

---

# Requirement Document

## **SCIS Curriculum Management**

by

Gabrielle Moestar and Oscar Aparicio

Course Instructor: Masoud Sadjadi

Mentor: Tim Downey

Client: Eric Johnson

CIS 4911 – Senior Project

12/04/2012

## **COPYRIGHT AND TRADEMARK NOTICES**

SCIS Curriculum Management Feasibility Study and Project Plan (“Document”) is claimed by Gabrielle Moestar and Oscar Aparicio.

### **Restrictions on copying or distributing the documentation**

With proper attribution and for non-commercial purposes, this Document governs your right to share and adapt. This, however, does not signify that the authors endorses or support your work.

### **Contractual obligations or disclaimers**

The Document is supported by Florida International University (“FIU”), School of Computing and Information Sciences. Data, tools, or results remain the property of their respective copyright holders.

### **Warnings and cautions**

Parts of the Document that fall in the public domain are not affected by this license. If the Document is distributed, this license must be clearly stated or a copy of the above link displayed.

### **Waiver**

The above conditions can be waived if the copyright holder grants permission.

### **Contact Information**

Florida International University - Modesto Madique Campus  
School of Computer and Information Sciences  
11200 SW 8th Street, Room ECS-258  
Miami, FL 33199

# Abstract

SCIS Curriculum Management intends to create, implement, and develop a content management system that is sophisticated and user friendly enough for the least technologically savant user to allow the creation and maintenance of course information and requirements for the curriculum of user-created workflows to be used on the data fed in to it. This system uses pre-existing course information data and requirements collected over the years of the existing educational programs provided by the SCIS Department and curriculum committee.

This document serves to describe the requirements and their expected implementation by listing the functional and nonfunctional requirements. There are 4 main chapters throughout the body of this document. Chapter 1 introduces the project with the problem definition and the scope of the system. Chapter 2 announces the current system. Chapter 3 discusses the project plan for this deliverable. Finally, Chapter 4 elaborates on the proposed system requirements by displaying the functional requirements and nonfunctional requirements, as well as the complete functional specification and its associated models. Finally, the last two chapters, Appendix and References, serving as the locations for more information referred to throughout the body of this document.

## TABLE OF CONTENTS

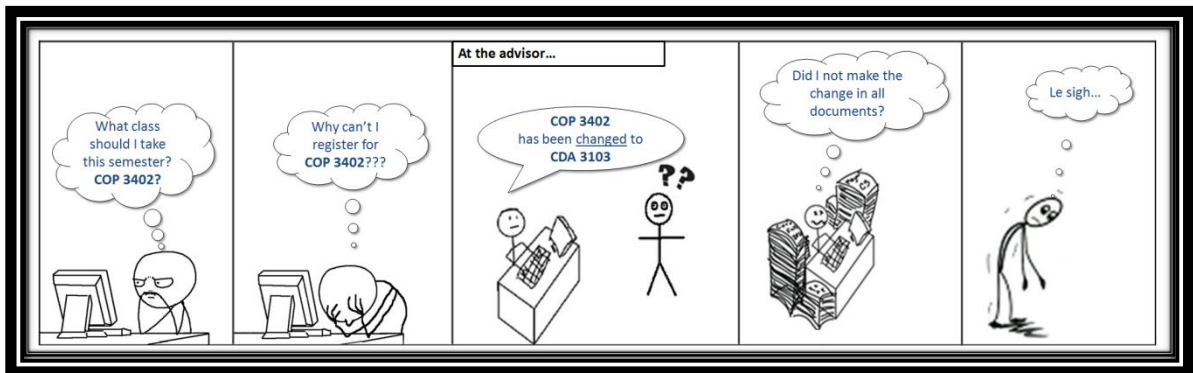
<b>1.</b>	<b>INTRODUCTION.....</b>	<b>5</b>
1.1	Problem Definition .....	5
1.2	Scope of System .....	6
1.3	Definitions, Acronyms, and Abbreviations .....	6
1.4	Overview of document .....	6
<b>2.</b>	<b>CURRENT SYSTEM .....</b>	<b>7</b>
<b>3.</b>	<b>PROJECT PLAN.....</b>	<b>8</b>
3.1	Project Organization .....	8
3.2	Work Breakdown.....	8
3.3	Cost Estimate .....	Error! Bookmark not defined.
<b>4.</b>	<b>PROPOSED SYSTEM REQUIREMENTS.....</b>	<b>9</b>
4.1	Functional Requirements .....	9
4.2	Analysis of System Requirements .....	10
4.2.1	Use Case Model.....	10
4.2.2	Static Model .....	10
4.2.3	Dynamic Model .....	11
<b>5.</b>	<b>GLOSSARY .....</b>	<b>12</b>
<b>6.</b>	<b>APPENDIX .....</b>	<b>13</b>
6.1	Appendix A – Project Schedule: Gantt Chart .....	13
6.2	Appendix B - Use Cases .....	13
6.3	Appendix C - Use Case Diagrams Using UML.....	14
6.4	Appendix D - Static UML Diagrams .....	14
6.5	Appendix E - Dynamic UML Diagrams .....	16
6.6	Appendix F - User Interface Designs.....	17
6.7	Appendix G - Diary of Meeting and Tasks .....	Error! Bookmark not defined.

# 1. Introduction

This first chapter briefly discusses the SCIS Management, and the scope of the system. In addition, an overview of the document is provided to give a brief explanation what to expect in chapter 2 through 6.

## 1.1 Problem Definition

The problem defined for this current system is the inability to easily create and maintain course information and requirements of the curriculum across documents consistent, which allows confusion for students.



This information can be obtained through different sources, an advisor, the departmental website, the catalog, and a major map. The data distribution through these documents appears to be inconsistent, and not in sync. (Undergraduate Programs)

The reason for this unfortunate inconvenience is initiated at the creation and maintenance of the curriculum by the curriculum committee.

## 1.2 Scope of System

The SCIS Curriculum Management system will be developed to serve as a central repository and main source of data to generate all the documents that needed for the documents.

The new system will have a graphical user interface that allows creation and modification to the curriculum. It will allow regular users, students, to view the curriculum, and allow special users with editor privileges, to create and modify the curriculum.

The system will permit flexibility for additional features in the future.

## 1.3 Definitions, Acronyms, and Abbreviations

**SCIS** – School of Information and Computing and Information Sciences

**Students** – Users with read only privileges

**Advisors** – Users with read and write privileges

**CRUD** – Create, Retrieve, Update, Delete

**DGU** – Degree Granted United: A college/school offering majors.

## 1.4 Overview of document

The remaining part of the document consist of 3 main chapters and 3 reference chapters serving as the locations for more information referred to throughout the body of this document

Chapter 2 – Current System describes the steps and system currently used by the department.

Chapter 3 – Project Plan lists the project organization for this deliverable and its work breakdown structure.

Chapter 4 – Proposed System Requirements contains the requirements and models needed for this system.

The last three chapters, 5 – Glossary, 6 – Appendix, and 7 – References, contain information and diagrams referenced throughout the body of this document.

## 2. Current System

The current system yields towards old file system maintenance. Changes to the curriculum are approved by the faculty and sent to the university curriculum committee to be reviewed and approved. Varied documents are created that are based on this information: catalogs, pamphlets, fliers, web sites, plans of study, major maps. When changes are made to the curriculum, someone has to be sure that all the documents are updated. Over time, the contents of these documents diverge and contain outdated information.

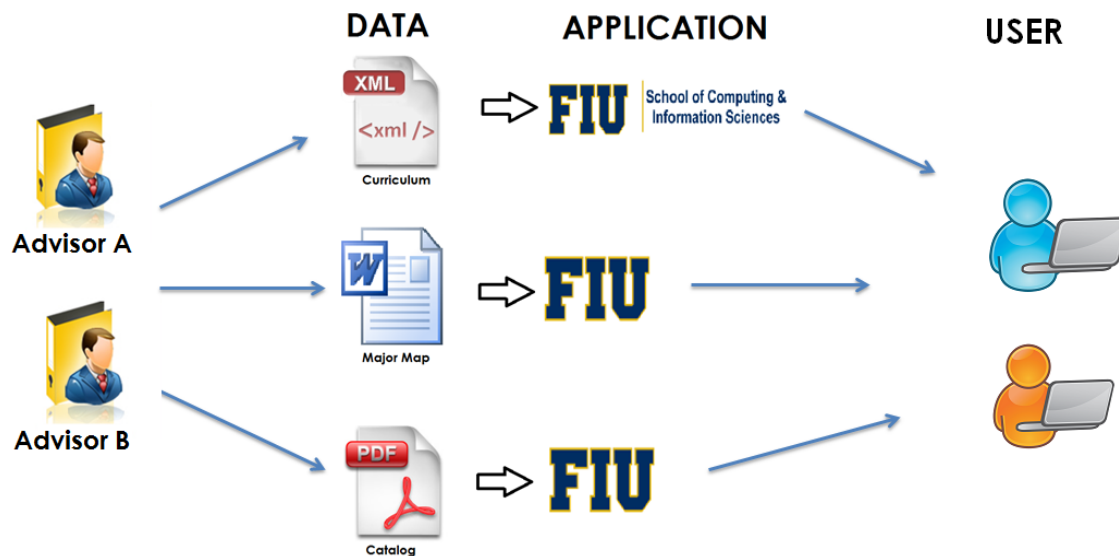


Figure 2-1 Current System process

### 3. Project Plan

This third chapter describes the organization for the project as a whole, describing roles of all team members throughout this phase of the project. The chapter closes with a work break down of the tasks needed to accomplish in this phase of the project.

#### 3.1 Project Organization

The assignment of project and roles for group members is presented below.

Name	Project Role
Gabrielle Moestar	Documenter
Oscar Aparicio	Project Manager   Developer

#### 3.2 Work Breakdown

For this deliverable, the Software Requirement phase, the project is broken down in to 7 different yet dependable tasks, presented in the table below. For a Gantt chart visually depicting the tasks below, reference in **Appendix A – Project Schedule: Gantt Chart**

	Tasks	Task Dependencies
1	Review current system	x
2	Problem Definition	1
3	Obtain High Level User Requirements	2
4	Identify Alternative Solutions	3
5	Determine Solution and Recommendation	4
6	Hardware / Software Identification and Requests	3,5
7	Requirement Analysis	2
	Milestone: System Analysis	



## 4. Proposed System Requirements

This chapter describes in the functional and non-functional requirements for the SCIS Curriculum Management system. The last section and its subsections visually portray the analysis of the system requirement by use case, static, and dynamic models.

### 4.1 Functional Requirements

The intended system shall accomplish the following for a student i.e. a regular user. The system shall:

- Allow the user to choose a major, track, and year he got accepted to the university.
- Allow the user to view the curriculum information.

The intended system shall accomplish the following for an editor user i.e. an advisor.

The system shall:

- Allow the user to login and logout.
- Allow the user to create, manipulate, and modify curriculum information and course requirements.
- Allow the user to create, manipulate, and modify majors and tracks.

The intended system shall accomplish the following seen as a whole. The system shall:

- Provide a database stored on SCIS network servers.

The non-functional requirements for the system are as follows:

#### **Usability**

The user interface should be understandable to non-technical users, allowing them to navigate to view the curriculum information. The font of the graphical user interface of the system should be clear and easy to read. Color scheme should make use of FIU's color schemes, yet maximize contrast i.e. light background with dark foreground.

#### **Reliability**

The system should be highly available, with 99% up time. Maintenance of the system should not be required more than once in a quarter year.

### **Performance**

The system should respond within less than two seconds for any user action, including curriculum information retrieval, update submission, and any other user interaction with the system. The system should be available 24/7, with downtime allowed as specified in the above.

### **Supportability**

The system will not interfere with existing curriculum data provided by the SCIS department nor its current services. The system will only be available in English. The system is web-based, therefore, compatible with any operating system that can run a supported web-browser mentioned below and connect to the internet.

### **Implementation**

The system will be a web-based application supported in Internet Explorer 7+, Firefox 5+. The system should support cross-platform compatibility without the need to change ports across platforms.

## **4.2 Analysis of System Requirements**

This section contains the complete functional specification for the SCIS Curriculum Management system. It describes the diagrams referenced in **Appendix C-E** and confirms the models against the use cases presented in **Appendix B**.

### **4.2.1 Use Case Model**

The Use Case Model found in **Appendix C** demonstrates the relationship and communication between the SCIS Curriculum Management system and the users.

### **4.2.2 Static Model**

**Appendix D** – Static UML Diagrams contains the complete diagram of the system.

### **4.2.3 Dynamic Model**

Appendix E contains the sequence diagram description for a few of the use cases. Below are descriptions of the sequence diagrams depicting the flow.

#### **Retrieve Curriculum:**

This use case begins when the student selects a major. The system responds with an updated list for tracks, where the student can select the track for the major selected. The student also needs to select the year he entered the university. Once submitted, the database will retrieve the data with the associated criteria for major, track, and year. The system will successfully display this information i.e. the curriculum.

#### **Create Course:**

This use case begins when the advisor clicks the Manage option in the advisor view, at which point the system responds by retrieving all courses from the database, followed by displaying the courses in the view. The advisor clicks the create course option. The system in turn, displays a form with the fields to be filled out with the course information. The advisor inputs the data. Once submitted, the data will be written to the database. The course has been successfully added to the list of courses.

#### **Update Course:**

This use case begins when the advisor clicks the Manage option in the advisor view, at which point the system responds by retrieving all courses from the database, followed by displaying the courses in the view. The advisor clicks the course he wants to update. The system display the data retrieved from the database in a form with the fields that have the ability to be edited. Once the change has been submitted, the data will be written to the database. The course has been successfully modified.

## 5. Glossary

This chapter contains definitions of terms used throughout the document.

**Functional requirement:** A description of a set of inputs, behavior, and output of a software system.

**Non-functional requirements:** A requirement that specifies criteria used to judge the system.

**Use case:** A description of a potential series of interactions between software and users.

**Sequence Diagram:** A diagram that shows how objects operate with one another and in what order.

## 6. Appendix

### 6.1 Appendix A – Project Schedule: Gantt Chart



Figure 6-1 Work breakdown for this deliverable

### 6.2 Appendix B - Use Cases

The following section contains the use cases implemented for the system

ID	Name
1	Login
2	Logout
3	Get catalog for students
4	Get catalog for authenticated user
5	CRUD DGU
6	CRUD Major
7	CRUD Group
8	CRUD Track
9	CRUD Course

#### Use Case ID: 5 – 9Login

**Details:** The user has to log into the system to begin working.

Actor(s):

- 1) Advisor

Pre-Conditions:

- 1) The user has logged in.

Description:

- 1) Use case begins when the system displays the entity<sup>1</sup> fields.
- 2) The user applies the modifications to the system.
- 3) The user hits SAVE.

<sup>1</sup> Entity refers to either DGU, Major, Group, Track, or Course

- 4) The system processes the data.
- 5) Use case ends The system displays confirmation page.

Post-Conditions:

- 1) The entity has been updated.
- 2) The main entity page is re-displayed.

**Exceptions:**

- 1) The user fails to fill out all fields requiring information, at which point the system notifies the user to fill out the required fields.

## 6.3 Appendix C - Use Case Diagrams Using UML

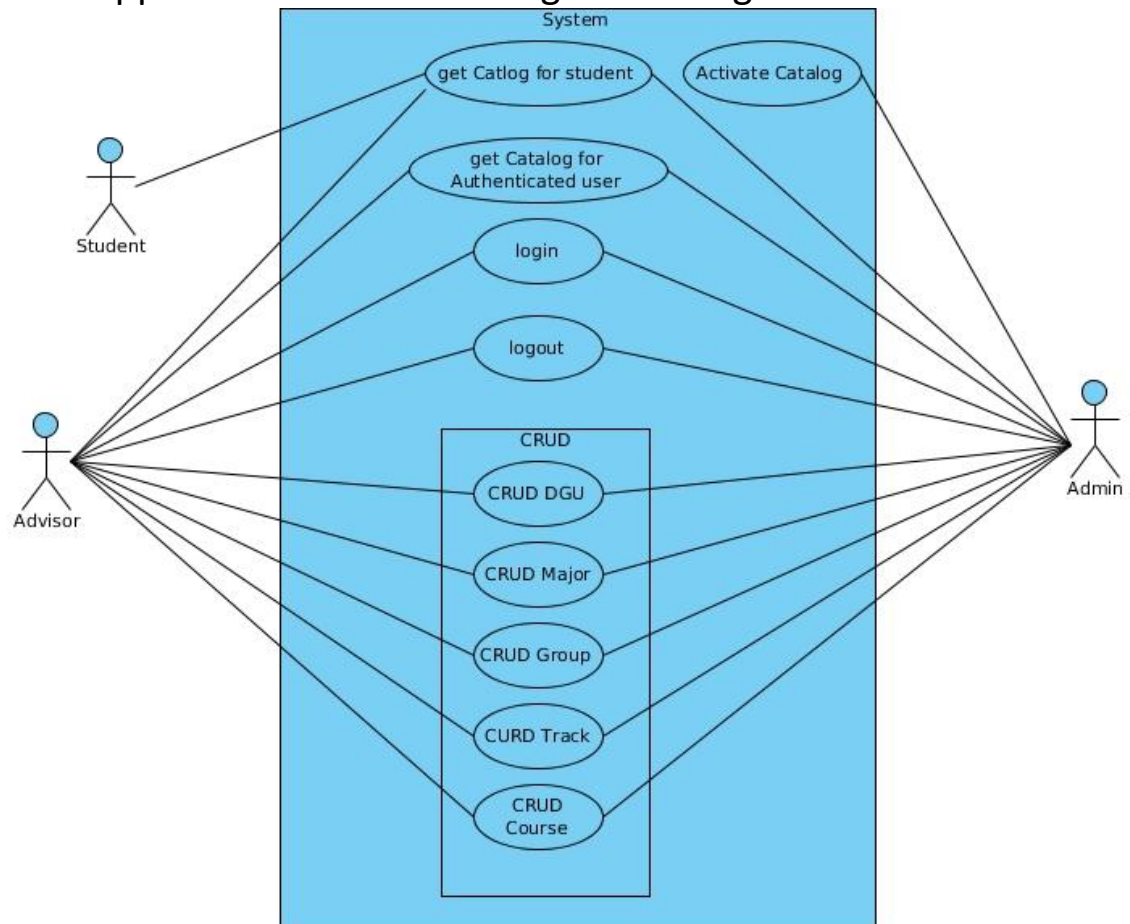


Figure 6-2 Complete use case diagram

## 6.4 Appendix D - Static UML Diagrams

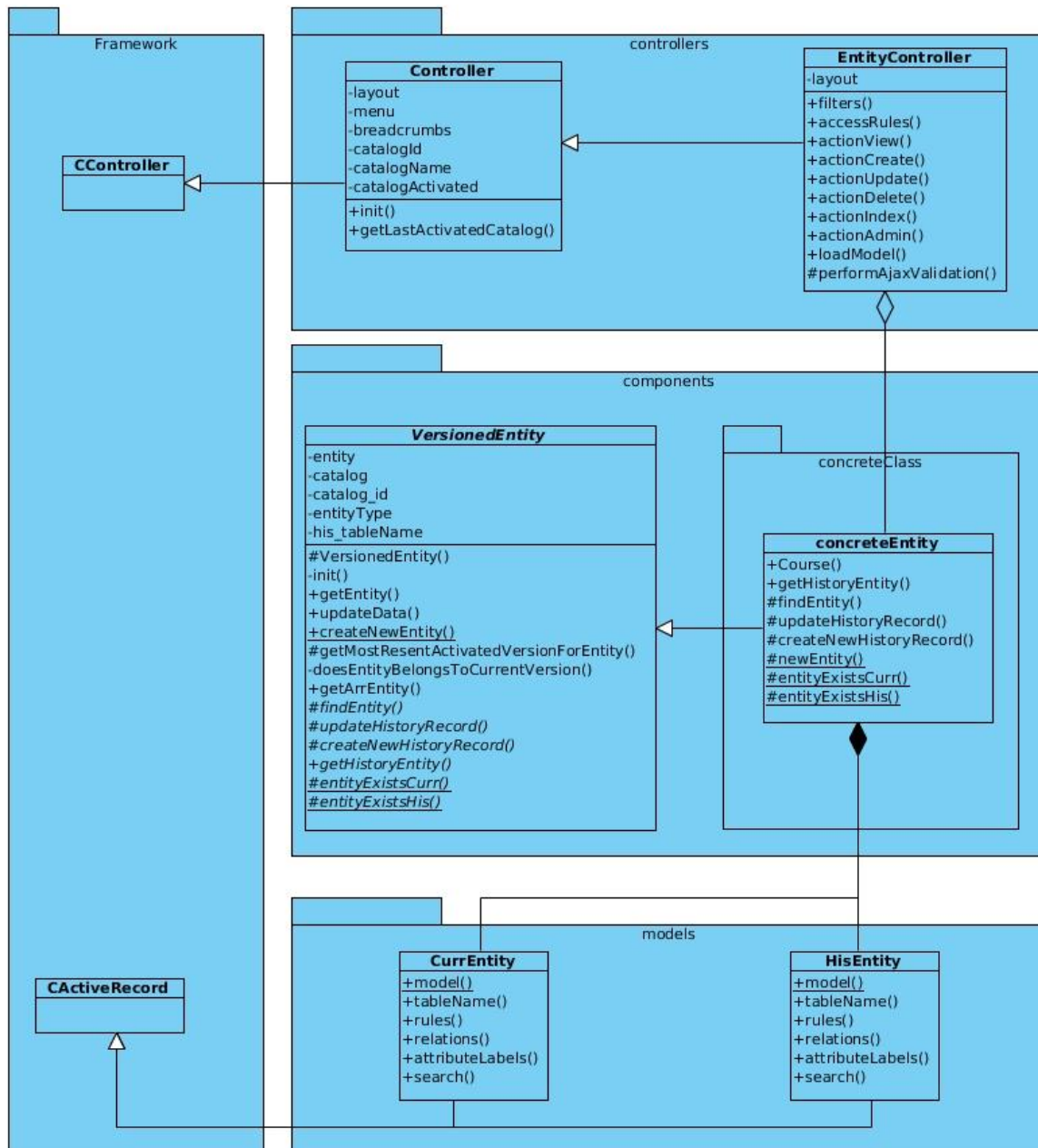


Figure 6-3 Overall View of class diagram relations

## 6.5 Appendix E - Dynamic UML Diagrams

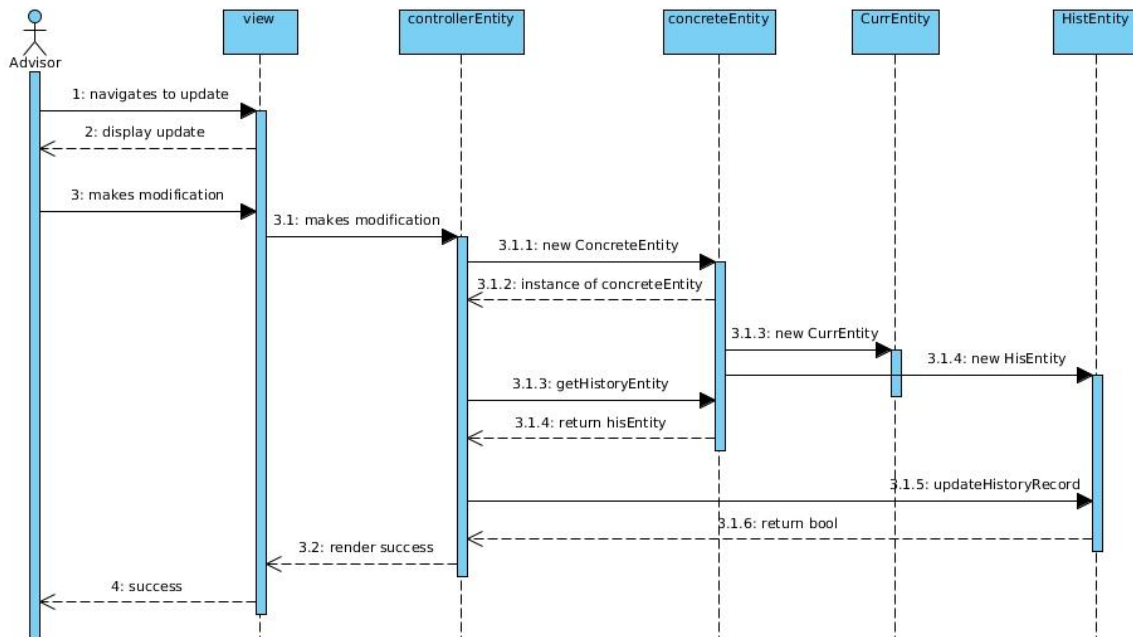


Figure 6-4 General Sequence Diagram for Update of CRUD for an entity.

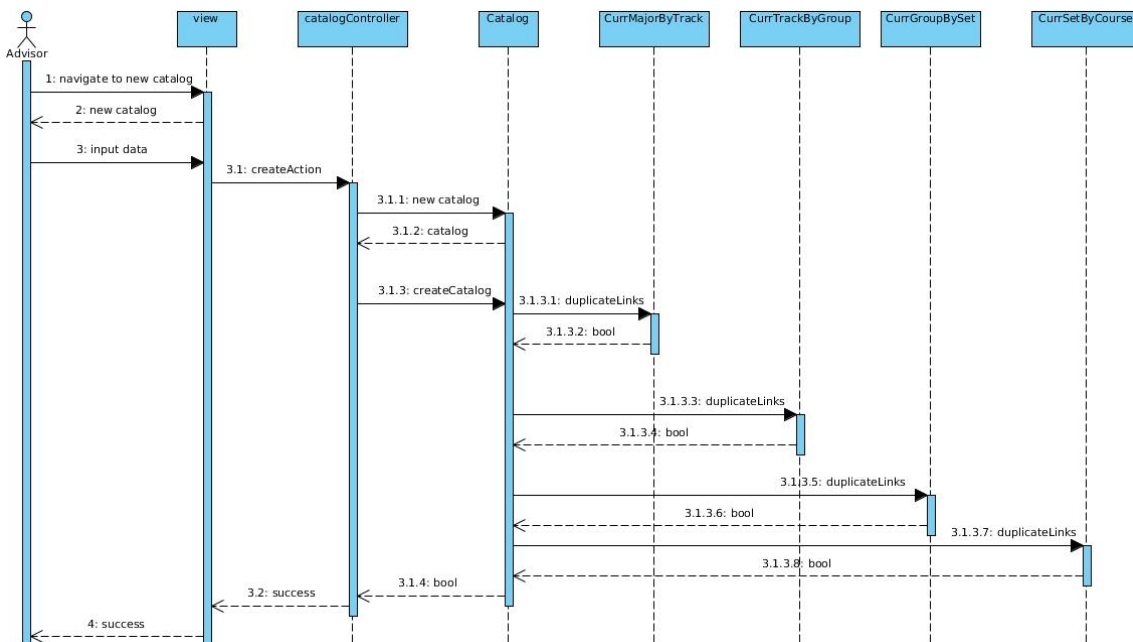


Figure 6-5 Sequence diagram for Use Case ID: 3 -- Generate catalog for advisor



## 6.6 Appendix F - User Interface Designs

The following view will be the interfaces depicting the process to update a course. In the example below, we are updating information on COP 3402.

**FIU** | School of Computing & Information Sciences

Home Contact Manage Profile Logout (advisor) Catalog: [first year](#)

[Home](#) » Courses


### Courses

Displaying 71-80 of 225 results.  
Sort by: [ID](#) [Course Name](#) ▲

<a href="#">CGS 3990</a> Designing Web Pages
<a href="#">COP 3835</a> Designing Web Pages
<a href="#">CGS 3993</a> Designing Web Pages
<a href="#">CGS 3996</a> Designing Web Pages
<a href="#">CEN 6501</a> Distributed Processing
<a href="#">CNT 6207</a> Distributed Processing
<a href="#">CEN 5120</a> Expert Systems
<a href="#">CDA 3103</a> Fundamentals of Computer Systems
<a href="#">COP 3402</a> Fundamentals of Computer Systems
<a href="#">CEN 4072</a> Fundamentals of Software Testing

Go to page: [< Previous](#) | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | [Next >](#)

Figure 6-6 Course List



School of Computing &  
Information Sciences

[Home](#)
[About](#)
[Contact](#)
[Login](#)


[Home](#) » [Courses](#) » Fundamentals of Computer Systems

Fundamentals of Computer Systems	
Course Name	COP 3402
Title	Fundamentals of Computer Systems
Abstract	Overview of the computer systems organization. Data representation. Machine and assembly language programming.
Credit	3
Prerequisite	
Corequisite	
Notes	

Operations

[List Courses](#)
[Create Course](#)
[Update Course](#)
[Delete Course](#)

Figure 6-7 Course information



School of Computing &  
Information Sciences

[Home](#)
[Contact](#)
[Manage](#)
[Profile](#)
[Logout \(advisor\)](#)

Catalog: [first year](#)

[Home](#) » [Courses](#) » [Fundamentals of Computer Systems](#) » Update

## Fundamentals of Computer Systems

Fields with \* are required.

Course Name \*

Course Prefix \*

Number \*

Abstract \*

Notes

Credits \*

Operations

[List CurrCourse](#)
[Create CurrCourse](#)
[View CurrCourse](#)
[Manage CurrCourse](#)

Figure 6-8 Modify information and save

The following view will display the interface of use case ID 3 – Generate catalog for students.

**FIU** | School of Computing & Information Sciences

[Home](#) [About](#) [Contact](#) [Login](#)

Fields with \* are required.

**Major \***

**Track \***

**Term you where accepted into the University \***

Copyright © by Oscar&Tiffany Productions.  
 All Rights Reserved.  
 FIU - School of Computer Science!

Figure 6-9 Input and Select Major, Track, and Term

**FIU** | School of Computing & Information Sciences

[Home](#) [About](#) [Contact](#) [Login](#)

[Home](#) » [Computer Science](#) >> [Software Design and Development Track \(1\)](#)

SDD Prerequisite	
COP 2210	Computer Programming I
MAC 2311	Calculus I
MAC 2311	Calculus I
PHY 2049	Physics with Calculus I w/Lab
PHY 2049	Physics with Calculus II w/Lab

SDD Core Course	
CDA 3103	Fundamentals of Computer Systems
CDA 4101	Structured Computer Organization
CEN 4010	Software Engineering I
CGS 1920	Introduction to Computing
CGS 3092	Professional Ethics and Social Issues in Computer Science
CGS 3095	Technology in the Global Arena
CNT 4713	Net-centric Computing
COP 3337	Computer Programming II
COP 3530	Data Structures
COP 4338	Programming III
COP 4555	Principles of Programming Languages
COP 4610	Operating Systems Principles
COP 4710	Database Management
COT 3420	Logic for Computer Science

SDD Elective Courses	
COP 4009	Windows Components Technology
CNT 4403	Computer and Network Security
COP 4225	Advanced Unix Programming
COP 4226	Advanced Windows Programming
CNT 4513	Data Communications
CAP 4710	Principles to Computer Graphics

Figure 6-10 Display Curriculum information