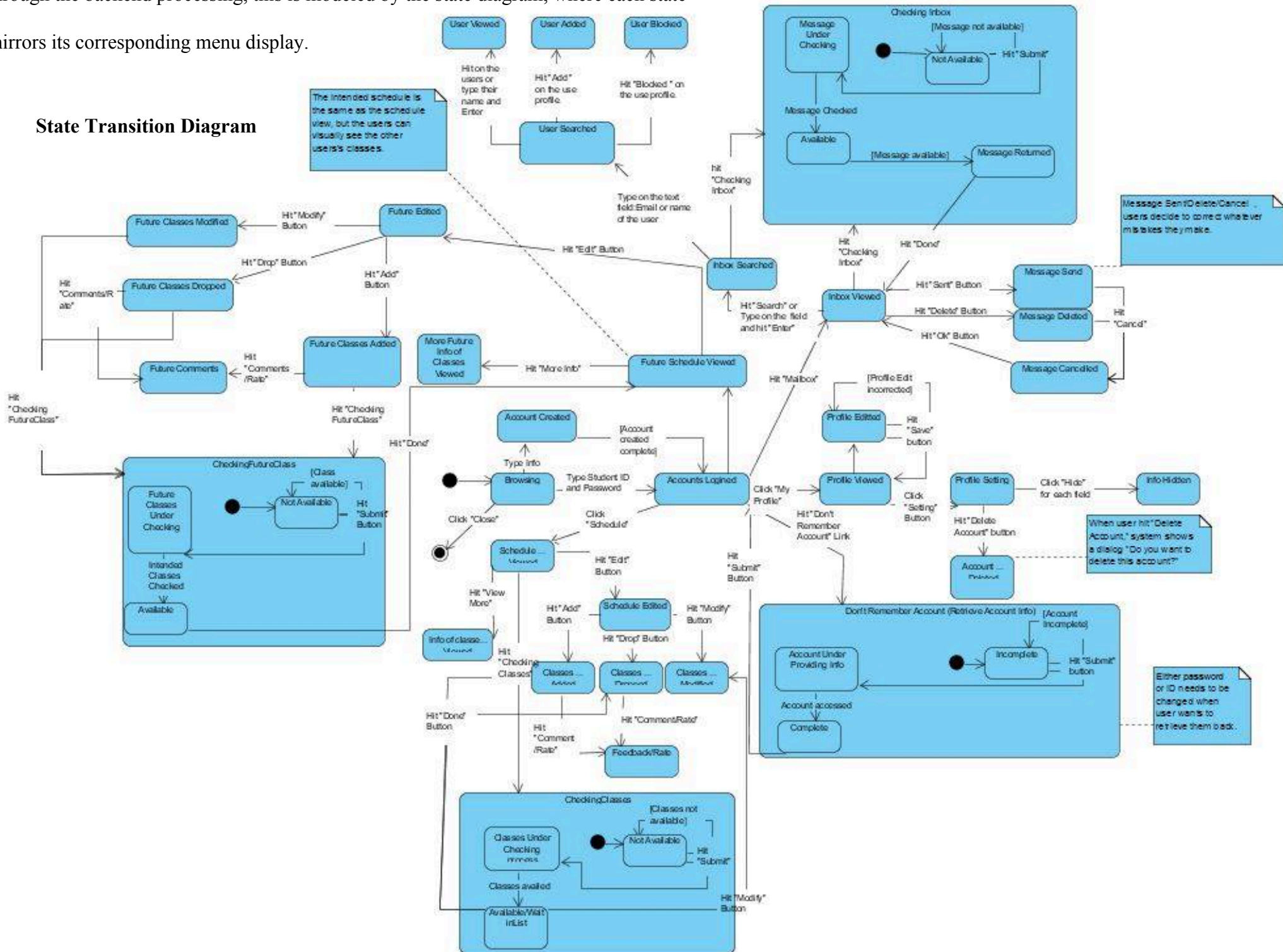
PROFILE



through the backend processing, this is modeled by the state diagram, where each state mirrors its corresponding menu display.

```

graph TD
    Start(( )) --> UV[User Viewed]
    Start --> UA[User Added]
    Start --> UB[User Blocked]
    style Start fill:none,stroke:none
    style UV fill:#add8e6,stroke:#000,stroke-width:1px
    style UA fill:#add8e6,stroke:#000,stroke-width:1px
    style UB fill:#add8e6,stroke:#000,stroke-width:1px
  
```



5.0 Additional Material

This component is accompanied by anymore miscellaneous portions of the entire document.

5.1 Architectural View of Deployment

The architectural view of deployment models all the important aspects that will be needed when the product is ready for release. It describes the physical environment in which the product runs and the technical requirements for each node.

