|  |
| --- |
| Designer drug database |
| Requirement Document |
| Senior Project, CIS 4911- U01 |
| **Professor: Seyedmasoud Sadjadi Mentor: Dr. Luis Arroyo** |
| **Team Member: Carlos Dominguez** |
| **2/2/2015** |

|  |
| --- |
|  |

# Copyrights and Trademarks Notices

Copyright © by Florida International University – Senior Project – DDD Team

All rights reserved. No part of the Designer Drug Database System Project or its documentation may be reproduced or transmitted in any form or by any means without prior written consent of the DDD Team.

# Abstract

The Designer Drug Database project is designed to provide a way to manage electronically information gathered by researchers at the Forensic Research Institute of FIU as well as allow them to share important information with the world of Academia and the Professionals Community. The system is sophisticated and highly user friendly in this aspect.

This document contains information about the required documentation and project plan for the Designer Drug Database project. The main chapters on this document are the Introduction, Current System (none in this case), Project Plan, and Proposed System Requirements. The introduction presents the problem definition, background, terminology to be used in this document, and Overview of the document. The Current System should talk about the current system, but since this is an application that is being developed from the beginning there is no information about the current system. The Project Plan shows how the project is organized, how the work is broken down, as well as the cost estimate for the project. Finally the Proposed system requirements will contain the necessary UML diagrams to understand the flow and how the system has been implemented.

# Table of contents

Table of Contents

[Copyrights and Trademarks Notices 1](#_Toc418114532)

[Abstract 2](#_Toc418114533)

[Table of contents 3](#_Toc418114534)

[1. Introduction 5](#_Toc418114535)

[1.1 Problem Definition. 5](#_Toc418114536)

[1.2 Scope of system. 5](#_Toc418114537)

[1.3 Terminology - Definitions, acronyms, and abbreviations. 5](#_Toc418114538)

[1.3.1 Definitions 5](#_Toc418114539)

[1.3.2 Acronyms and abbreviations 6](#_Toc418114540)

[1.4 Overview of document 6](#_Toc418114541)

[2. Current System 7](#_Toc418114542)

[3. Project Plan (This deliverable only) 8](#_Toc418114543)

[3.3 Cost Estimate – cost to develop the software system. 8](#_Toc418114544)

[3.1 Project Organization 8](#_Toc418114545)

[3.1.1 Project Personnel Organization 8](#_Toc418114546)

[3.1.2 Hardware and Software Resources 8](#_Toc418114547)

[3.2 Identification of Tasks, Milestones and Deliverables (work breakdown) 9](#_Toc418114548)

[3.3 Cost Estimate – cost to develop the software system. 9](#_Toc418114549)

[4. Proposed System Requirements 10](#_Toc418114550)

[4.1 Functional Requirements – describes high-level functionality 10](#_Toc418114551)

[4.2 Analysis of System Requirements 14](#_Toc418114552)

[4.2.1 Scenarios 14](#_Toc418114553)

[4.2.2 Use case model 14](#_Toc418114554)

[4.2.3 Static model e.g., object diagrams, class diagram 14](#_Toc418114555)

[4.2.4 Dynamic model e.g., sequence diagrams or state machines 15](#_Toc418114556)

[5. Glossary - define terms used in document, especially domain specific terms. 16](#_Toc418114557)

[5.1 Definitions 16](#_Toc418114558)

[5.2 Acronyms and abbreviations 16](#_Toc418114559)

[6. Appendix 17](#_Toc418114560)

[6.1 Appendix A - Complete use cases 17](#_Toc418114561)

[6.2 Appendix B - Use case diagram using UML 26](#_Toc418114562)

[6.3 Appendix C - Static UML diagram 27](#_Toc418114563)

[6.4 Appendix D - Dynamic UML diagrams 37](#_Toc418114564)

[6.5 Appendix E - User Interface designs 47](#_Toc418114565)

[6.6 Appendix F - Diary of meeting and tasks. 55](#_Toc418114566)

[Sprint 1 55](#_Toc418114567)

[Sprint 2 56](#_Toc418114568)

[Sprint 3 56](#_Toc418114569)

[Sprint 4 57](#_Toc418114570)

[Sprint 5 57](#_Toc418114571)

[7. References 58](#_Toc418114572)

# Introduction

The Designer Drug Database System is a system designed to easily add new compounds and manage the information through a centralized user friendly web application. This application was done with a responsive web design to be friendly in phones, tablets and any other device that you access the web application from.

This section will be divided in the following sections:

* 1. Problem Definition.
  2. Scope of system.
  3. Terminology - Definitions, acronyms, and abbreviations.
  4. Overview of document – brief explanation of what to expect in chapters 2 through 6.

## 1.1 Problem Definition.

Dr. Arroyo and the team at the Forensic Research Institute produce several reports and useful information from research they perform at the Institute. One of the challenges they have face is sharing the results that yields from their research. The purpose of this project will be to provide the Institute with a software solution that allows them to share the research result with the world of Academia and the Professionals Community

## 1.2 Scope of system.

The Forensic Research institute at FIU is a prestigious research facility that among all the things they research and do, they are developing a drug database from where any compound listed in it can be easily identified. They have wanted to have a medium to share this information with the rest of Academia and professional community.

## 1.3 Terminology - Definitions, acronyms, and abbreviations.

Following we will have the definitions. Acronyms and abbreviations that are used in this document.

### 1.3.1 Definitions

**Single-page application (SPA):** is a web application or web site that fits on a single web page with the goal of providing a more fluid user experience akin to a desktop application.

**Platform independent framework:** is a framework that allows programmers to create one application that can be seen over different platforms.

**Native system:** is a system that can only be seen in its own target devices, such as apple devices or android devices.

**Administrator:** Registered user with credentials. User has elevated privileges and can give access to other users.

**User:** Any general person who uses the system

### 1.3.2 Acronyms and abbreviations

**Admin:** Administrator

**CSS :** Cascading CIS Style Sheets

**DB:** Database

**FIU:** Florida International University

**GUI:** Graphical User Interface

**HTML:** Hypertext Markup Language

**JS:** JavaScript

**MVC:** Model View Controller

**PHP:** PHP: Hypertext Preprocessor

## 1.4 Overview of document

Throughout this document information about the required documentation and project plan for the Designer Drug Database project will be provided. The following chapters, from two to six, on are the Current System (none in this case), Project Plan, and Proposed System Requirements. The Current System should talk about the current system, but since this is an application that is being developed from the beginning there is no information about the current system. The Project Plan shows how the project is organized, how the work is broken down, as well as the cost estimate for the project. Finally the Proposed system requirements will contain the necessary UML diagrams to understand the flow and how the system has been implemented.

# **C**urrent System

In my case there is no current system. The system will be implemented for the first time during this semester

# Project Plan (This deliverable only)

In order to ensure a successful project I have created this initial plan with deadlines which we intend to stick by throughout the semester. This will allow us to see whether or not we are on track with our requirements.

This section divides as follows:

* 1. Project Organization
     1. Project Personnel Organization
     2. Hardware and Software Resources
  2. Identification of Tasks, Milestones and Deliverables (work breakdown)

## 3.3 Cost Estimate – cost to develop the software system.

## 3.1 Project Organization

### 3.1.1 Project Personnel Organization

In this project there is only one team member, me (**Carlos Dominguez)**, and I am the Developer, Tester, System Designer, GUI Designer, Database Manager. On the other hand, the mentor is the product owner that is contacted when question about the product itself are needed.

Moreover, every week on Friday at 1:30pm there is a meeting in person, for the developer to report what was done over the course of that week.

Lastly, every weekday the developer will upload and update the project and its status.

### 3.1.2 Hardware and Software Resources

The resources for this project are:

* GitHub repository, for code development.
* Virtual machine, to host the final product
* My computer, to develop test and upload changes
* PHP, mysql, javascript, angular, bootstrap, and html

## 3.2 Identification of Tasks, Milestones and Deliverables (work breakdown)

|  |  |  |
| --- | --- | --- |
|  | Tasks | Task Dependencies |
| **1** | Problem Definition |  |
| **2** | Obtain High Level User Requirements | 1 |
| **3** | Identify Alternatives Solutions | 2 |
| **4** | Determine Solutions | 3 |
| **5** | Requirement Analysis and Elicitation | 1 |
|  | **Milestone: System Analysis and Implementation** |  |
| **6** | Implement Database Structural Changes | 5 |
| **7** | Populate Database with Data | 5, 6 |
| **8** | Modify System Architecture | 5 |
| **9** | Begin UI Design Modification for Forms | 7, 8 |
| **10** | Begin UI Design for Flowchart | 7, 8 |
| **11** | Implement Functions | 7, 9, 10 |
| **12** | Initial Testing | 7, 11 |
|  | **Milestone: Implemented System Functionality** |  |
| **13** | Finalize System Implementation | 12 |
| **14** | Functional Testing | 12 |
| **15** | Evaluate Test Results | 13, 14 |
| **16** | Complete Final Documentation and Presentation | 15 |
|  | **Milestone: System Complete** | 16 |

## 3.3 Cost Estimate – cost to develop the software system.

The cost of this system will be $0.00.

|  |  |  |
| --- | --- | --- |
| Item | Description | Cost |
| Human Resources | Team member working on the project during the entire development process. | $0.00 |
| Hardware Tools | Laptops | $0.00 |
| FIU Computers | $0.00 |
| Software Tools | PHPAdmin | $0.00 |
| FIU virtual matchine | $0.00 |
| Notepad++ | $0.00 |
| Total cost | | $0.00 |

# Proposed System Requirements

This chapter defines the functional requirement to be implemented in version 1 of the Designer Drug Database project. These requirements are described in terms of functional requirements and their respective nonfunctional requirements.

The section will be divided as follows:

* 1. Functional Requirements – describes high-level functionality
  2. Analysis of System Requirements
  3. Scenarios
  4. Static model e.g., object diagrams, class diagram
  5. 4.5 Dynamic model e.g., sequence diagrams or state machines

## 4.1 Functional Requirements – describes high-level functionality

Below are the functional requirements for Designer Drug Database System.

The system shall allow…

**Admins and lab operator must be able to login with their credentials**

* **Usability**: The form to be filled must be easy to follow when filling up.
* **Reliability**: System must work flawlessly 99% of the time.
* **Performance**: Data must be saved with the time according to the size of file being imported.
* **Supportability**: This functionality must work for any browser.

**Admins and lab Operator must be able to add new compounds**

* **Usability**: The form to be filled must be easy to follow when filling up.
* **Reliability**: System must work flawlessly 99% of the time.
* **Performance**: Data must be save within 3s seconds.
* **Supportability**: This functionality must work for any browser.

**Admins and lab Operator must be able to edit compounds**

* **Usability**: The form to be modified must be easy to follow when changing the data.
* **Reliability**: System must work flawlessly 99% of the time.
* **Performance**: Data must be save within 3s seconds.
* **Supportability**: This functionality must work for any browser.

**Admins and lab Operator must be able to delete compounds**

* **Usability**: The compound needs to be deleted from all places
* **Reliability**: System must work flawlessly 99% of the time.
* **Performance**: Data must be save within 3s seconds.
* **Supportability**: This functionality must work for any browser.

**Admins must be able to import compounds**

* **Usability**: The compound needs to be imported from a csv file and jpg pictures.
* **Reliability**: System must work flawlessly 99% of the time.
* **Performance**: Data must be saved with the time according to the size of file being imported.
* **Supportability**: This functionality must work for any browser.

**Admins and Lab Operators must be able to modify their name in the system**

* **Usability**: The form to be filled must be easy to follow when filling up.
* **Reliability**: System must work flawlessly 99% of the time.
* **Performance**: Data must be save within 3s seconds.
* **Supportability**: This functionality must work for any browser.

**Admins and Lab Operators must be able to modify their password in the system**

* **Usability**: The form to be filled must be easy to follow when filling up.
* **Reliability**: System must work flawlessly 99% of the time.
* **Performance**: Data must be save within 3s seconds.
* **Supportability**: This functionality must work for any browser.

**Admins must be able to create new users in the system**

* **Usability**: The form to be filled must be easy to follow when filling up.
* **Reliability**: System must work flawlessly 99% of the time.
* **Performance**: Data must be save within 3s seconds.
* **Supportability**: This functionality must work for any browser.

**Admins must be able to delete users of the system**

* **Usability**: The form to be filled must be easy to follow when filling up.
* **Reliability**: System must work flawlessly 99% of the time.
* **Performance**: Data must be save within 3s seconds.
* **Supportability**: This functionality must work for any browser.

**Admins must be able to modify user roles of the system**

* **Usability**: The form to be filled must be easy to follow when filling up.
* **Reliability**: System must work flawlessly 99% of the time.
* **Performance**: Data must be save within 3s seconds.
* **Supportability**: This functionality must work for any browser.

**Admins and Lab Operators must be able to see summary of the system in a dashboard.**

* **Usability**: The information needs to be clear
* **Reliability**: System must work flawlessly 99% of the time.
* **Performance**: Data must be save within 3s seconds.
* **Supportability**: This functionality must work for any browser.

**Admins and Lab Operators must be able to create new compound classes in the system**

* **Usability**: The form to be filled must be easy to follow when filling up.
* **Reliability**: System must work flawlessly 99% of the time.
* **Performance**: Data must be save within 3s seconds.
* **Supportability**: This functionality must work for any browser.

**Admins and Lab Operators must be able to modify compound classes in the system**

* **Usability**: The form to be filled must be easy to follow when filling up.
* **Reliability**: System must work flawlessly 99% of the time.
* **Performance**: Data must be save within 3s seconds.
* **Supportability**: This functionality must work for any browser.

**Admins and Lab Operators must be able to delete empty compound classes (no compounds inside).**

* **Usability**: The steps must be easy to follow.
* **Reliability**: System must work flawlessly 99% of the time.
* **Performance**: Data must be save within 3s seconds.
* **Supportability**: This functionality must work for any browser.

**Admins and Lab Operators must be able to read contact emails sent to system**

* **Usability**: The form to be filled must be easy to follow when filling up.
* **Reliability**: System must work flawlessly 99% of the time.
* **Performance**: Data must be save within 3s seconds.
* **Supportability**: This functionality must work for any browser.

**Admins and Lab Operators must be able to delete contact emails sent to system**

* **Usability**: The form to be filled must be easy to follow when filling up.
* **Reliability**: System must work flawlessly 99% of the time.
* **Performance**: Data must be save within 3s seconds.
* **Supportability**: This functionality must work for any browser.

**Admins and Lab Operators must be able to reply contact emails sent to system**

* **Usability**: The form to be filled must be easy to follow when filling up.
* **Reliability**: System must work flawlessly 99% of the time.
* **Performance**: Data must be save within 3s seconds.
* **Supportability**: This functionality must work for any browser.

**Any user of the system must be able to download the information of an entire compound class.**

* **Usability**: The steps must be easy.
* **Reliability**: System must work flawlessly 99% of the time.
* **Performance**: Data must be save within 3s seconds.
* **Supportability**: This functionality must work for any browser.

**Any user of the system must be able to search and visualize the information of a compound.**

* **Usability**: The steps must be easy.
* **Reliability**: System must work flawlessly 99% of the time.
* **Performance**: Data must be save within 3s seconds.
* **Supportability**: This functionality must work for any browser.

## 4.2 Analysis of System Requirements

This section contains scenarios for some system requirements as well as the different models related to the requirements such as use case model, static model, and dynamic model.

### 4.2.1 Scenarios

This section contains the scenarios to be developed for this first version of this solution.

#### Managing Compounds

Any administrator or lab operator after having signed in in the system can go to a compound management area where he/she can add, modify or delete existing compounds of the system, as well as any information regarding compounds.

#### Managing Users

Admins after they have logged in they have to be able to create new users, delete users and edit user type, except for their own.

#### Change User Information

Admins and Lab Operators have to be able to change their account information such as their name and password.

#### Manage Contact messages.

Admins and lab operators have to be able to see message of people contacting the application, as well as reply to them and delete them

#### Show the accepted information

Any user of the system, including guest, have to be able to search for a compound and see the information about that compound.

### 4.2.2 Use case model

Appendix B contains the Use Case diagrams with all use cases pertaining to the Designer Drug Database System. This diagram clearly shows the different actors that will be interacting with the system. These are admin, lab Operator, and the guest of the system. Each of these actors is linked with his/her respective capabilities within the system.

### 4.2.3 Static model e.g., object diagrams, class diagram

Appendix C shows some of the different class diagrams for the Designer Drug Database System. These diagrams show the relationship between classes. The class diagrams provided in Appendix C are minimal due to space. On top of each class diagram, the name of diagram can be seen.

### 4.2.4 Dynamic model e.g., sequence diagrams or state machines

Appendix D shows the different sequence diagrams for the Designer Drug Database System. These diagrams show the interaction among the objects as they are planned to occur when the system is in execution. On top of each sequence diagrams, its use case id is displayed as well as its name.

# Glossary - define terms used in document, especially domain specific terms.

Following we will have the definitions. Acronyms and abbreviations that are used in this document.

### 5.1 Definitions

**Single-page application (SPA):** is a web application or web site that fits on a single web page with the goal of providing a more fluid user experience akin to a desktop application.

**Platform independent framework:** is a framework that allows programmers to create one application that can be seen over different platforms.

**Native system:** is a system that can only be seen in its own target devices, such as apple devices or android devices.

**Administrator:** Registered user with credentials. User has elevated privileges and can give access to other users.

**User:** Any general person who uses the system

### 5.2 Acronyms and abbreviations

**Admin:** Administrator

**CSS :** Cascading CIS Style Sheets

**DB:** Database

**FIU:** Florida International University

**GUI:** Graphical User Interface

**HTML:** Hypertext Markup Language

**JS:** JavaScript

**MVC:** Model View Controller

**PHP**: Hypertext Preprocessor

# Appendix

## 6.1 Appendix A - Complete use cases

|  |
| --- |
| Name: User Login |
| Participating actor: a non-logged in user |
| Entry condition:   * User is in login page. * User is not logged in. |
| Exit condition:   * User is on the account main page |
| Event flow:   1. User clicks the Login button 2. System loads login page 3. User enters email and password on their respective boxes and click submit button 4. System loads the account main page |
| Alternative Exit condition: System says that login and password are incorrect. |

|  |
| --- |
| Name: Modify Name of User |
| Participating actor: a logged in user. |
| Entry condition:   * User is logged in. * User is in Account Page |
| Exit condition:   * Confirmation name of user is changed is shown. |
| Event flow:   1. User clicks on its name 2. System shows a view to change the name of the user 3. User replaces old name with new name and clicks submit 4. System changes the name of the user and shows a confirmation. |

|  |
| --- |
| Name: Modify user password |
| Participating actor: a logged in user. |
| Entry condition:   * User is logged in. * User is in Account Page |
| Exit condition:   * Confirmation that password has been changed |
| Event flow:   1. User clicks on the password link 2. System shows a view to change the password 3. User enters old, new and confirmation of new password. 4. System changes the password of the user and shows a confirmation. |

|  |
| --- |
| Name: Search For Compound |
| Participating actor: any user of the system |
| Entry condition:   * User is in main page or home page |
| Exit condition:   * User found the data of the search compound |
| Event flow:   1. User enters the email or formula of compound in the search criteria box and click the search button. 2. Systems show the information of the compound |
| Alternative Exit condition: System does not find a compound and show a message saying the search was unsuccessful. |

|  |
| --- |
| Name: Download Compound Class |
| Participating actor: any user of the system |
| Entry condition:   * User is in main page or home page |
| Exit condition:   * User saved to his computer the desired data |
| Event flow:   1. User searches for the class of the compound he desires to download in the class download list and clicks on it 2. A window is pop up to ask user where he wants to save the information 3. User pick location where information will be saved and click save 4. System downloads the file |

|  |
| --- |
| Name: add new Compound |
| Participating actor: either an admin or a labOP |
| Entry condition:   * User is logged in the system * User is in compound page. |
| Exit condition: Compound data is added in the system |
| Event flow:   1. User click on the add compound button 2. System shows add compound form page 3. User selects the picture of the compound, fill the fields: name, formula, other name, mass, fragments, retention time, precursor, CAS. User add the corresponding transitions and click submit all button. 4. System confirm that compound has been saved |

|  |
| --- |
| Name: delete Compound |
| Participating actor: either an admin or a labOP |
| Entry condition:   * User is logged in the system * User is in the compound page. |
| Exit condition: Compound data is deleted from the system |
| Event flow:   1. User enters the name or formula in the search box and clicks the search button. 2. System searches for compound and goes to modify compound view 3. User clicks delete compound button. 4. System deletes the compound and shows a confirmation |

|  |
| --- |
| Name: edit Compound |
| Participating actor: either an admin or a labOP |
| Entry condition:   * User is logged in the system * User is in their account home page. |
| Exit condition: Compound data is updated in the system |
| Event flow:   1. User enters the name or formula in the search box and clicks the search button. 2. System searches for compound and goes to modify compound view. 3. User edit any of the information on the modify compound view and click the update button. 4. System updates the compound and shows a confirmation |

|  |
| --- |
| Name: import Compound |
| Participating actor: either an admin or a labOP |
| Entry condition:   * User is logged in the system * User is in the compound page. * File to be imported has a csv extension and has the following rows: Compound Name, Compound Class, Formula, Mass, Precursor, Product, No. of Transitions, Frag, CE, Abundance, Relative Ion Intensity, RT (Zorbax), CAS |
| Exit condition: Compounds data in csv file and pictures has been added to the system |
| Event flow:   1. User click in the import compound button 2. System shows the import page 3. User clicks the choose file to choose location of csv file from import csv panel 4. A pop up menu will show for user to choose the file he wishes to import. 5. User selects the csv file and clicks submit button. 6. System saves the compounds information and shows a confirmation. 7. User click the choose file to pick the location of the folder where the compound pictures are at, from import picture panel. 8. A pop up menu will show for user to choose the folder where pictures of compounds are 9. User selects all the pictures to be upload and clicks the submit button 10. System upload all the pictures associated with the compounds uploaded previously. |

|  |
| --- |
| Name: Add New User |
| Participating actor: admin user |
| Entry condition:   * User is logged in the system * User is in the user management page. |
| Exit condition: User invite is sent and confirmation is shown |
| Event flow:   1. User enters the email of the user to be invited and clicks the invite button. 2. System sends an email to the user and shows a confirmation. |

|  |
| --- |
| Name: edit User Rights |
| Participating actor: an admin |
| Entry condition:   * User is logged in the system. * User is in the user management page. * User is not trying to changes rights to itself. |
| Exit condition: User access right is updated |
| Event flow:   1. User enters the email of the user to be searched and clicks the search button. 2. System shows a view with the searched user information. 3. User picks the downgrade or upgrade option 4. System updates user rights and show confirmation. |

|  |
| --- |
| Name: delete User |
| Participating actor: an admin |
| Entry condition:   * User is logged in the system * User is in the user Management page. * user is not trying to delete itself |
| Exit condition: User is deleted |
| Event flow:   1. User enters the email of the user to be searched and clicks the search button. 2. System shows a view with the user information. 3. User clicks the delete button 4. System deletes the user and shows confirmation. |

|  |
| --- |
| Name: Add New Compound Class |
| Participating actor: admin or LabOP user |
| Entry condition:   * User is logged in the system * User is in the compound class page. |
| Exit condition: Compound Class is added and confirmation is shown |
| Event flow:   1. User Clicks compound classes button. 2. System shows the Compound Classes view 3. User clicks button add class 4. System shows view to add new class 5. User enters the class name and clicks submit. 6. System saves the new class and shows confirmation. |

|  |
| --- |
| Name: Edit Compound Class Name |
| Participating actor: admin or LabOP user |
| Entry condition:   * User is logged in the system * User is in the compound class page. |
| Exit condition: Compound Class name is updated and confirmation is shown |
| Event flow:   1. User Clicks compound classes button. 2. System shows the Compound Classes view 3. User clicks on the class that is going to be updated 4. System shows view where the class name can be updated 5. User updates the class name and clicks submit. 6. System updates the class name and shows confirmation. |

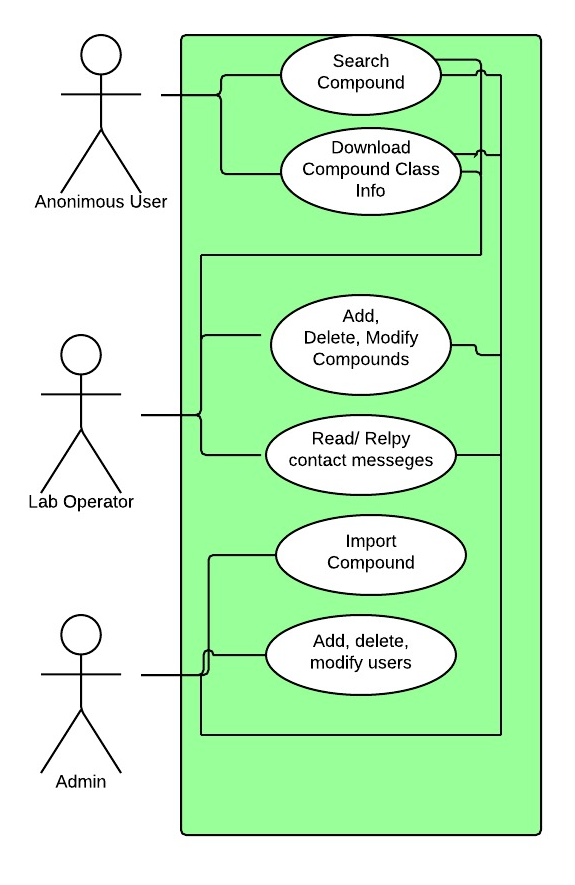
|  |
| --- |
| Name: delete Compound Class |
| Participating actor: admin or LabOP user |
| Entry condition:   * User is logged in the system * User is in the compound class page. * Compound class to be deleted has no compounds associated with it |
| Exit condition: Compound Class is deleted and confirmation is shown |
| Event flow:   1. User Clicks compound classes button. 2. System shows the Compound Classes view 3. User clicks on the class that is going to be deleted. 4. System shows view with the class name to be deleted 5. User clicks the delete button. 6. System deletes the class and shows confirmation. |

|  |
| --- |
| Name: Add Read contact email |
| Participating actor: admin or LabOP user |
| Entry condition:   * User is logged in the system * User is in the Messages page. |
| Exit condition: Message content is shown to the user |
| Event flow:   1. User Clicks on a message to be read from list of messages 2. System shows the message content. |

|  |
| --- |
| Name: reply contact message |
| Participating actor: admin or LabOP user |
| Entry condition:   * User is logged in the system * User is in the Messages page. |
| Exit condition: Message reply is sent and confirmation is shown. |
| Event flow:   1. User Clicks on a message to be replied from list of messages 2. System shows the message content. 3. User clicks on the reply button 4. System shows a view to reply the message 5. User enters the text to be replied and click reply button 6. System replies the message and show a confirmation |

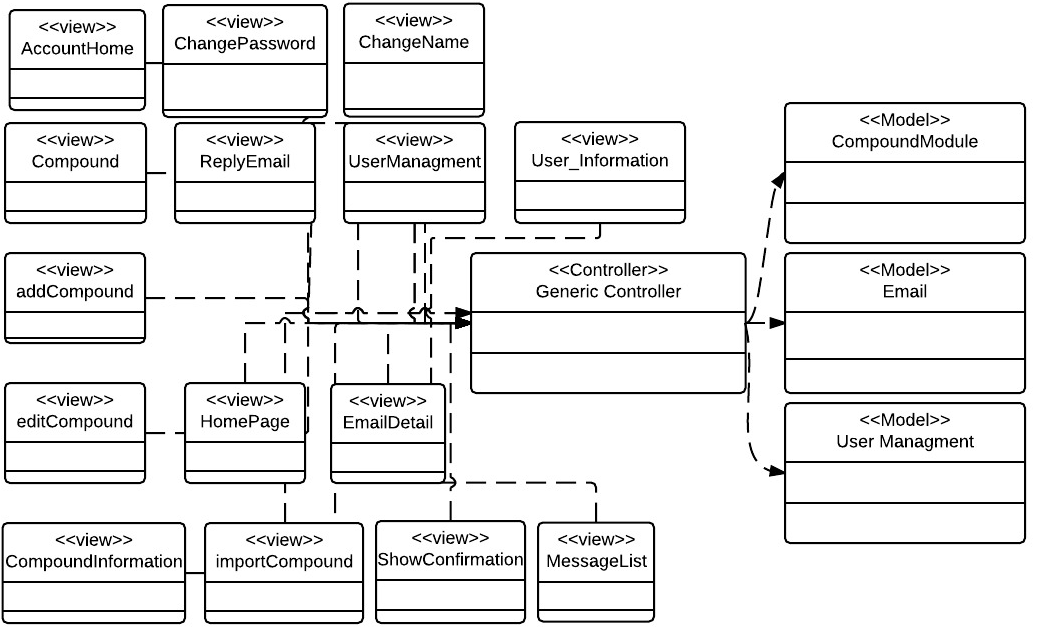
|  |
| --- |
| Name: delete message |
| Participating actor: admin or LabOP user |
| Entry condition:   * User is logged in the system * User is in the Messages page. |
| Exit condition: Message content is shown to the user |
| Event flow:   1. User Clicks on a message to be deleted from list of messages 2. System shows the message content. 3. User clicks the delete button 4. System deletes the message and shows confirmation message. |

## 6.2 Appendix B - Use case diagram using UML

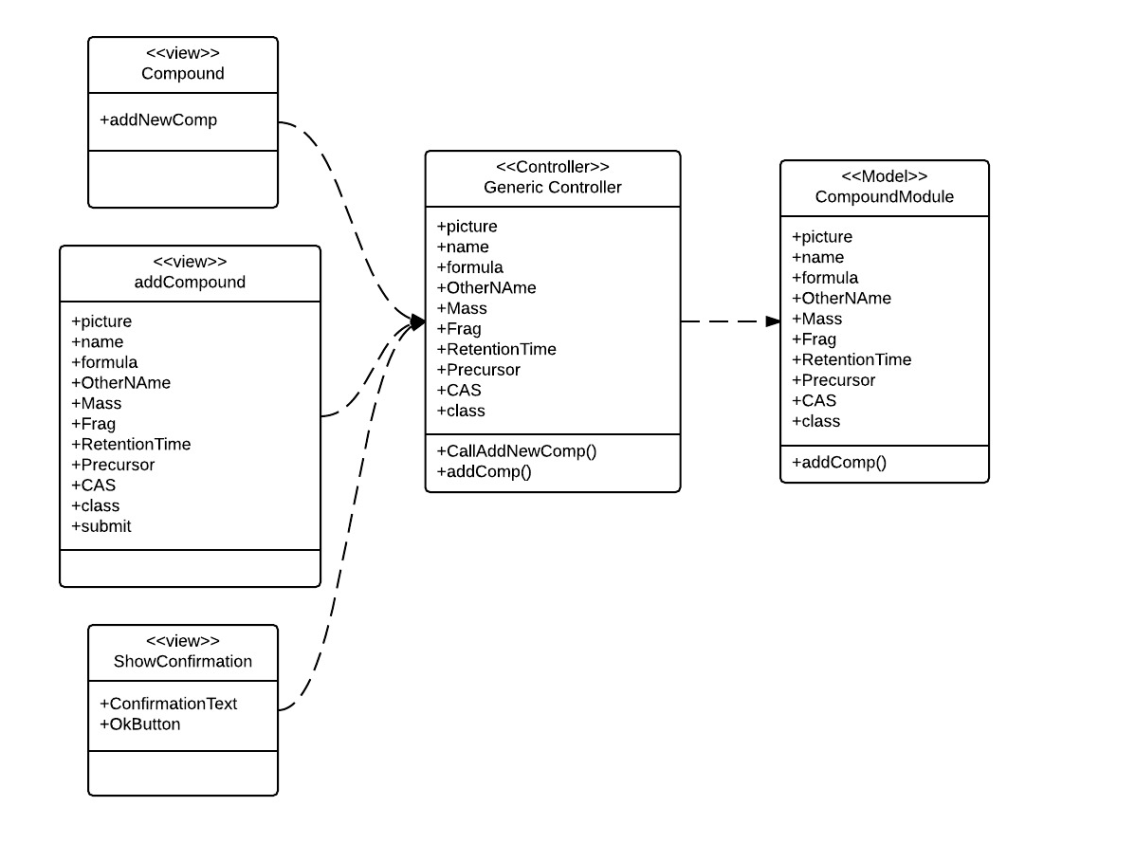


## Appendix C - Static UML diagram

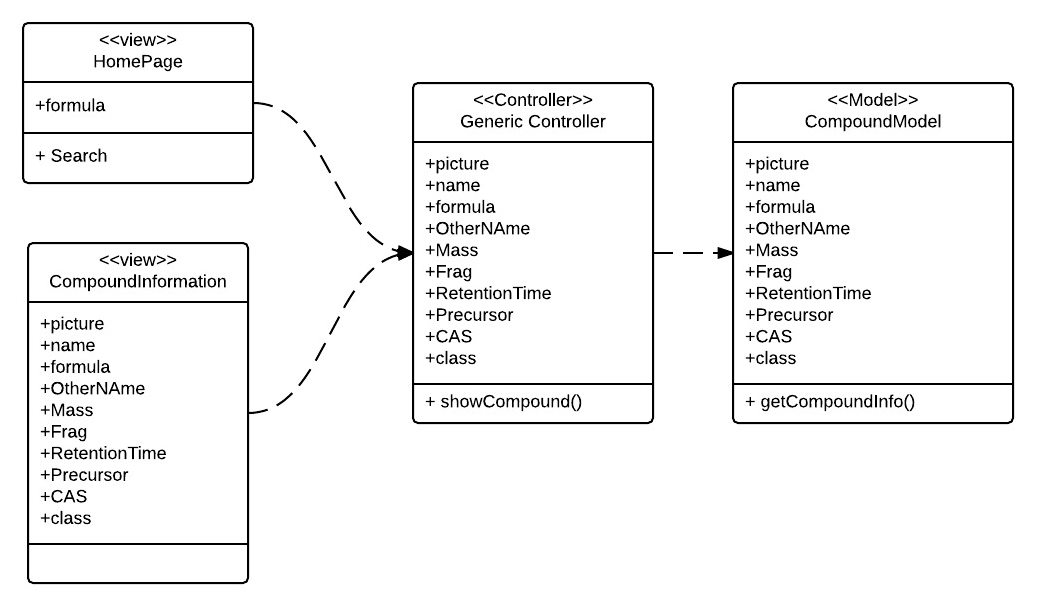
Minified Class Diagram



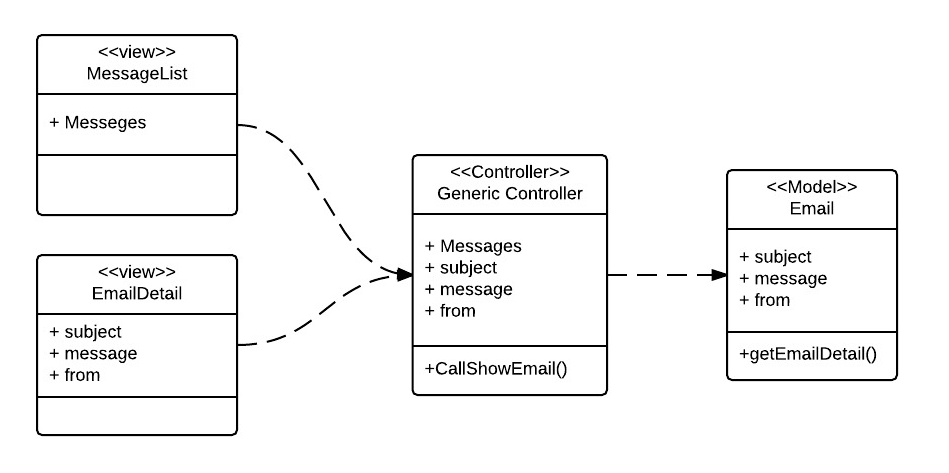
Add Compound Class Diagram



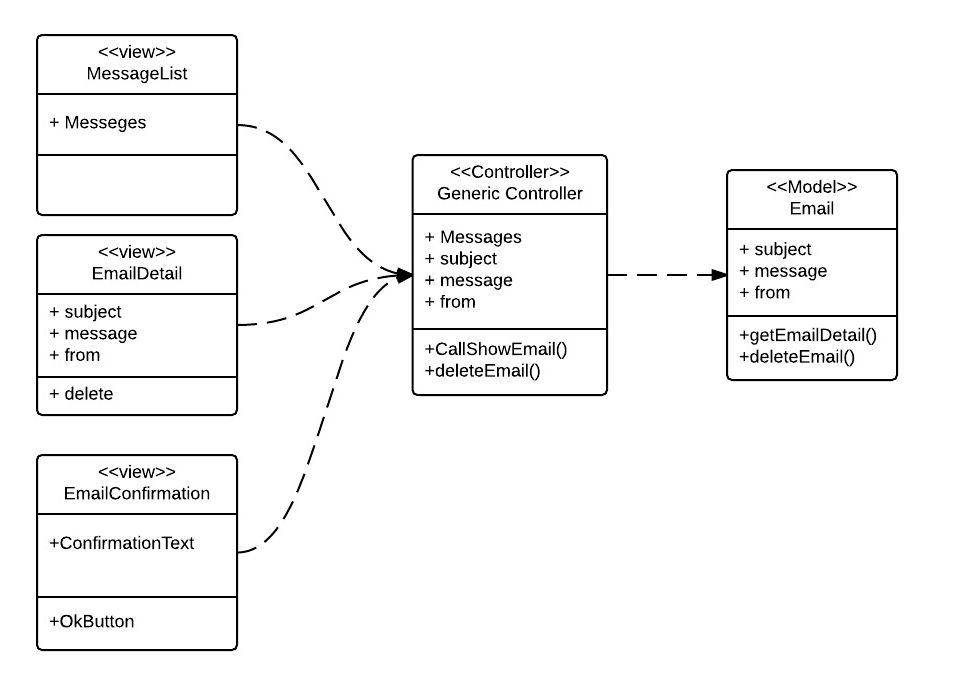
Show Compound Class Diagram



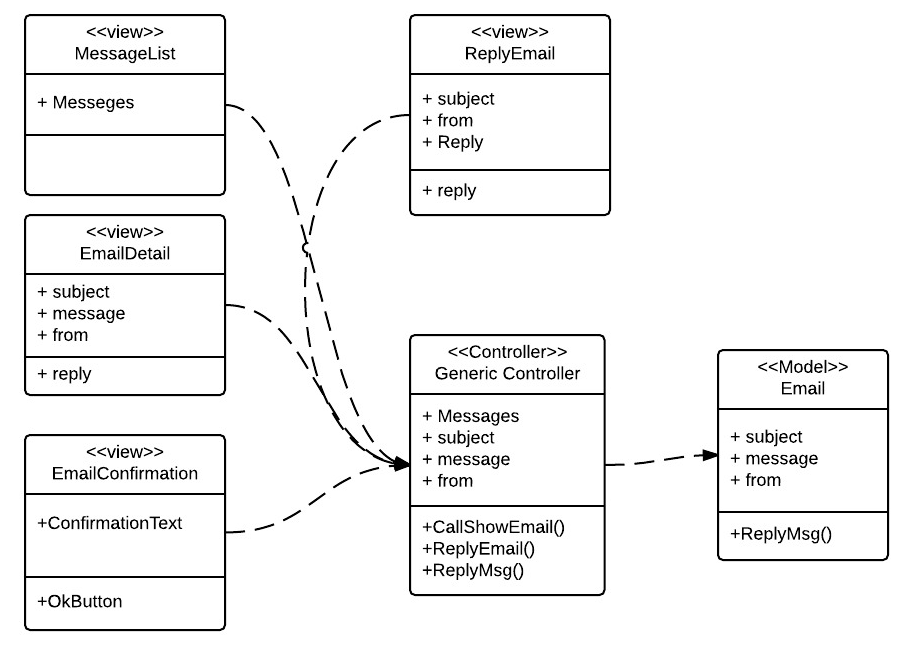
Read Email Class Diagram



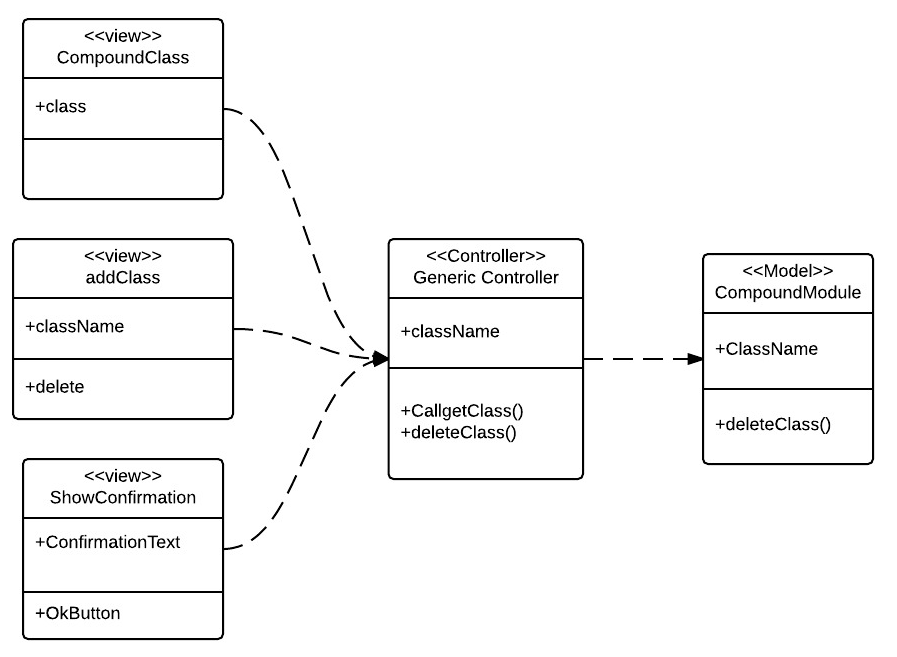
Delete Email Class Diagram



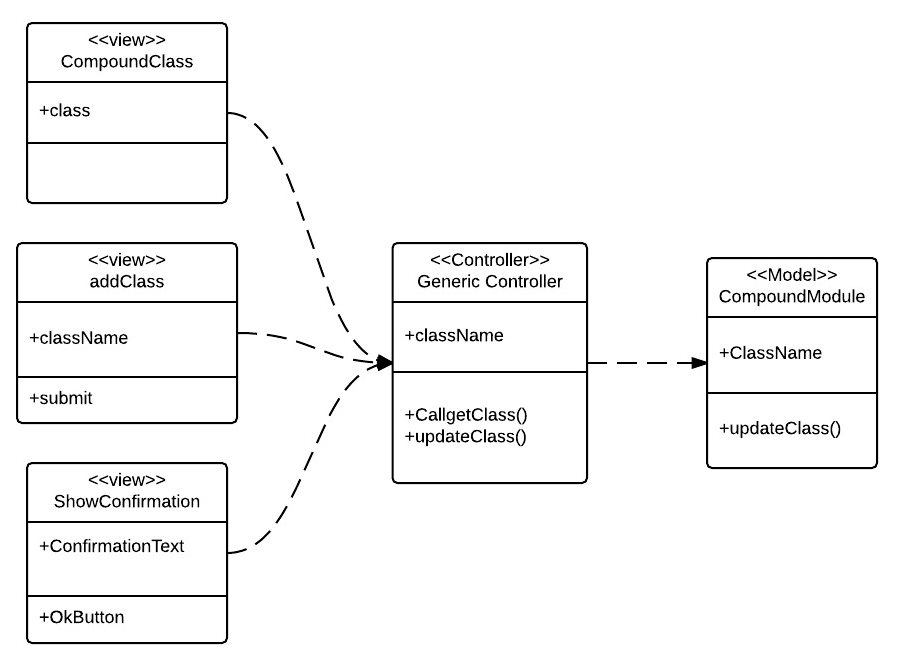
Reply Email Class Diagram



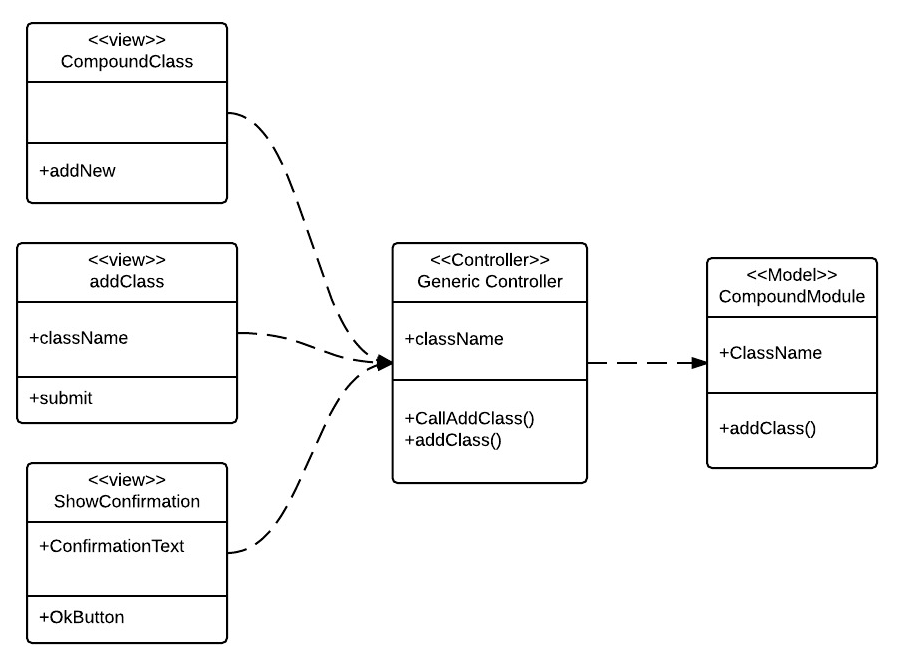
Delete Compound Classes Class Diagram



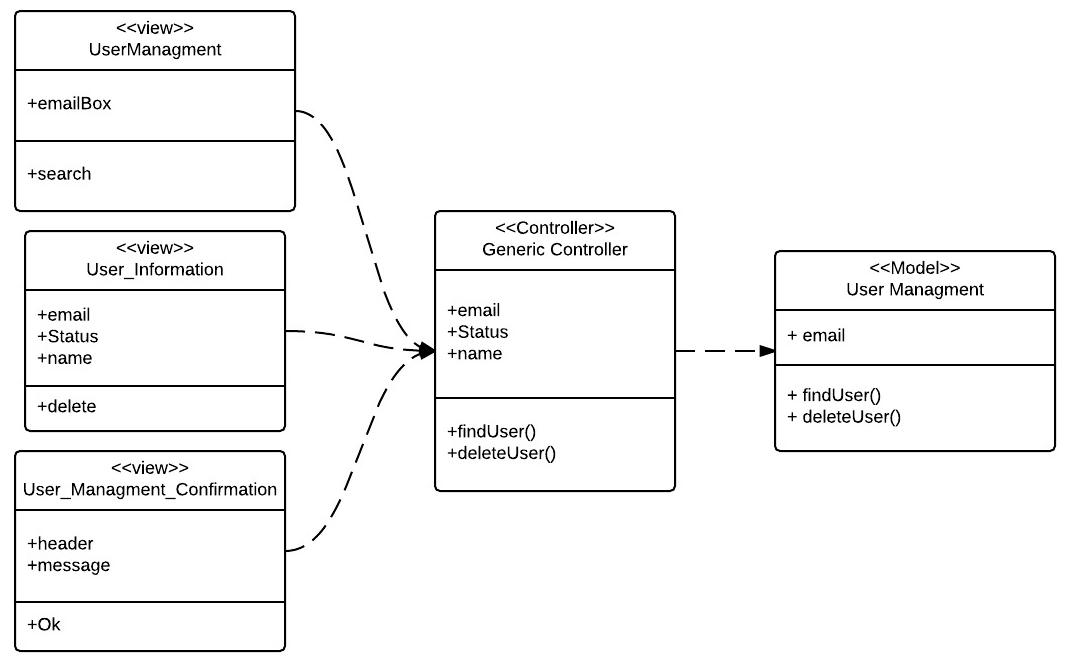
Edit Compound Classes Class Diagram



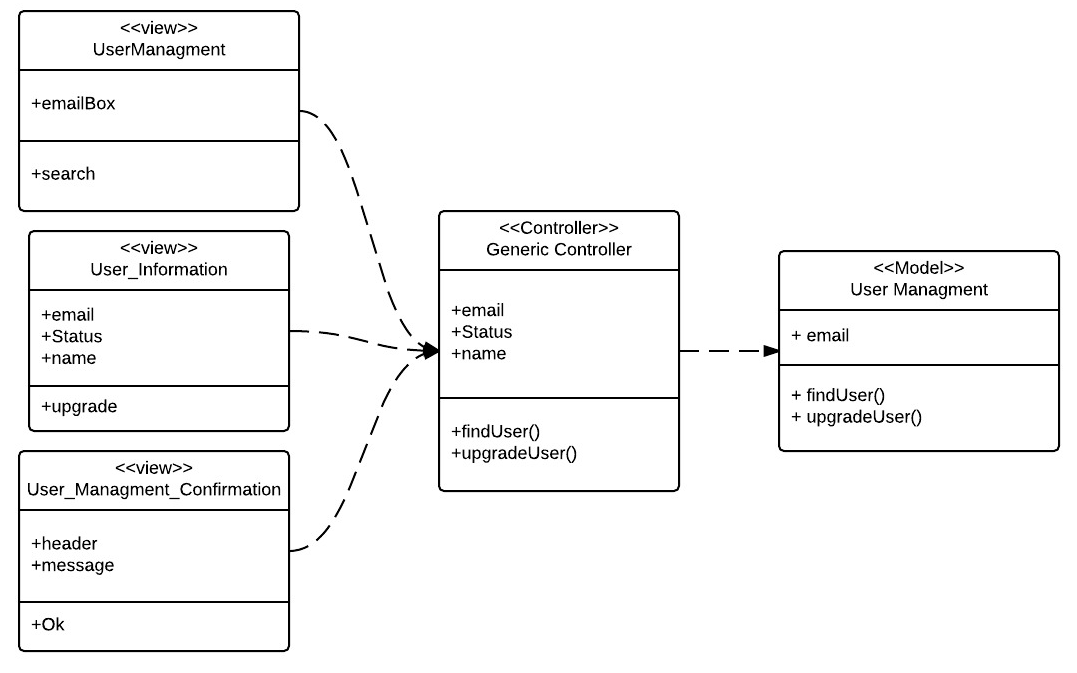
Add Compound Classes Class Diagram



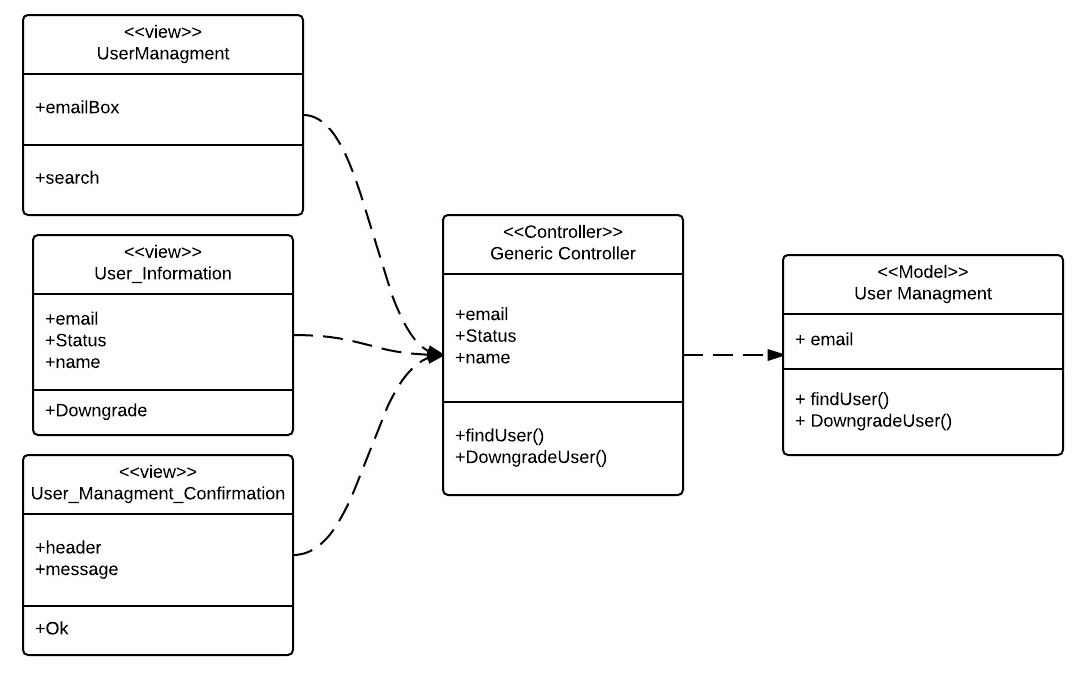
Delete User Class Diagram



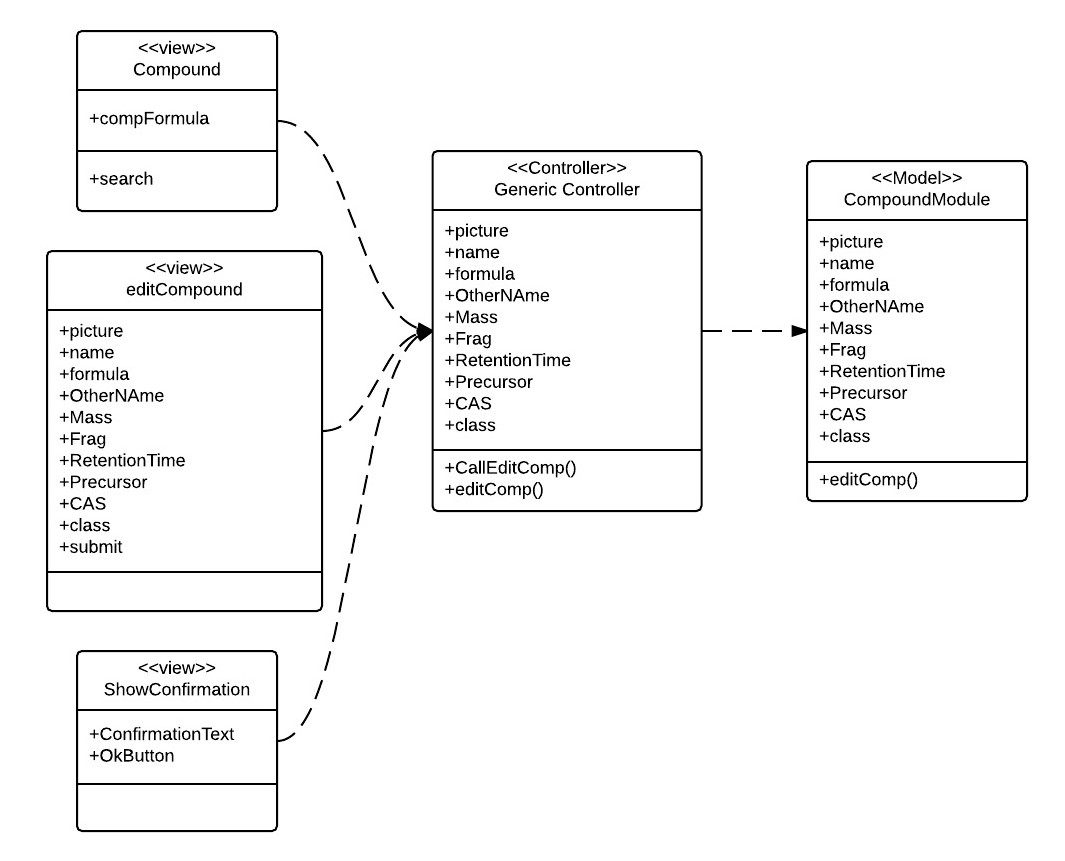
Upgrade User Role Class Diagram



Downgrade User Role Class Diagram



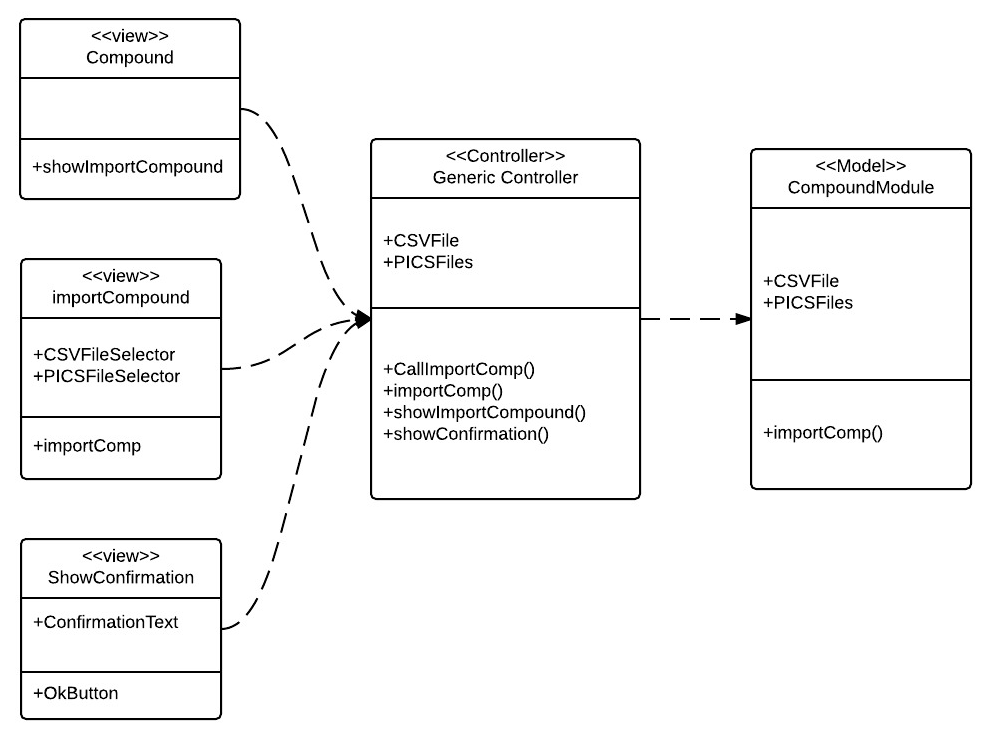
Edit Compound Class Diagram



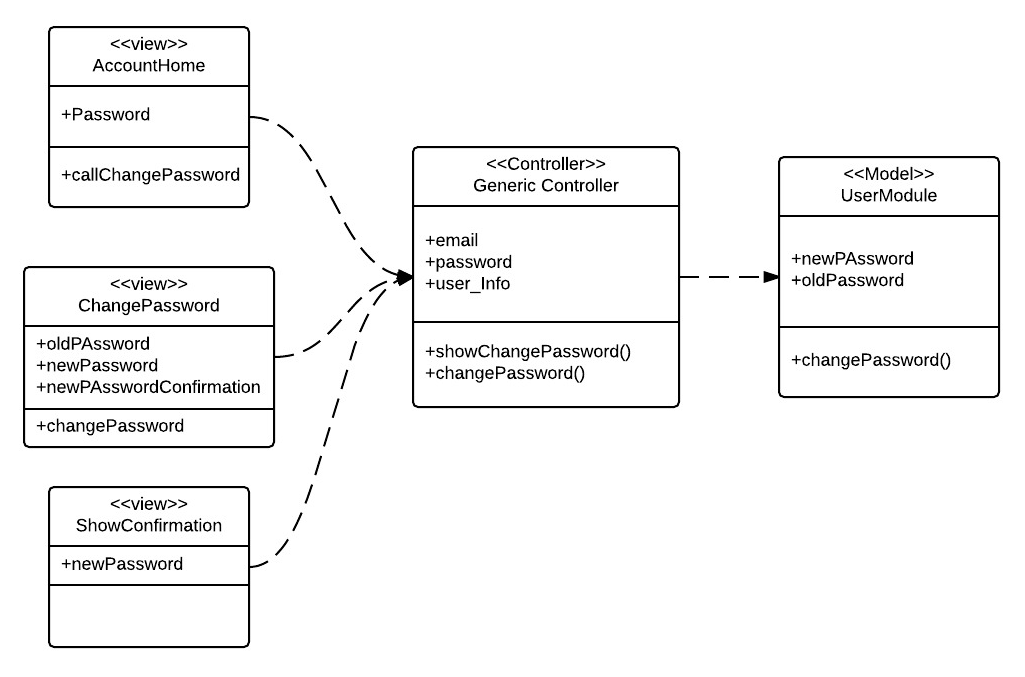
Delete Compound Class Diagram



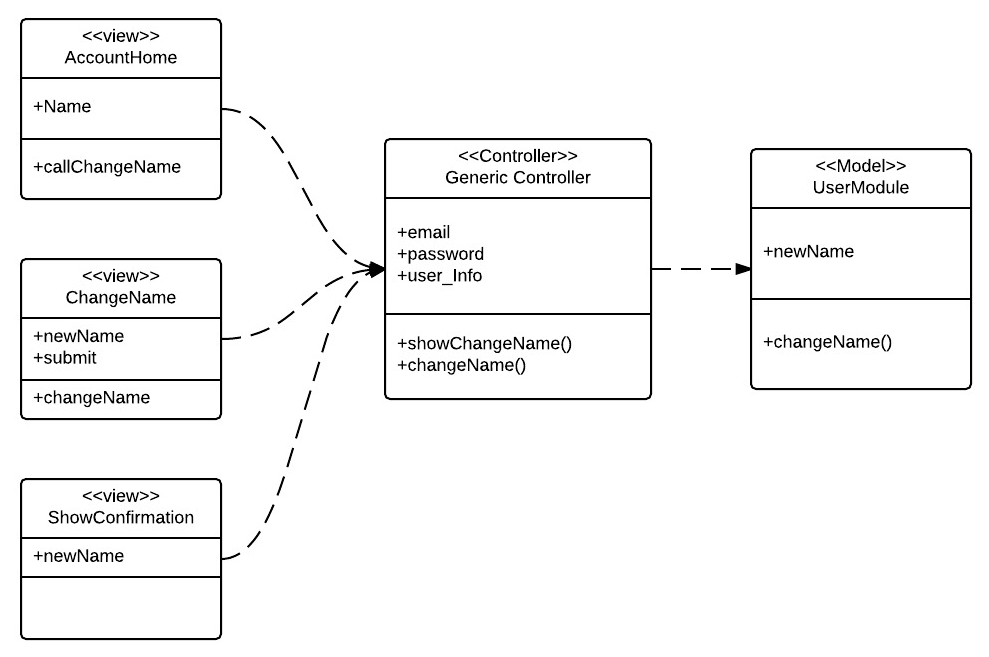
Import Compound Class Diagram



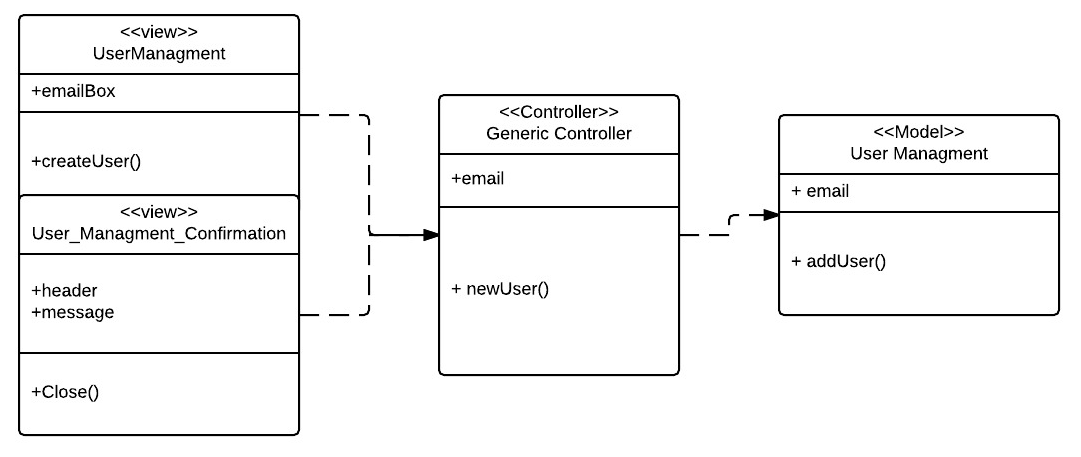
Change User Password Class Diagram



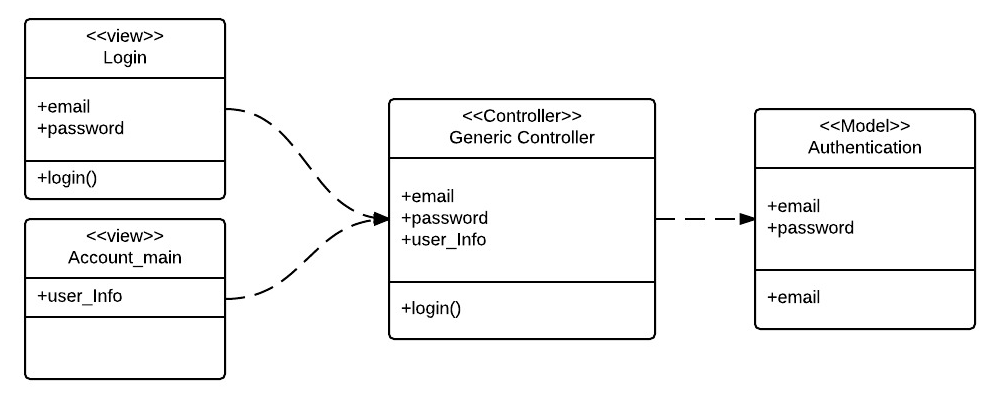
Change User Name Class Diagram



Invite User Class Diagram

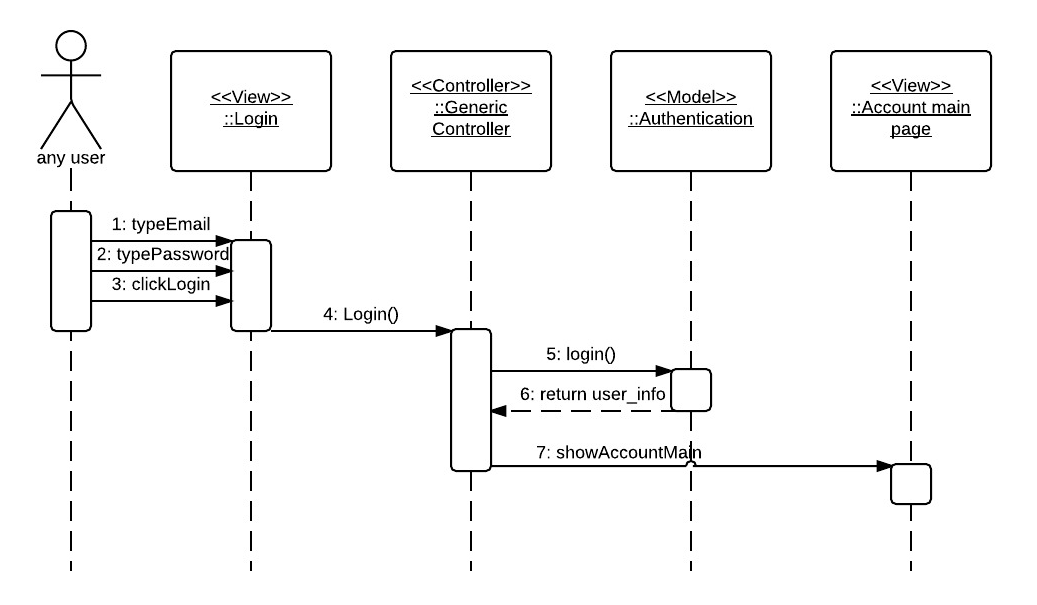


User Login Class Diagram

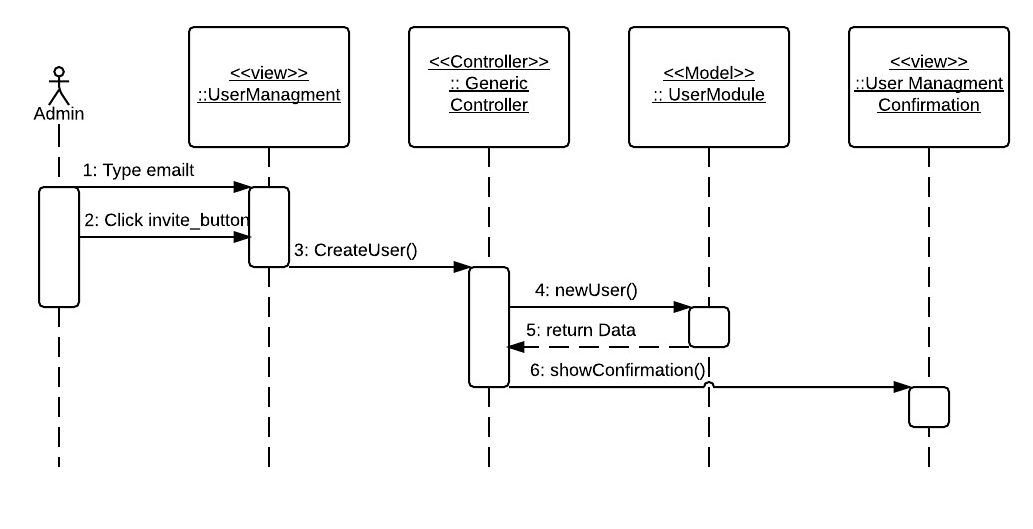


## 6.4 Appendix D - Dynamic UML diagrams

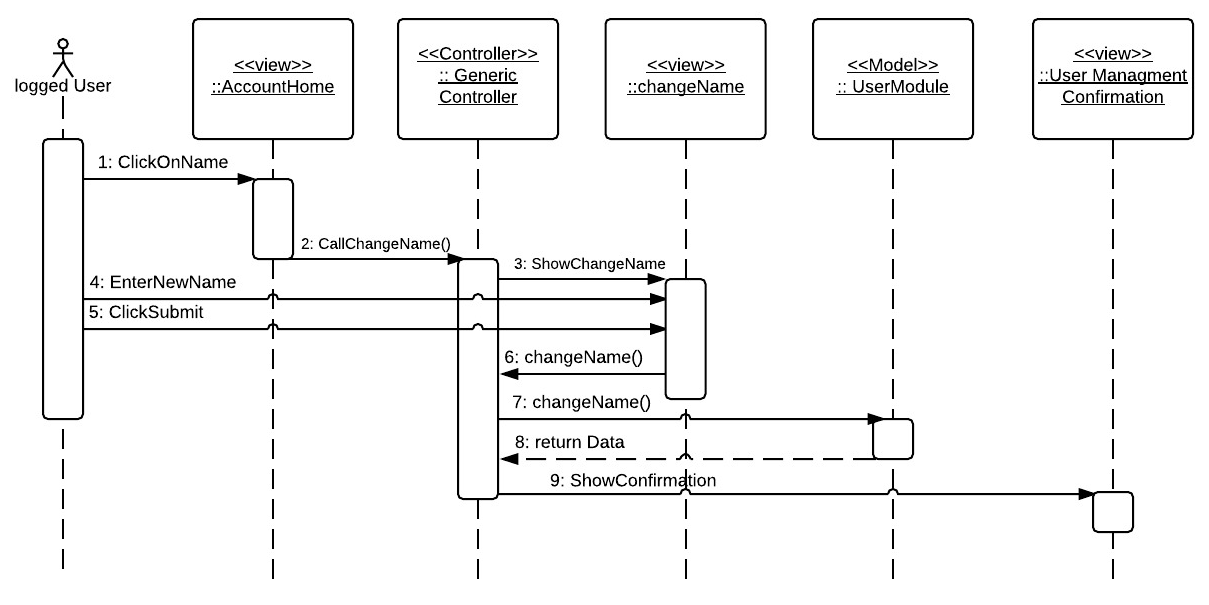
User Login Sequence Diagram



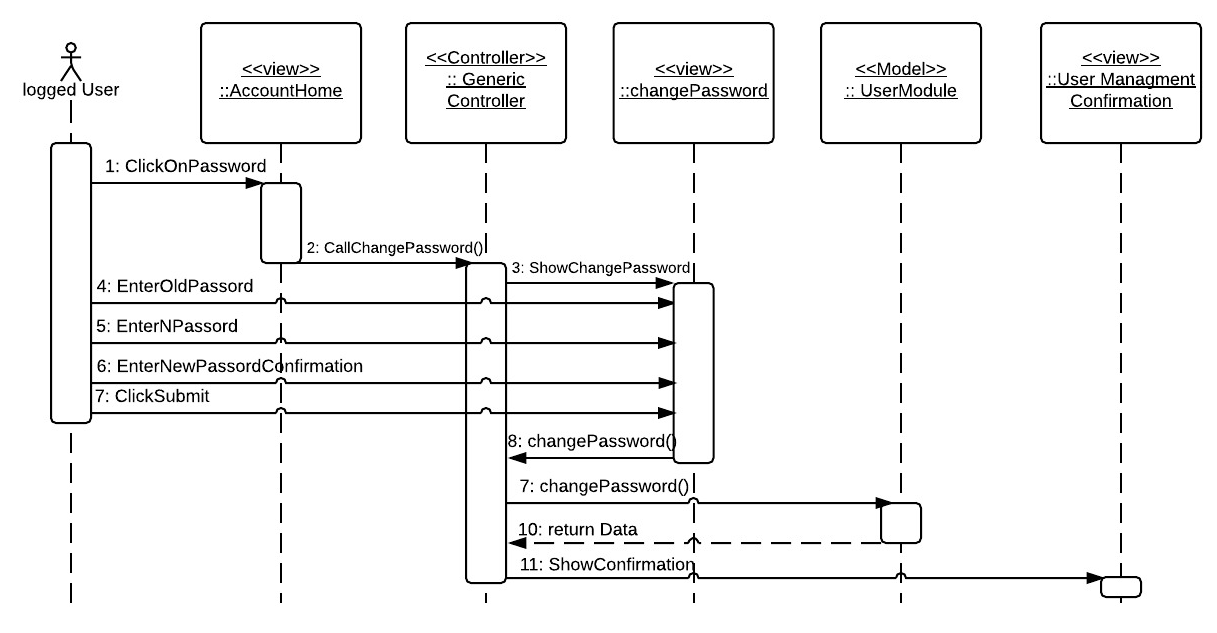
Invite User Sequence Diagram



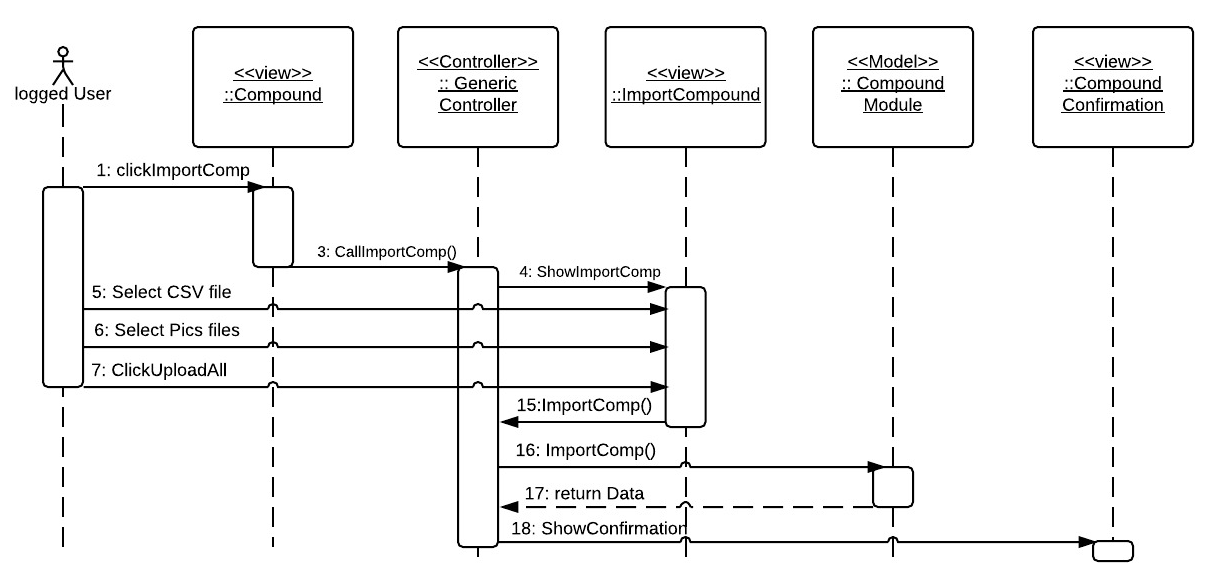
Change User Name Sequence Diagram



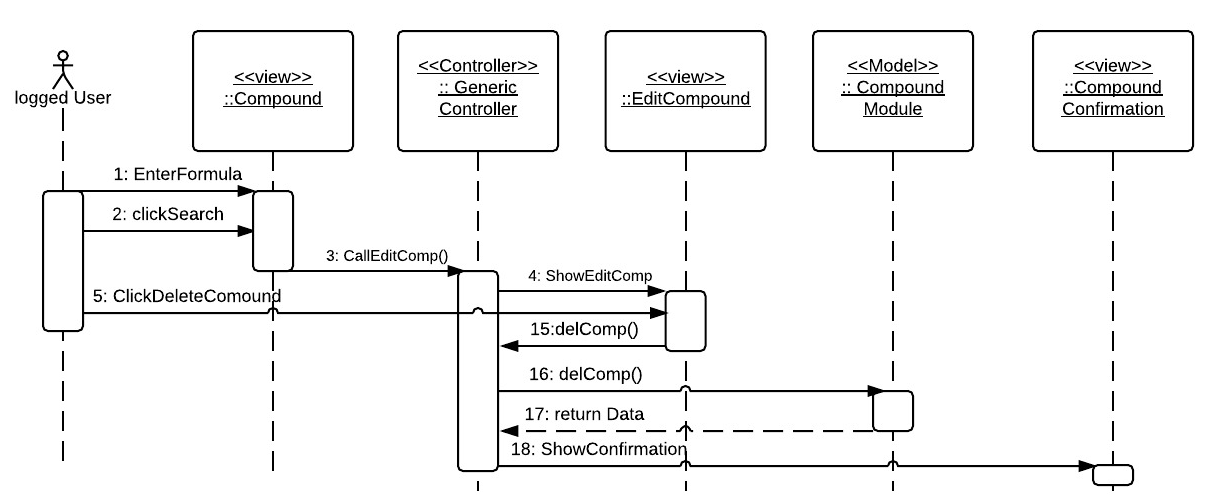
Change User Password Sequence Diagram



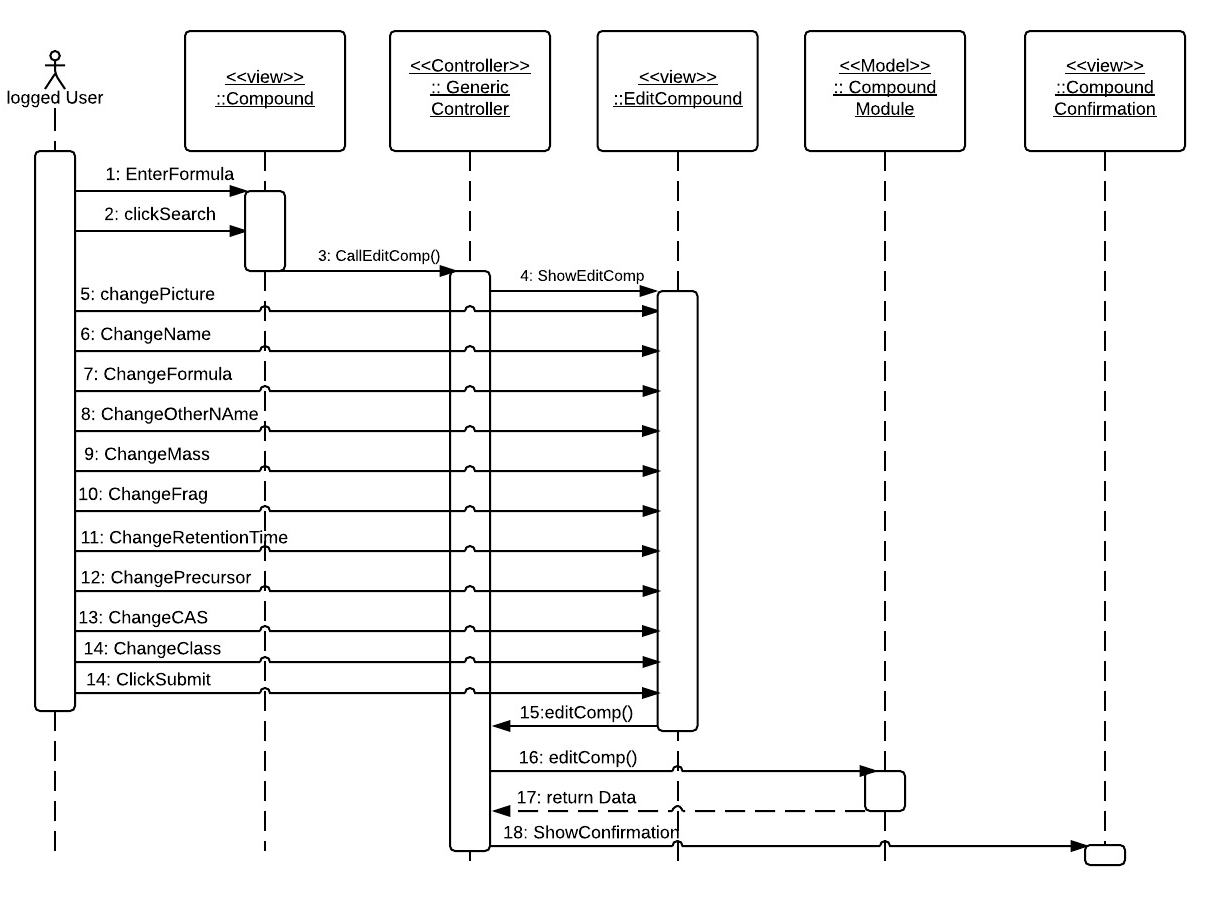
Import Compound Sequence Diagram



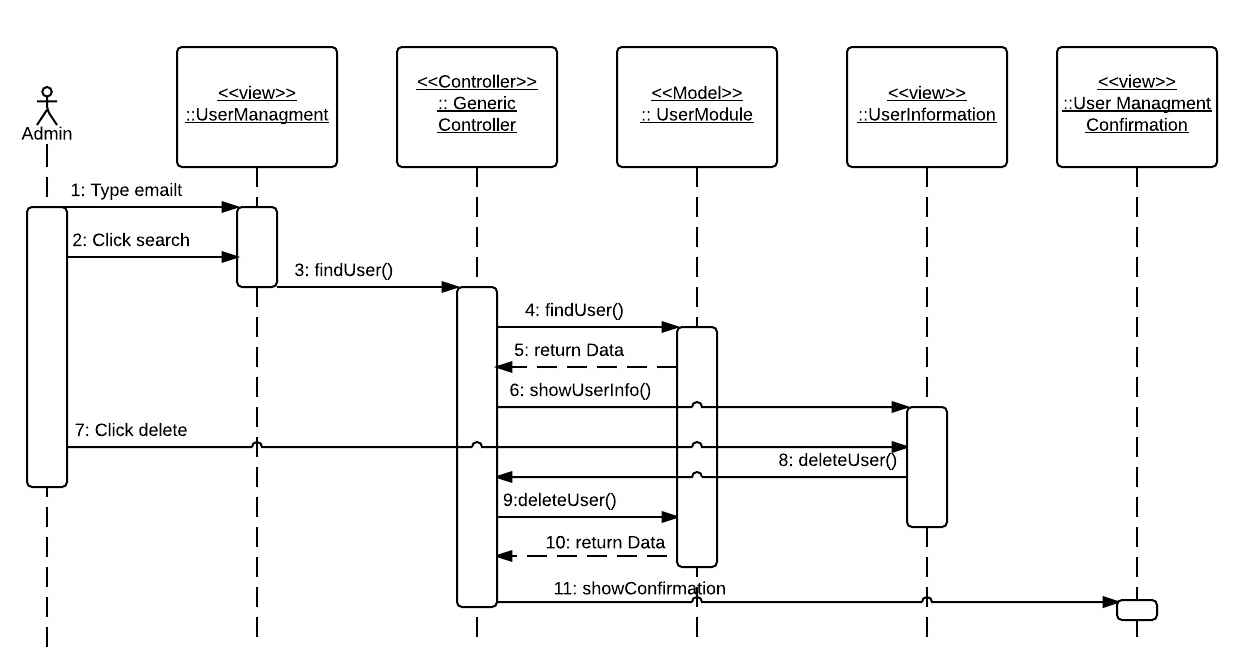
Delete Compound Sequence Diagram



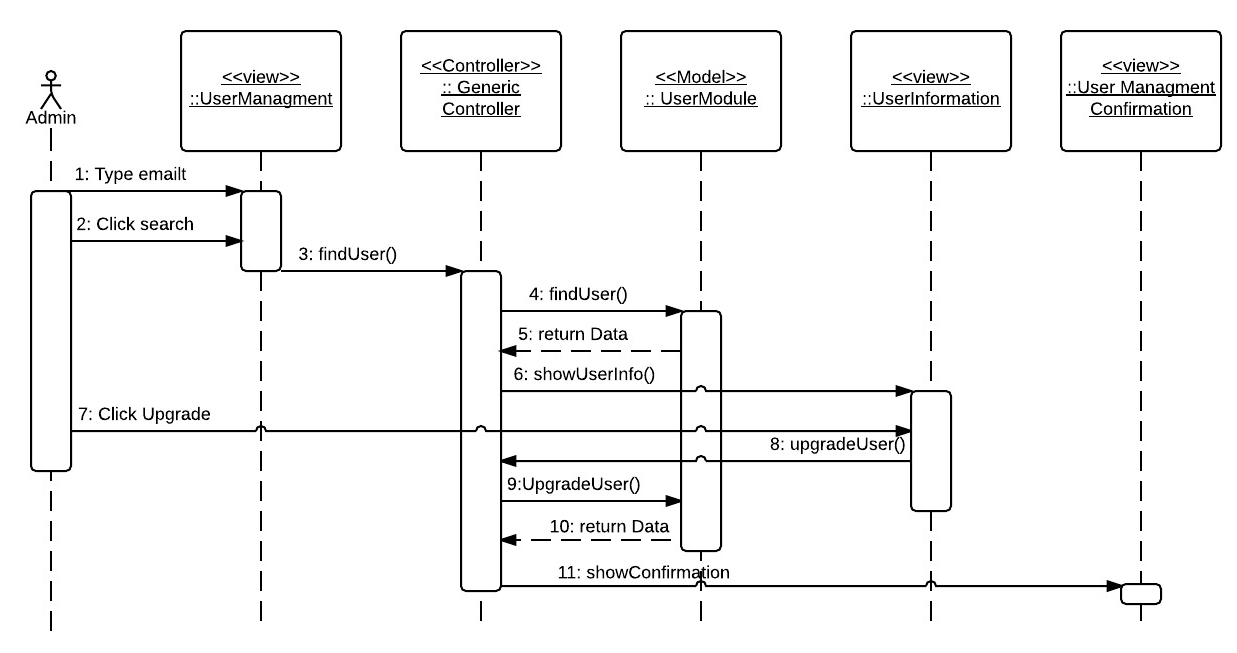
Edit Compound Sequence Diagram



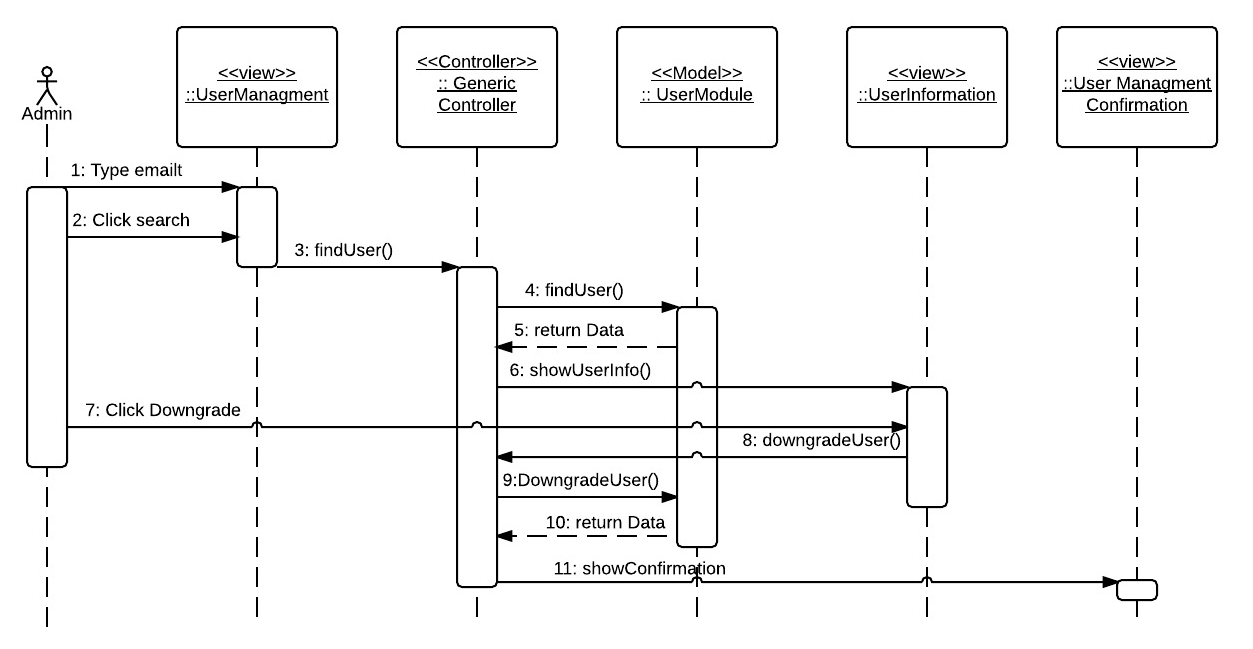
Delete User Sequence Diagram



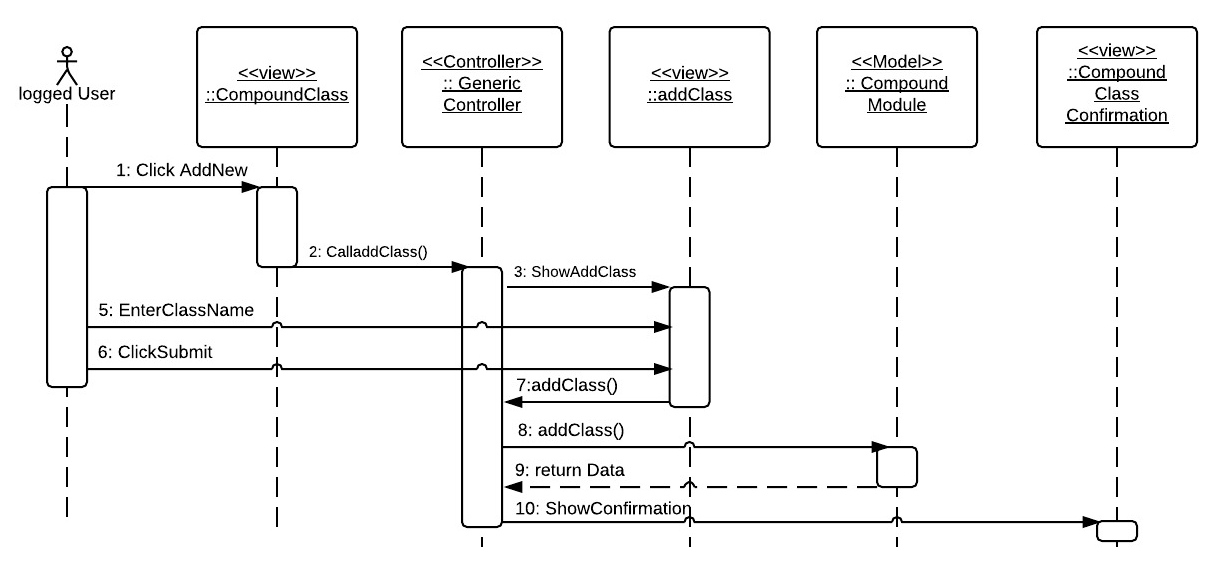
Upgrade User Role Sequence Diagram



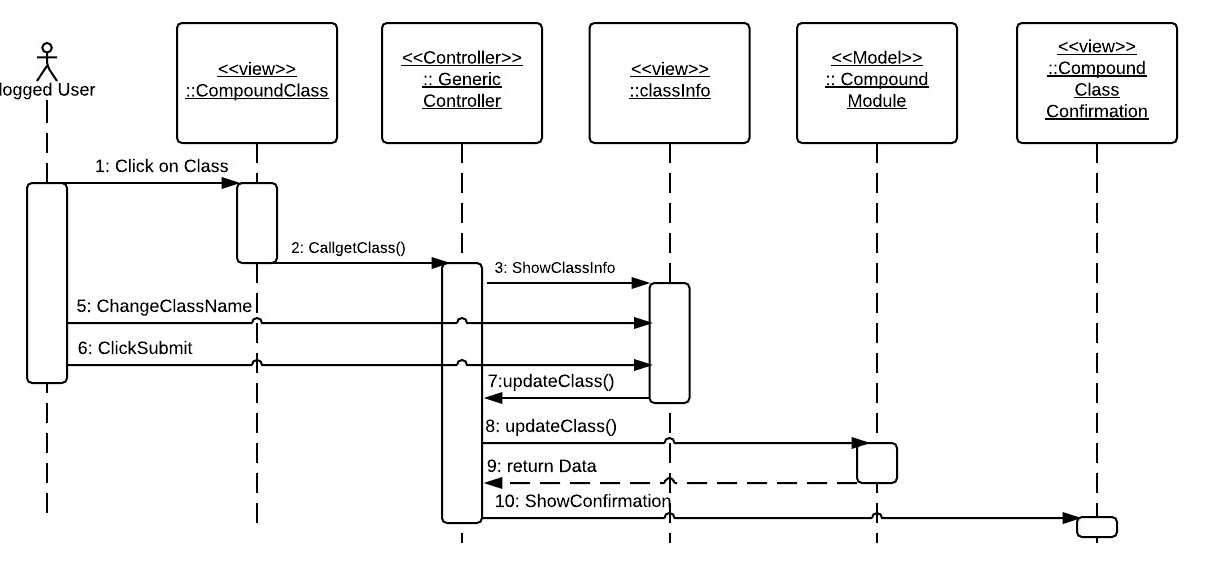
Downgrade User Role Sequence Diagram



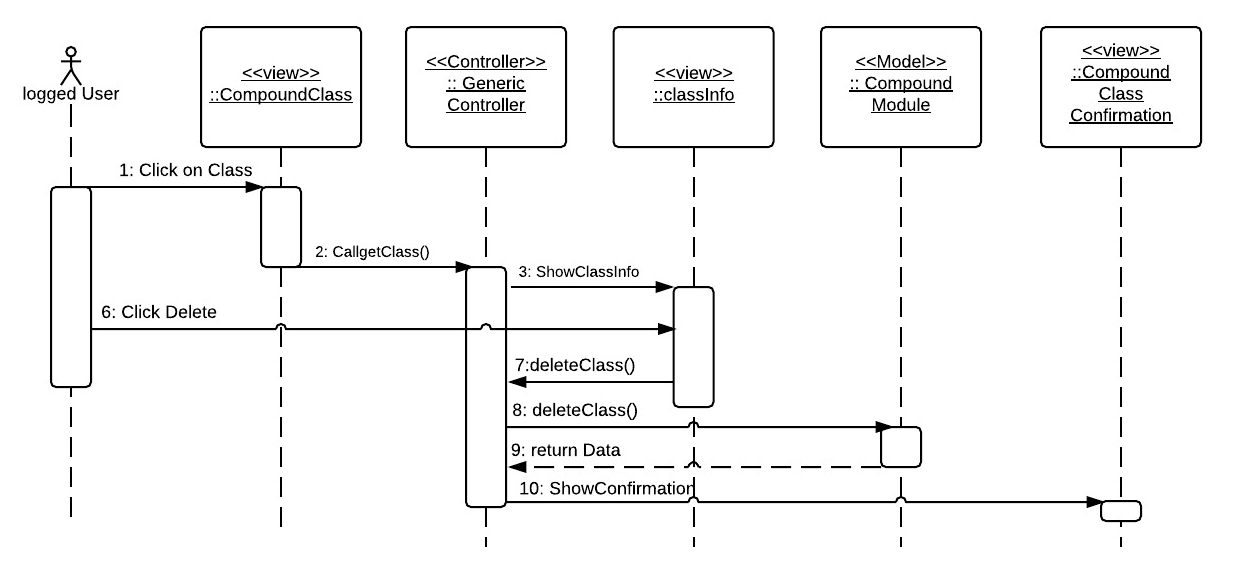
Add Compound Class Sequence Diagram



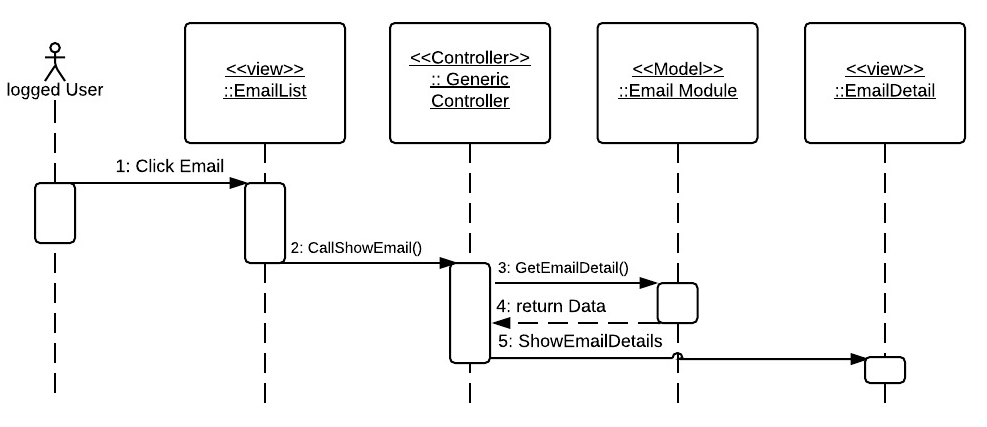
Edit Compound Class Sequence Diagram



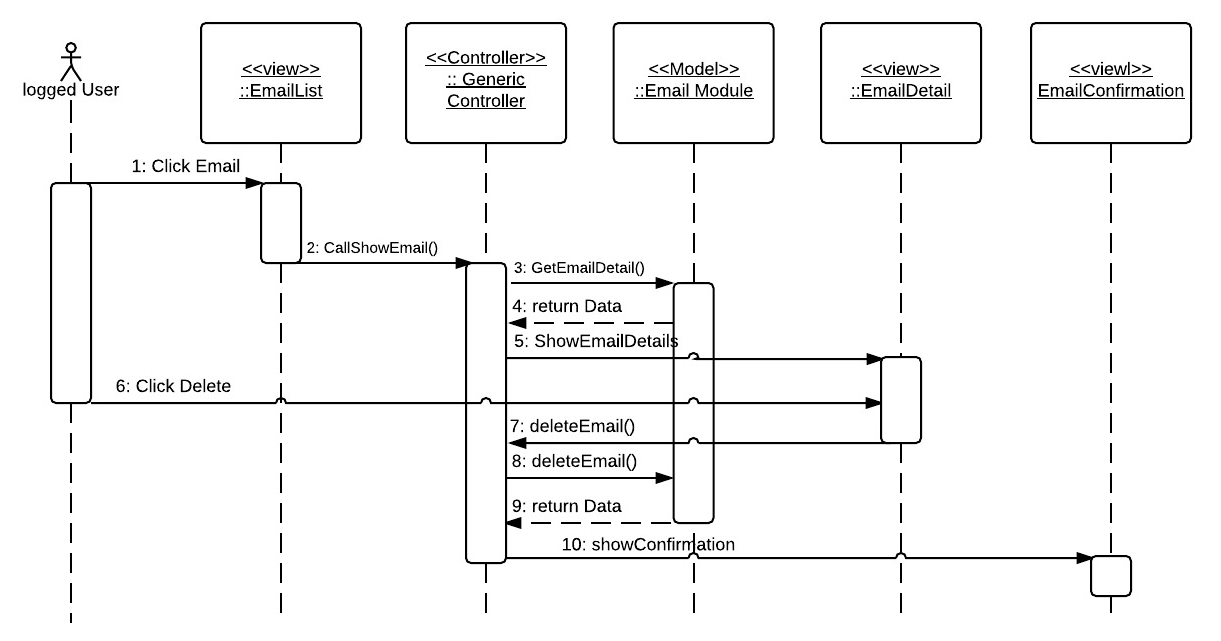
Delete Compound Class Sequence Diagram



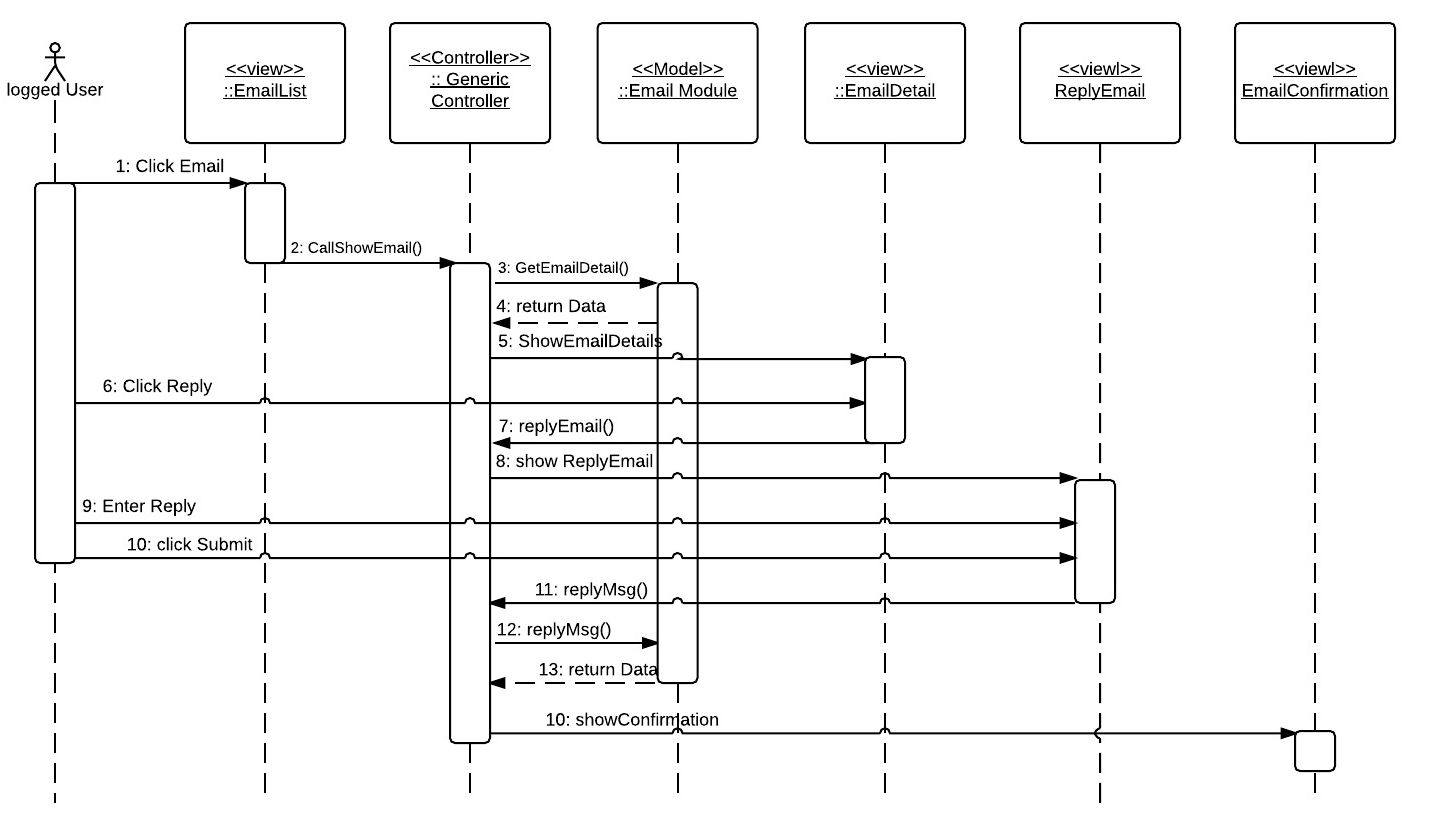
Read Email Sequence Diagram



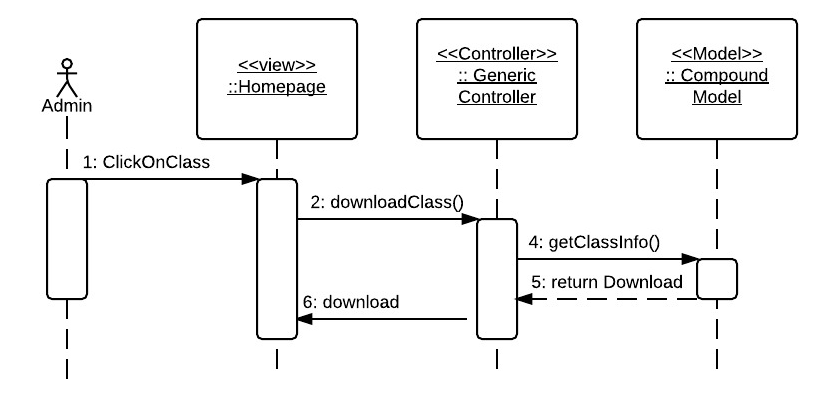
Delete Email Sequence Diagram



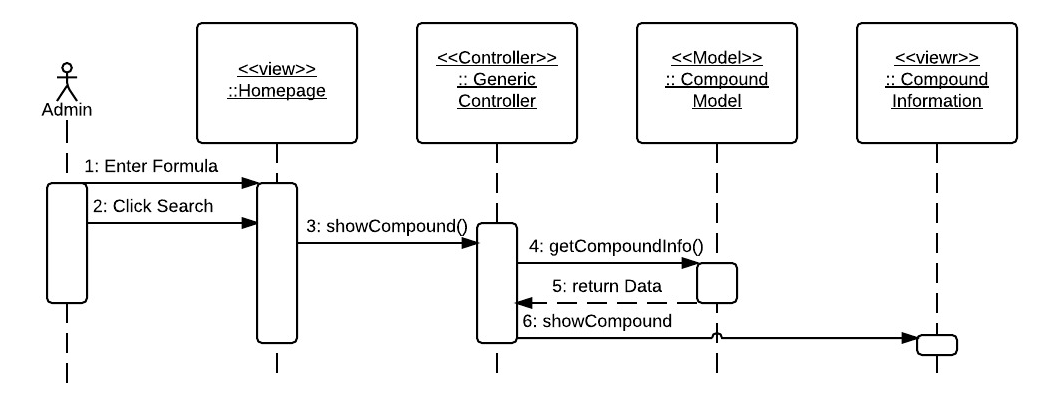
Reply email Sequence Diagram



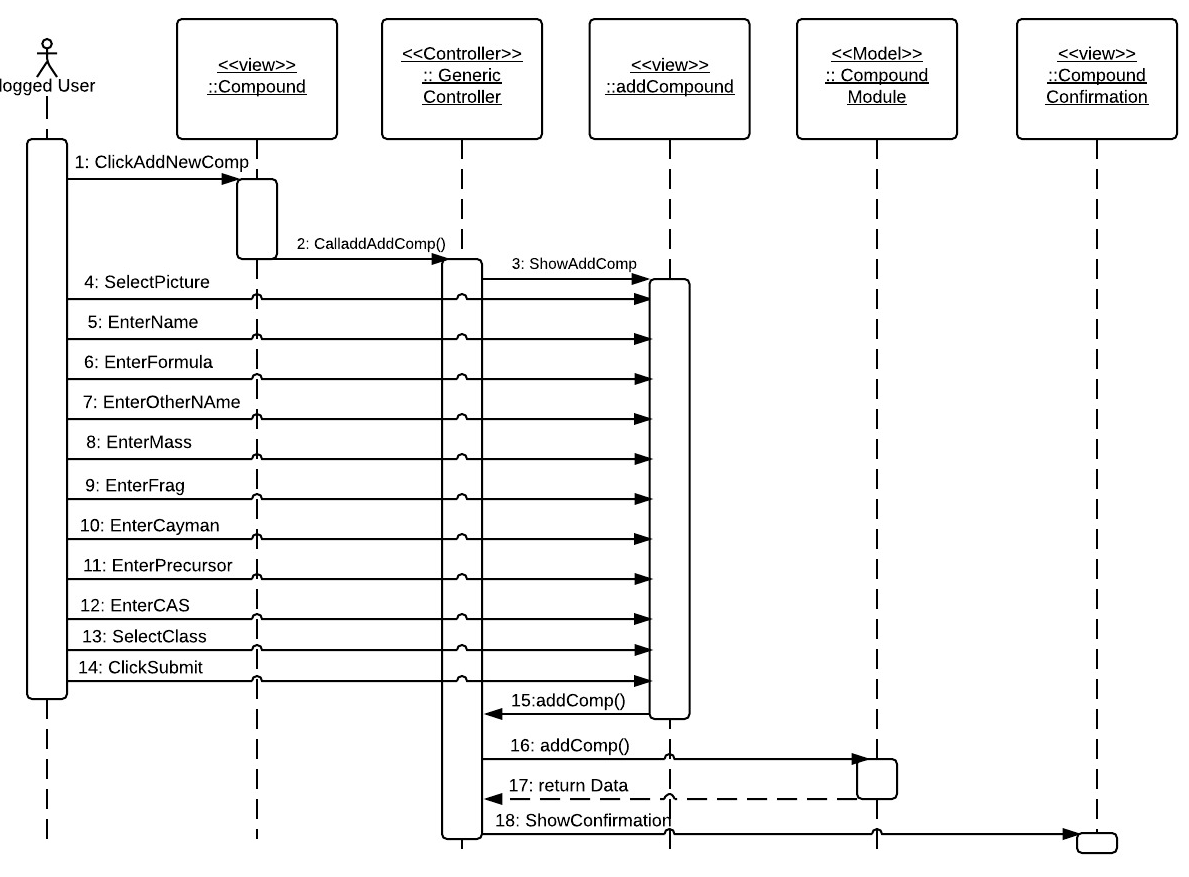
Download Class Sequence Diagram



Show Compound Sequence Diagram



Add Compound Sequence Diagram

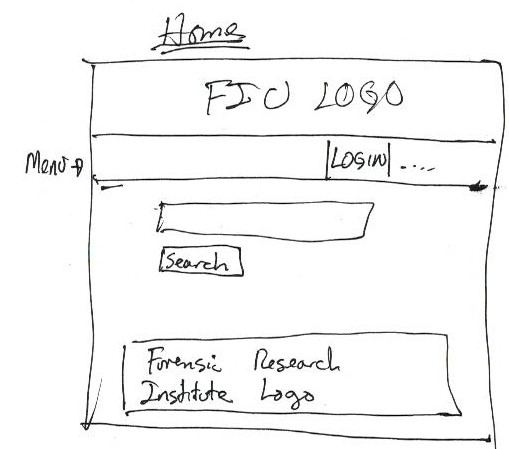


## 6.5 Appendix E - User Interface designs

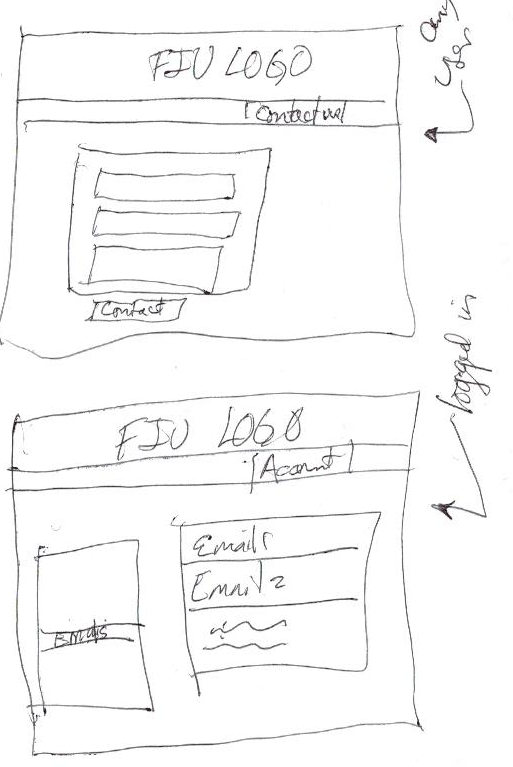
Account Menu



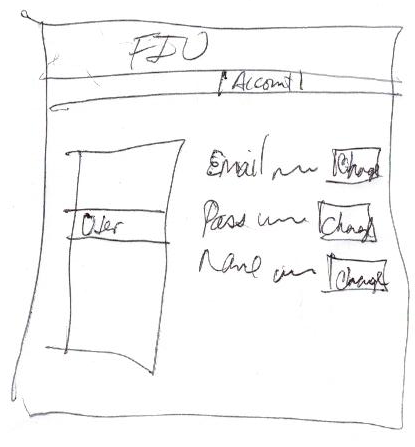
Account Home



Account Contact Us and Read Email



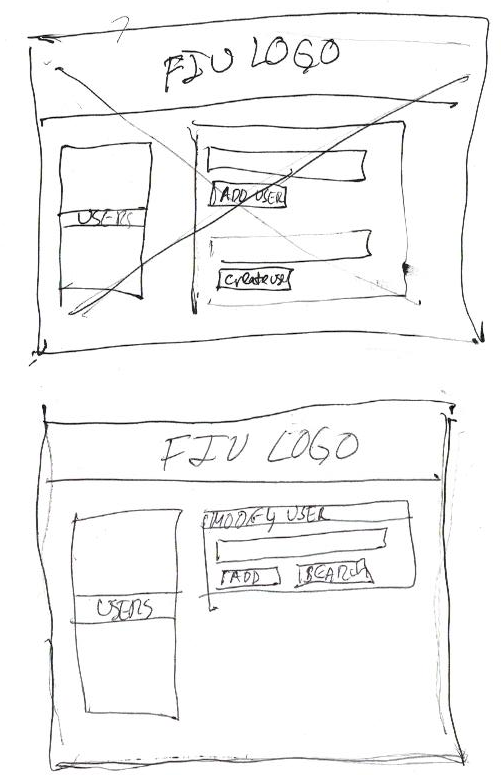
Account User Information Management



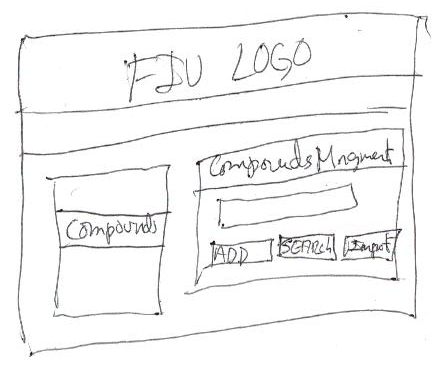
Compound Information



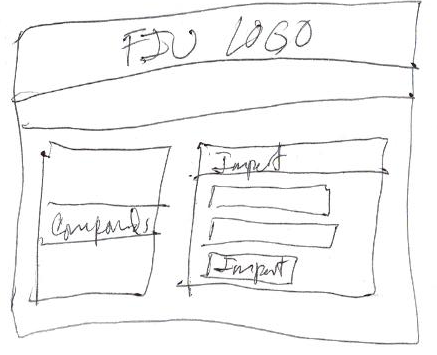
User Managment



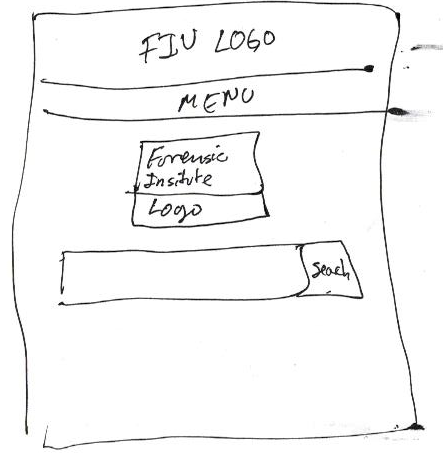
Compound Management

****

Compound Import Management

****

App Home

****

## 6.6 Appendix F - Diary of meeting and tasks.

### Sprint 1

|  |  |
| --- | --- |
| Diary Entry 1 | |
| Date | Monday, February 2rd, 2015 |
| Location | OE building |
| Start | 1:00 PM |
| End | 2:00 PM |
| Attendees | * Carlos Dominguez * Dr. Luis Arroyo |
| Agenda | * Talk about the system that needs to be done. * Create system user stories. |
| Summary | * Explanation of the current system. * Brief definition of the functionalities to be implemented. * Created some user stories based on the explanation |
| Assigned Tasks | * Get familiar with the tools to be used. * Create a mock up for acceptance. |

|  |  |
| --- | --- |
| Diary Entry 2 | |
| Date | Friday, February 6th, 2015 |
| Location | OE building |
| Start | 1:00 PM |
| End | 2:00 PM |
| Attendees | * Carlos Dominguez * Dr. Luis Arroyo |
| Agenda | * Show the mock up to where it was. * Show authentication system. |
| Summary | * Showed the mocked up to where it was done. * Talked about authentication system and showed what I had done. |
| Assigned Tasks | * Continue working on first Sprint |

|  |  |
| --- | --- |
| Diary Entry 3 | |
| Date | Friday, February 13th, 2015 |
| Location | OE building |
| Start | 1:00 PM |
| End | 2:00 PM |
| Attendees | * Carlos Dominguez * Dr. Luis Arroyo |
| Agenda | * Show what was done on Sprint 1 * Assigned for User Stories for Sprint 2 |
| Summary | * Showed the mocked up. * Show all that was done during Sprint 1. * New user story were assigned |
| Assigned Tasks | * Get ready to start spring 2 next Monday |

### Sprint 2

|  |  |
| --- | --- |
| Diary Entry 4 | |
| Date | Friday, February 27th, 2015 |
| Location | OE building |
| Start | 1:00 PM |
| End | 2:00 PM |
| Attendees | * Carlos Dominguez * Dr. Luis Arroyo |
| Agenda | * Show what was done in Sprint 2. * Assign User Stories to Sprint 3 |
| Summary | * Showed what was done in sprint 2. * Dr. Arroyo assigned user stories for sprint 3 |
| Assigned Tasks | * Get ready to start spring 3 next Monday |

### Sprint 3

|  |  |
| --- | --- |
| Diary Entry 5 | |
| Date | Friday, March 20th, 2015 |
| Location | OE building |
| Start | 1:00 PM |
| End | 2:00 PM |
| Attendees | * Carlos Dominguez * Dr. Luis Arroyo |
| Agenda | * Show what was done on Sprint 3 * Assigned for User Stories for Sprint 4 |
| Summary | * Show all that was done during Sprint 3. * New user story were assigned for Sprint 4 * Dr. Arroyo pointed out that different compounds could have the same formula, yet provide different information. |
| Assigned Tasks | * Get ready to start spring 4 next Monday * Fix what needed to be fixed on Sprint 4 regathing the new clarification on compounds formulas. |

### Sprint 4

|  |  |
| --- | --- |
| Diary Entry 5 | |
| Date | Friday, April 3rd, 2015 |
| Location | OE building |
| Start | 1:00 PM |
| End | 2:00 PM |
| Attendees | * Carlos Dominguez * Dr. Luis Arroyo |
| Agenda | * Show what was done on Sprint 4 * Assigned for User Stories for Sprint 5 |
| Summary | * Show all that was done during Sprint 4. * New user story were assigned for Sprint 5 * Dr. Arroyo said he wanted to change some of the information the institute is going to be sharing. |
| Assigned Tasks | * Get ready to start spring 5 next Monday * Fix the information that is to be shared on Sprint 5. |

### Sprint 5

|  |  |
| --- | --- |
| Diary Entry 5 | |
| Date | Friday, April 17th, 2015 |
| Location | OE building |
| Start | 1:00 PM |
| End | 2:00 PM |
| Attendees | * Carlos Dominguez * Dr. Luis Arroyo |
| Agenda | * Show what was done on Sprint 4 * Assigned for User Stories for Sprint 5 |
| Summary | * We discussed the agenda.. He was happy with the end result |
| Assigned Tasks | * No more tasks assigned. |

# References