|  |
| --- |
| Mobile Student Lookup  CSSE 374 - Milestone 5  2/13/2012  Brandon Knight, Mark Vitale, Ann Say, Chris Gropp |

Contents

[Introduction 3](#_Toc316909321)

[Gang of Four Discussion 4](#_Toc316909322)

[Adapter 4](#_Toc316909323)

[Abstract Factory 5](#_Toc316909324)

[Singleton 5](#_Toc316909325)

[Composite 6](#_Toc316909326)

[Observer 7](#_Toc316909327)

[Acceptance Test Plan 8](#_Toc316909328)

[Invalid or no credentials use case 8](#_Toc316909329)

[View Own Schedule 9](#_Toc316909330)

[Search By Username 10](#_Toc316909331)

[Search By Classroom 11](#_Toc316909332)

[Search By Course ID 12](#_Toc316909333)

[Search Future or Past Schedule 13](#_Toc316909334)

[Overlaying Schedules 14](#_Toc316909335)

[Find Contact Information Use Case 15](#_Toc316909336)

[E-mail User Use Case 16](#_Toc316909337)

[E-mail a Schedule Use Case 17](#_Toc316909338)

[Add to Favorites Use Case 18](#_Toc316909339)

[View Favorites 19](#_Toc316909340)

[Sync Schedule Use Case 20](#_Toc316909341)

[View Class Roster Use Case 21](#_Toc316909342)

[Activity Diagrams 22](#_Toc316909343)

[Person Info Display 22](#_Toc316909344)

[Settings View Controller 23](#_Toc316909345)

[Schedule Display 24](#_Toc316909346)

[New Sequence Diagrams 25](#_Toc316909347)

[Change Authentication 25](#_Toc316909348)

[Display Schedule 26](#_Toc316909349)

[Display Student 27](#_Toc316909350)

[The Who Done What Table 28](#_Toc316909351)

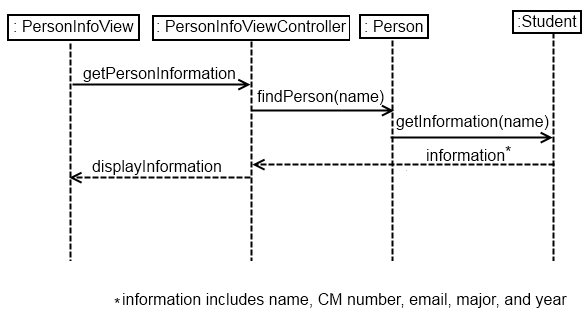
# Introduction

We are producing a mobile application for iPhone and iPod Touch (iOS) for the clients Tim Ekl and Eric Stokes. The features of this application will include, but not be limited to: searching for a student’s schedule by username, searching by classroom, searching by separate terms, and overlaying schedules (comparing more than one schedule). The user needs this application to be easy to use, easy to navigate, and easy to read. Having an application on the iPhone will drastically improve the process of looking up schedules on the iPhone. In the future, we may extend this project to iPad, or even Android platforms.

Currently to view their schedule on the iPhone or iPod Touch, students have to manually navigate through the Safari mobile web browser to the Rose-Hulman website, and then to the registrar’s office, logging in with their Kerberos password several times before they can access their schedule. We want to simplify this experience for mobile devices, and make it easier and quicker for students to access this data. We will be designing a mobile application on iOS for iPhone, which will allow students to access all available schedule data from the registrar’s office website. The application will be capable of storing the password, so students will not have to type it in more than once. Students should be able to easily look up their own schedules, easily find a classmate’s schedule, and view all of the members of their classes.

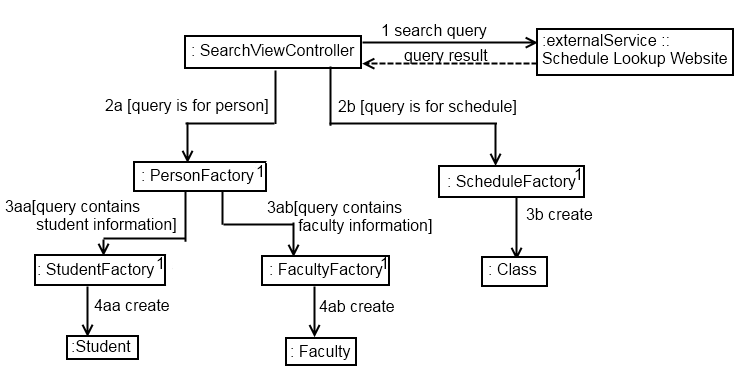
# Gang of Four Discussion

## Adapter

Our system scrapes information from the school website and displays this information in a suitable format. These information types are encapsulated in Adapter classes, whose naming scheme comes from the domain model, such as Student, Faculty, and Class. These classes are Adapters because they adapt the raw data brought in from the website into a class with attributes, which can be manipulated easier in the domain.

## Abstract Factory

There are various Factories in our design, used to receive and parse raw data from the website, and send the information to the various model classes, such as Student, Faculty and Classes. This way, only a few classes have to deal with html code, and not the individual model classes. This also keeps the cohesion high within the Factory and model classes. The abstract factories are the PersonFactory and the Schedule Factory. The PersonFactory then passes the information to either the StudentFactory or the FacultyFactory, depending on what type the query information is. These sub factories that correspond to classes, such as the StudentFactory and ClassFactory, further transform the parsed data received and create their corresponding classes. We decided to create sub-factories instead of having the high level factories create all the model classes. This was to keep cohesion on the factories high, and instead of having to know how to parse the various results into people or classes or schedules, it sends the information to the specialized factory.



## Singleton

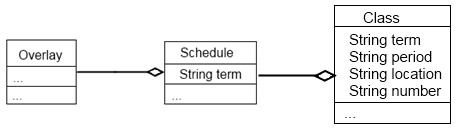
A Singleton class is a class where there is only one instance of it in the entire design model. The various factories used in the parsing of the web page are Singletons, since only one instance is needed to parse the information. This design was chosen instead of having a factory created at every query, because it is uncertain how many searches the user may have in the lifetime of the application. If a new factory was created at every search query, then the application would get cluttered with excess and redundant factories very quickly.

## Composite

The overlay is composed of one or more schedules. The schedules are interlaced together so that the class times line up. In essence, an overlay looks like a normal schedule, except with the possibility for multiple classes to appear in a single time slot. Thus, it followed that the overlay should be treated like a single schedule, since like a schedule it is a combination of classes.

Other compositions in the system are the Favorites class - a Favorites list is composed of people - and the Schedule class, which is composed of Courses.

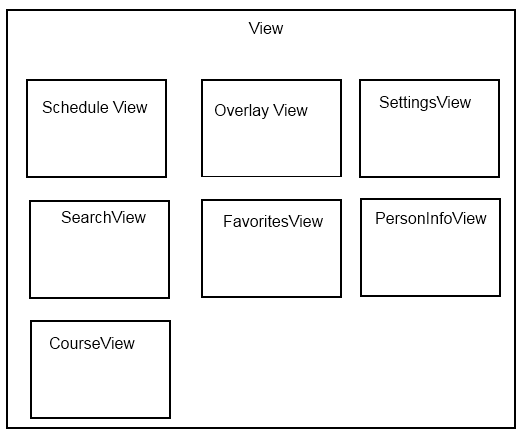
The following class diagram, taken from the domain model, shows the composition relationships between classes.



## Observer

Because of the nature of the iOS environment, all of the View classes are Observers, since they are constantly listening for an input from the user, namely a touch on the screen. The response action is dependent on where and how the user touches the screen, such as touching a button, or dragging their finger across the screen in a scrolling manner.

All of the classes in the view architectural layer – ScheduleView, OverlayView, SearchView, FavoritesView, CourseView, SettingsView, and PersonInfoView are observers. They correspond to all of the various views the user of the application would see.



# Acceptance Test Plan

The following test cases cover the full planned functionality of the app. Many have not yet been implemented; as of this writing, the app is a work in progress. All test cases implicitly assume the user is on the proper navigation page for the task they are attempting.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Scenario | Description | Condition: Username and password are correct | Condition: Network connection available | Expected Result |
| 1 | 1 | Basic Flow: everything is normal | V | V | User is logged in and is viewing his/her own schedule |
| 2 | 1 | Basic Flow: username or password is incorrect | I | V | User is notified and prompted enter valid to username and password |
| 3 | 2 | Alternate Flow: No Network Connection | N/A | I | User views error stating that network connection required for this app. |

## Invalid or no credentials use case

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Scenario | Description | Condition: User is currently on “My Schedule” tab | Condition:  Valid credentials are stored in the app | Condition: Network connection available | Expected Result |
| 1 | 1 | Basic Flow: everything is normal | V | V | V | User views his/her own schedule in a tabular format |
| 2 | 2 | Alternate Flow: username or password is incorrect | N/A | I | V | User is notified and prompted enter valid to username and password |
| 3 | 3 | Alternate Flow: No Network Connection | N/A | N/A | I | User views error stating that network connection required for this app. |

## View Own Schedule

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Scenario | Description | Condition: User typed in complete, correct username | Condition:  Valid credentials are stored in the app | Condition: Network connection available | Expected Result |
| 1 | 1 | Basic Flow: everything is normal | V | V | V | User views an information page on the user with the specified username. |
| 2 | 2 | Alternate Flow: User typed an incomplete or partial username | I | V | V | User views a list of usernames that match the current search entry. Selecting any one will take the user to the information page on the user with the selected username. If no matches, an empty list will be displayed. |
| 3 | 3 | Alternate Flow: username or password is incorrect | N/A | I | V | User is notified and prompted enter valid to username and password |
| 4 | 4 | Alternate Flow: No Network Connection | N/A | N/A | I | User views error stating that network connection required for this app. |

## Search By Username

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Scenario | Description | Condition: User typed in complete, correct classroom number | Condition:  Valid credentials are stored in the app | Condition: Network connection available | Expected Result |
| 1 | 1 | Basic Flow: everything is normal | V | V | V | User views an information page on the classroom with the specified classroom number. |
| 2 | 2 | Alternate Flow: User typed an incomplete or partial classroom number | I | V | V | User views a list of classroom numbers that match the current search entry. Selecting any one will take the user to the information page on the classroom with the selected classroom number. If no matches, an empty list will be displayed. |
| 3 | 3 | Alternate Flow: username or password is incorrect | N/A | I | V | User is notified and prompted enter valid to username and password |
| 4 | 4 | Alternate Flow: No Network Connection | N/A | N/A | I | User views error stating that network connection required for this app. |

## Search By Classroom

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Scenario | Description | Condition: User typed in complete, correct username | Condition:  Valid credentials are stored in the app | Condition: Network connection available | Expected Result |
| 1 | 1 | Basic Flow: everything is normal | V | V | V | User views an information page on the course with the specified course ID. |
| 2 | 2 | Alternate Flow: User typed an incomplete or partial course ID | I | V | V | User views a list of course IDs that match the current search entry. Selecting any one will take the user to the information page on the course with the selected course ID. If no matches, an empty list will be displayed. |
| 3 | 3 | Alternate Flow: username or password is incorrect | N/A | I | V | User is notified and prompted enter valid to username and password |
| 4 | 4 | Alternate Flow: No Network Connection | N/A | N/A | I | User views error stating that network connection required for this app. |

## Search By Course ID

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Scenario | Description | Condition: User selects valid quarter from individual’s information page | Condition:  Valid credentials are stored in the app | Condition: Network connection available | Expected Result |
| 1 | 1 | Basic Flow: everything is normal | V | V | V | User views the schedule of the individual from the selected quarter |
| 2 | 2 | Alternate Flow: User selected an invalid quarter | I | V | V | User will see a warning that the individual was not enrolled during that quarter and will view the current schedule for the individual |
| 3 | 3 | Alternate Flow: username or password is incorrect | N/A | I | V | User is notified and prompted enter valid to username and password |
| 4 | 4 | Alternate Flow: No Network Connection | N/A | N/A | I | User views error stating that network connection required for this app. |

## Search Future or Past Schedule

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Scenario | Description | Condition: User is trying to overlay an allowed number of schedules | Condition: User wants to clear the overlay | Condition: User adds the same schedule multiple times | Condition:  Valid credentials are stored in the app | Condition: Network connection available | Expected Result |
| 1 | 1 | Basic Flow: everything is normal | V | I | I | V | V | User views selected schedules in the overlay view. |
| 2 | 2 | Alternate Flow: User has tried to overlay too many schedules | I | I | I | V | V | User sees a warning that no more schedules can be placed in the overlay, the most recent attempt to add a schedule will be unsuccessful. |
| 3 | 3 | Alternate Flow: User wants to clear the overlay | V | V | I | V | V | All schedules are removed from the overlay. User is returned to the empty list of schedules in the overlay. |
| 4 | 4 | Alternate Flow: | V | I | V | V | V | No checks are made, the user will see the same schedule appear multiple times in the overlay. |
| 5 | 5 | Alternate Flow: username or password is incorrect | N/A | N/A | N/A | I | V | User is notified and prompted enter valid to username and password |
| 6 | 6 | Alternate Flow: No Network Connection | N/A | N/A | N/A | N/A | I | User views error stating that network connection required for this app. |

## Overlaying Schedules

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Scenario | Description | Condition: Valid credentials are stored in app | Condition: Network connection available | Condition: Has search tab bar button selected | Condition:  Selects “username” option | Condition:  Types full username. | Expected Result |
| 1 | 1 | Basic Flow: user selects search bar | V | V | V | V | V | User will be able to search |
| 2 | 1 | Basic Flow: username or password is incorrect | I | V | N/A | N/A | N/A | User is notified and prompted enter valid to username and password |
| 3 | 2 | Alternate Flow: types only partial username | N/A | I | V | V | I | The application should list available users who match the partial username. (if any). |

## Find Contact Information Use Case

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Scenario | Description | Condition: User is viewing contact information | Condition: Network connection available | Condition: User has native e-mail app set up. | Expected Result |
| 1 | 1 | Basic Flow: User navigates to individuals contact information. | V | V | V | User will be able to go to users e-mail through touch. |
| 2 | 1 | Basic Flow: username or password is incorrect | N/A | V | N/A | User is notified and prompted enter valid to username and password |
| 3 | 2 | Alternate Flow: No e-mail account set up | N/A | V | I | The OS will prompt the user to create a e-mail account. |

## E-mail User Use Case

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Scenario | Description | Condition: User is viewing a schedule | Condition: Network connection available | Condition: User has native e-mail app set up. | Expected Result |
| 1 | 1 | Basic Flow: User navigates to any viewable schedule | V | V | V | User will be able to click the “e-mail schedule” button. |
| 2 | 1 | Basic Flow: username or password is incorrect | N/A | V | N/A | User is notified and prompted enter valid to username and password |
| 3 | 2 | Alternate Flow: No e-mail account set up | N/A | V | I | The OS will prompt the user to create a e-mail account. |

## E-mail a Schedule Use Case

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Scenario | Description | Condition: User has searched for a user | Condition: Network connection available | Expected Result |
| 1 | 1 | Basic Flow: User navigates to individuals contact information. | V | V | User will be able to hit “add to favorites” |
| 2 | 1 | Basic Flow: username or password is incorrect | N/A | V | User is notified and prompted enter valid to username and password |
| 3 | 2 | Alternate Flow: User is already in the favorites | V | V | The application will not add the user to the favorites list. |
| 4 | 2 | Alternate Flow: No internet or network connection | N/A | I | The application will display a toast message and will not display any information that requires network. |

## Add to Favorites Use Case

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Scenario | Description | Condition: User is any page. | Condition: Network connection available | Condition: Username and Password Correct | Expected Result |
| 1 | 1 | Basic Flow: User presses “My favorites tab” | V | V | V | User will view his or her favorites. |
| 2 | 1 | Basic Flow: username or password is incorrect | N/A | V | I | User is notified and prompted enter valid to username and password |
| 3 | 2 | Alternate Flow: No internet connection | N/A | I | N/A | The application will display a toast message and will not display any information that requires network. |

## View Favorites

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Scenario | Description | Condition: User is viewing his or her schedule. | Condition: Network connection available | Condition: Username and Password Correct | Expected Result |
| 1 | 1 | Basic Flow: User chooses to add his or her schedule to the calendar application. | V | V | V | The application will add the classes of the current term to the schedule. |
| 2 | 1 | Basic Flow: username or password is incorrect | N/A | V | I | User is notified and prompted enter valid to username and password |
| 3 | 2 | Alternate Flow: No internet connection | N/A | I | N/A | The application will display a toast message and will not display any information that requires network. |
| 4 | 2 | Alternate Flow: User chooses a different term to sync. | V | V | V | The application will add the classes to the schedule. |

## Sync Schedule Use Case

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Scenario | Description | Condition: User is viewing course information page. | Condition: Network connection available | Condition: Username and Password Correct | Expected Result |
| 1 | 1 | Basic Flow: User navigates to course roster | V | V | V | User will view the course roster. |
| 2 | 1 | Basic Flow: username or password is incorrect | N/A | V | I | User is notified and prompted enter valid to username and password |
| 3 | 2 | Alternate Flow: No internet connection | N/A | I | N/A | The application will display a toast message and will not display any information that requires network. |

## View Class Roster Use Case

# Integration Testing

While it is be possible to test the internal connections of the system independently from the GUI, for the most part it is simply easier to use the GUI to display the internal state. xCode provides extremely smooth integration between internal code and displays, and we have used built-in GUI functionality to test the internal workings.

While our system is unusual in this regard, there is still benefit to be had from testing components in isolation. Kerberos authentication, web scraping, keychain access, and regular expression parsing of html were all developed individually as code spikes. They were tested during their independent development, and function properly in isolation; the following tests are mostly apocryphal, but could prove useful were the system to be reconstructed from scratch (to run native on the Android, for example).

## Kerberos Authentication

**Goal:** Pass the authentication challenge given by the schedule lookup page.

**Inputs:** Username, password.

**Outputs:** Whether or not the given username and password are valid for the Rose-Hulman network.

**Other requirements:** Text entry must satisfy Apple’s UI guidelines, and must not auto-correct entered terms. Password entry must be secure (convert to asterisks, not saved anywhere unsafe).

## Keychain Access

**Goal:** Store a given piece of information in the Keychain for security, and then find it later.

**Input:** Username

**Output:** The same username as was entered.

**Other requirements:** Restart iPhone or otherwise clear app data between input and output stage to test persistence.

**Note:** On other platforms besides the iPhone, this test should be modified to account for that platform’s recommended security scheme. The intention is to securely store persistent data, however that is best accomplished.

## Regular Expression Parsing

**Goal:** Turn html into a specified domain object.

**Inputs:** Domain object type to be constructed, scraped html

**Outputs:** Properly entered domain object of specified type

## Web Scraping

**Goal:** Gather information from schedule lookup page, constructing necessary URLs based on search terms.

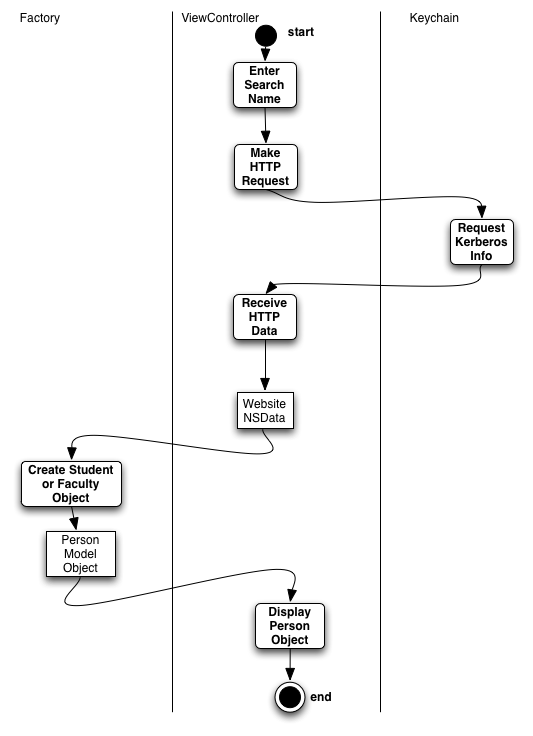
**Inputs:** Valid username and password, search terms

**Outputs:** html of pages relevant to search

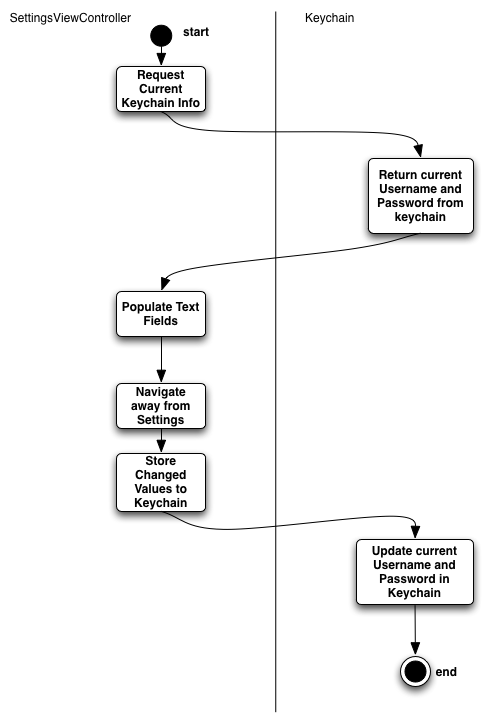
**Other Requirements:** This test relies on the functionality of Kerberos Authentication.

# Activity Diagrams

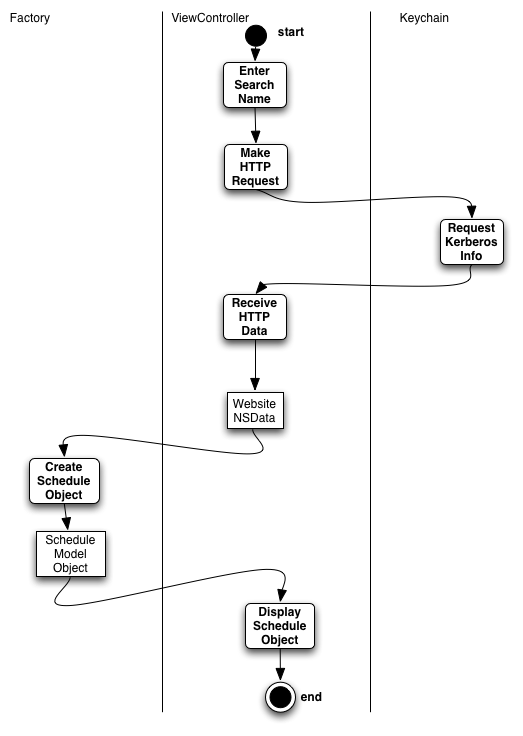
## Person Info Display



## Settings View Controller



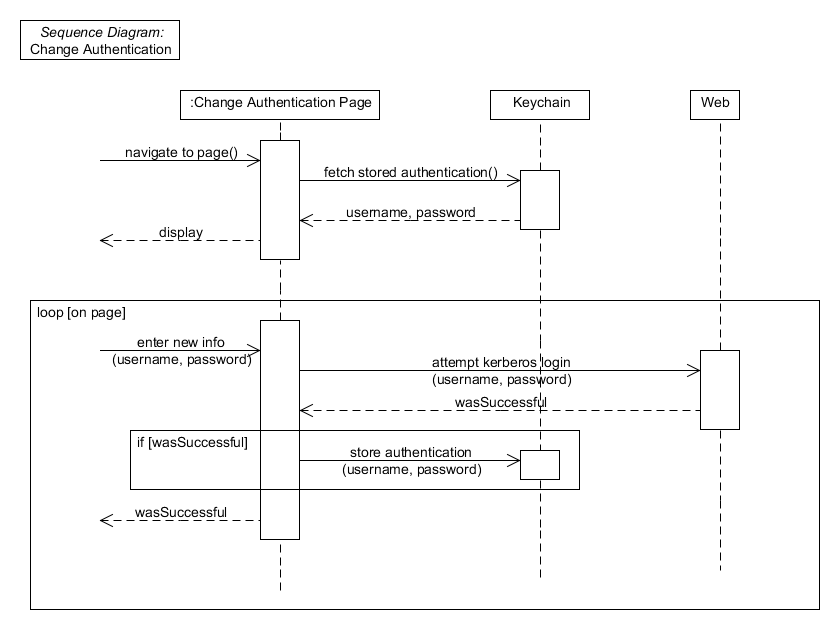
## Schedule Display



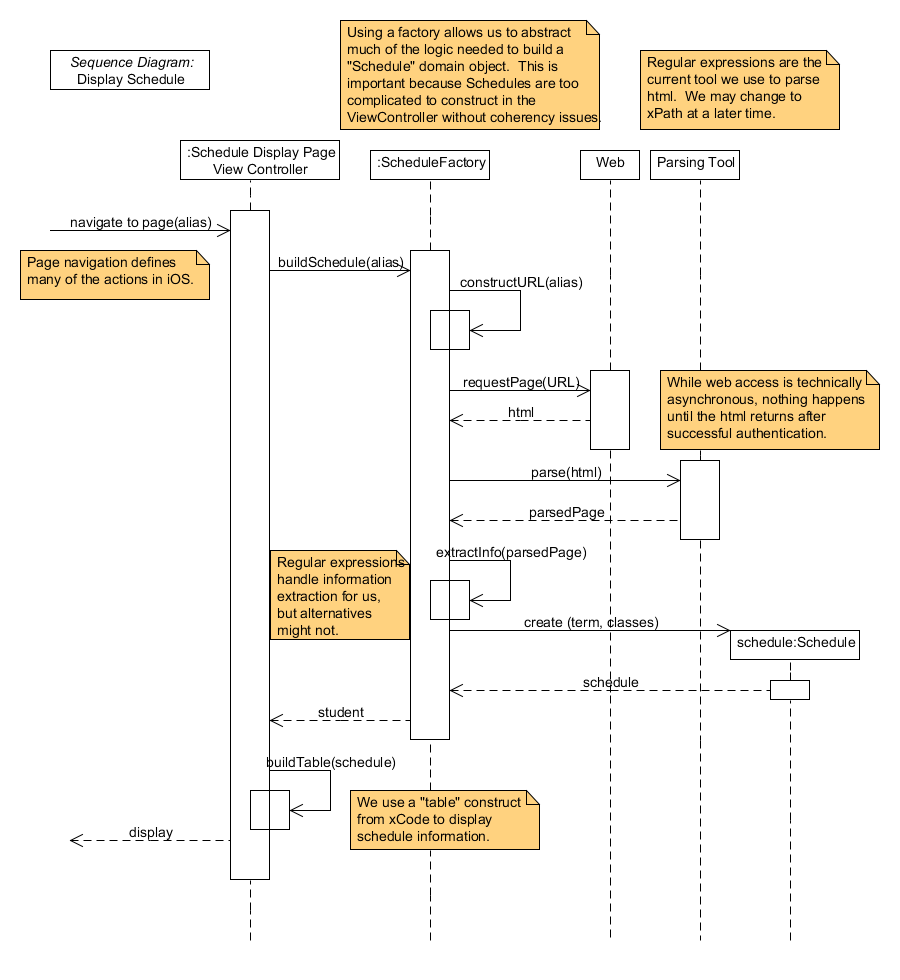
# New/Revised Sequence Diagrams

As the system matures, our design has been refined and elaborated upon. These newer diagrams are clearer and more accurate than previous iterations. They span present functionality.

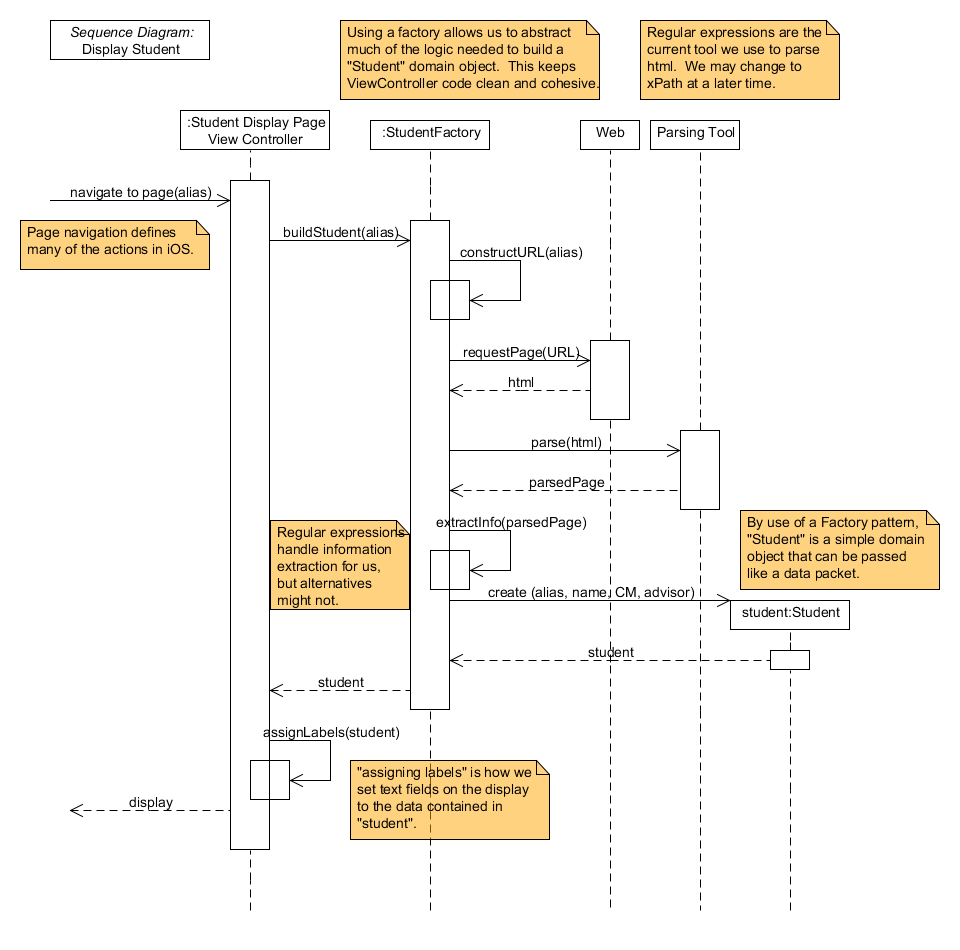
## Change Authentication



## **Display Schedule**



## Display Student



# The Who Done What Table

|  |  |  |  |
| --- | --- | --- | --- |
| Who Done It:  Team Member Names | Section/Part Completed | Task/Comments | # of hours effort |
| Brandon Knight | Coding | Got keychain access for Kerberos authentication working | 8 hours |
| Ann Say | Gang of Four Discussion | Wrote discussions, created partial diagrams. | 5 hours |
| Mark | Compiled Milestone. Updated Test Plan. Created Activity Diagrams. | Organized document, revised prior Test Plan content, created Activity Diagrams. | 7 hours |
| Chris Gropp | SDs, Integration testing, misc. comments | Created and updated the SDs Wrote Integration Testing section, added notes throughout | 6 hours |