Feasibility Study and Project Plan

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

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

The School of Computing and Information Sciences, SCIS, offers a range of educational programs. Each program has its own curriculum. The curriculum committee of the SCIS Department arranges a set of courses required for each program. This information is broadcasted through websites, pamphlets, catalogs, and many other documents.

This project aims to address the shortcomings by creating, implementing, and developing 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 feasibility study of the project as a whole, as well the project plan. The document is divided in to three main chapters, accompanied by a fourth containing extra information. The first chapter is the introduction briefly defining the problem at hand, continuing with some necessary background information. The second chapter describes the feasibility study for the project, which contains the description of the current system with its limitations and constraints, followed by the proposal of the new system, including a high-level definition of user requirements, alternatives and recommendations. The third chapter goes over the project plan describing the project organization, and identifying tasks, major milestones, and deliverables. 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

1. INTRODUCTION 5
1.1 Problem Definition 5
1.2 Background 6
1.3 Definitions, Acronyms, and Abbreviations 7

2. FEASIBILITY STUDY 8
2.1 Current System 8
2.2 Purpose of New System 8
2.3 User Requirements 8
2.4 Alternative Solutions 9
2.4.1 Description of Alternatives 9
2.1.2 Selection Criteria 10
2.1.3 Analysis of Alternatives 10

3. PROJECT PLAN 12
3.1 Project Organization 12
3.1.1 Project Personnel Organization 12
3.1.2 Hardware and Software Resources 12
3.2 Tasks, Milestones and Deliverables (work breakdown) 13

4. APPENDIX 14
4.1 Appendix A - Project Schedule 14

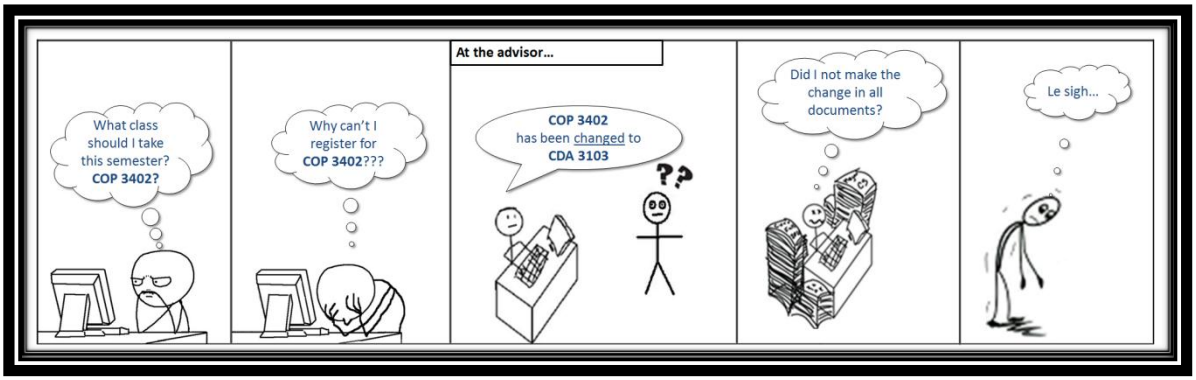
5. REFERENCES 15

1. Introduction

The SCIS Management system is a content management system designed to easily create and manipulate curriculum information with the help of a user-friendly web based application that will retrieve data from a database containing all course information and course requirements.

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.



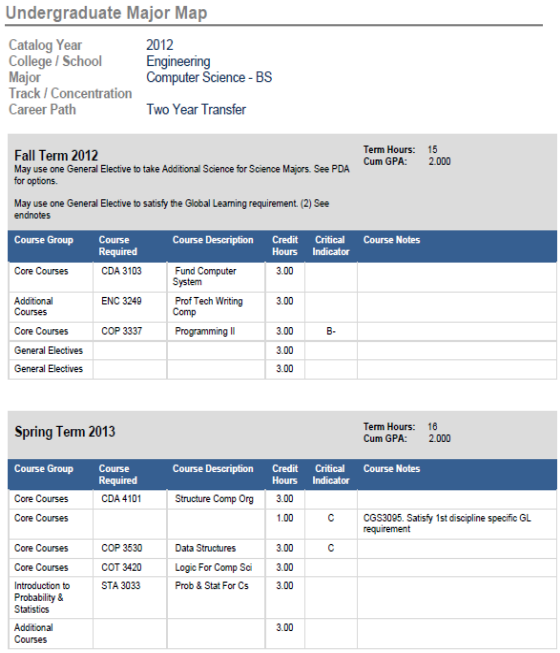
This information can be obtained through different sources, an advisor, the departmental website, the catalog, and a major map. The data distribution through these documents appears to be inconsistent, and not in sync.

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

1.2 Background

A curriculum is a group of related courses designed for a specific field of study. The curriculum can be found in three different documents:

- The department's website: http://www.cis.fiu.edu
- The Major Map: A program of study designed to help you graduate in a timely manner. The major map is divided in semesters and contains courses required for your major.
- The Catalog: A complete enumeration of major's offered by the department with its associated courses.



Undergraduate Major Map

Catalog Year 2012
College / School Engineering
Major Computer Science - BS
Track / Concentration
Career Path Two Year Transfer

Fall Term 2012 — Term Hours: 15 — Cum GPA: 2.000

May use one General Elective to take Additional Science for Science Majors. See PDA for options.

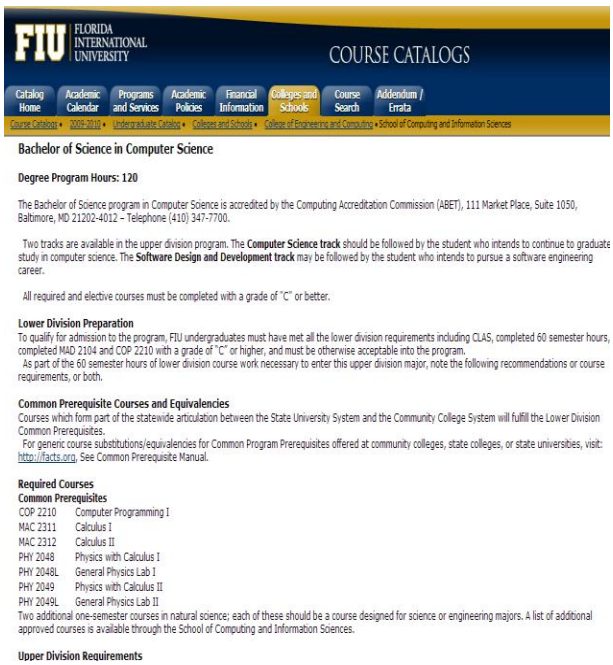
May use one General Elective to satisfy the Global Learning requirement. (2) See endnotes

Course Group	Course Required	Course Description	Credit Hours	Critical Indicator	Course Notes
Core Courses	CDA 3103	Fund Computer System	3.00		
Additional Courses	ENC 3249	Prof Tech Writing Comp	3.00		
Core Courses	COP 3337	Programming II	3.00	B-	
General Electives			3.00		
General Electives			3.00		

Spring Term 2013 — Term Hours: 16 — Cum GPA: 2.000

Course Group	Course Required	Course Description	Credit Hours	Critical Indicator	Course Notes
Core Courses	CDA 4101	Structure Comp Org	3.00		
Core Courses			1.00	C	CGS3095. Satisfy 1st discipline specific GL requirement
Core Courses	COP 3530	Data Structures	3.00	C	
Core Courses	COT 3420	Logic For Comp Sci	3.00		
Introduction to Probability & Statistics	STA 3033	Prob & Stat For Cs	3.00		
Additional Courses			3.00		

Figure 1-1 Major Map



FIU FLORIDA INTERNATIONAL UNIVERSITY — COURSE CATALOGS

Catalog Home | Academic Calendar | Programs and Services | Academic Policies | Financial Information | Colleges and Schools | Course Search | Addendum / Errata

Course Catalogs • 2009-2010 • Undergraduate Catalog • Colleges and Schools • College of Engineering and Computing • School of Computing and Information Sciences

Bachelor of Science in Computer Science

Degree Program Hours: 120

The Bachelor of Science program in Computer Science is accredited by the Computing Accreditation Commission (ABET), 111 Market Place, Suite 1050, Baltimore, MD 21202-4012 – Telephone (410) 347-7700.

Two tracks are available in the upper division program. The **Computer Science track** should be followed by the student who intends to continue to graduate study in computer science. The **Software Design and Development track** may be followed by the student who intends to pursue a software engineering career.

All required and elective courses must be completed with a grade of "C" or better.

Lower Division Preparation

To qualify for admission to the program, FIU undergraduates must have met all the lower division requirements including CLAS, completed 60 semester hours, completed MAD 2104 and COP 2210 with a grade of "C" or higher, and must be otherwise acceptable into the program.

As part of the 60 semester hours of lower division course work necessary to enter this upper division major, note the following recommendations or course requirements, or both.

Common Prerequisite Courses and Equivalencies

Courses which form part of the statewide articulation between the State University System and the Community College System will fulfill the Lower Division Common Prerequisites.

For generic course substitutions/equivalencies for Common Program Prerequisites offered at community colleges, state colleges, or state universities, visit: http://facts.org, See Common Prerequisite Manual.

Required Courses

Common Prerequisites

COP 2210	Computer Programming I
MAC 2311	Calculus I
MAC 2312	Calculus II
PHY 2048	Physics with Calculus I
PHY 2048L	General Physics Lab I
PHY 2049	Physics with Calculus II
PHY 2049L	General Physics Lab II

Two additional one-semester courses in natural science; each of these should be a course designed for science or engineering majors. A list of additional approved courses is available through the School of Computing and Information Sciences.

Upper Division Requirements

Figure 1-2 Catalog

Curriculum information is designed for students to know which courses to enroll for. This can be obtained through the documents mentioned above. Additionally, an advisor is another source that can provide you with the information.
The documents mentioned above are maintained in different files and formats after changes have been made to the curriculum.

1.3 Definitions, Acronyms, and Abbreviations

SCIS – School of Information and Computing and Information Sciences

Major Map – A program of study designed to help you graduate in a timely manner

Catalog – A complete enumeration of major's offered by the department with it associated courses.

2. Feasibility Study

2.1 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.

2.2 Purpose of New System

The purpose of the new system is to focus on creating a centralized repository as the main source of data to generate all the documents that needed for the documents along with an interface to manage all the data in an organized and efficient way.

The new system will have a graphical user interface that allows creation and modification to the curriculum and all related data needed to connect different parts of the curriculum.

The system will be designed to be easily expandable in the future, allowing the creation of new features and maintain the current modules. Therefore, we are considering all possible user requirements for the repository, and not the complete system implementation.

2.3 User Requirements

The following requirements are to be achieved:

- The system shall connect to a database for the curriculum, where changes will be dated, so a curriculum for any previous year can be easily generated and accessed.

- The system shall have a graphical user interface to input and retrieve curriculum data dynamically
- The system shall have a graphical user interface displaying the curriculum in an accessible manner and allow updates.
- The system will reference the database for a list of courses offered by the department.
- The system shall generate the different documents.
- The system shall run across all major web browser platforms and operating systems.
- The system shall provide flexibility to add new features in the feature.

2.4 Alternative Solutions

2.4.1 Description of Alternatives

There are different alternatives of the database design that would meet client's requirements. Discussed below are the three most contrasting and controversially alternatives.

The first option was a straight forward design where maintaining and displaying information about the curriculum and classes provided now to students and faculty uses the following approach. Every semester, a new table with all possible information about the classes offered by the school will be added to the database. Note that no other table is created.

A second approach for the design of the database cuts down data redundancy, and brings relational structure to the data stored. This approach will bring a relational database design, which normalized up to the fourth normal form that features a history set of tables. This set of tables is an exact copy of the core entities where new records will be produced every single time a new catalog (version) is created. To avoid increasing data and slow down database queries, only activated records will be on the curriculum set of tables and proposed/prospective records will remain in the history set of tables.

Finally, we envisioned a more sophisticated approach, where we would treat the changes in data as version. This approach would help manage creating and updating multiple releases of all entities needed to keep track of changes. This approach consists on a 2-tuple entity divided on the identifier table and the data table. The identifier would be a unique record that will differentiate the entity. The second element of the 2-tuple entity is the data table. This table is where the different versions of the data will remain to keep track of all changes generated in every new catalog year.

2.1.2 Selection Criteria

Decision making at all levels is about information collection, evaluation and tradeoffs to analyze complex problems. For this project, we will take the following criteria in to account:

- Data redundancy
- Usability
- Expandability
- Maintainability

2.1.3 Analysis of Alternatives

Based on the criteria described in the previous section, we are going to analyzed all three proposed approaches to select the better approach and the one that fits best for the system needs.

The first approach will bring a lot of data redundancy to the database, becoming a giant set of tables that contains the same information with, possibly, minor changes. Data will grow up exponentially saturating the database server at some point.

This approach would not completely complain with the usability and maintainability we want for the system, since we should change the source code every semester to reflect the changes of the tables.

The second approach complies more with our needs for maintain little data redundancy, and there would not be changes to the database at all. Since the database would be

fully normalized, data redundancy will be reduce significantly, but not enough considering all the data needed to change every term.

This approach would be fine for our purposes of the system, however at the time of implementation we discover the inability to completely implement this the way we expected. This was, since there are going to be two tables of the same information and we chose to InnoDb as our database engine, foreign key relationships where just broken. Therefore, we decided to move to the next approach.

The last approach meets our selection criteria. Formalized database with entities that can be versioned will bring a very low data redundancy allowing a better way of managing this information. The expandability still exists, since the overall design was kept as it was thought, and maintainability will increased due to integrity relationships between entities. Therefore, this approach is the one we decided to implement.

3. Project Plan

This third chapter will present the organization for the project as a whole, describing roles of all team members throughout the various phases of development. The chapter closes with a description of all major tasks, milestones, and deliverables.

3.1 Project Organization

This section describes in detail the roles of all group members throughout the major phases of the project, as well as the resources that will be needed.

3.1.1 Project Personnel Organization

Roles are not expected to change throughout the different phases of the project. Since this is a 2 person group, team members were forced function on multiple roles.

Name	Project Role
Gabrielle Moestar	Project Manager | Documenter | Front-End Developer | DBA | Tester
Oscar Aparicio	Project Manager | Version Control | Back-End Developer | DBA | Tester

The code that is to be written will be divided in front-end design and back-end work. The front-end design is

3.1.2 Hardware and Software Resources

This section covers both the hardware and software resources needed to complete this project and develop the requested system.

Hardware	Software
Server to Host Database	Ubuntu Server 12.10 LTS, 64 bit
Computer	Netbeans IDE 7.2.1
	Java SE Runtime Environment 1.7
	L.A.M.P.
	phpMyAdmin
	Microsoft Word 2007-2010
	Microsoft PowerPoint 2007-2010
	Visual Paradigm 8.3
	MacOSX MySQL Workbench 5.2

3.2 Tasks, Milestones and Deliverables (work breakdown)

	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	
8	Create Database Architecture	7
9	Populate Database	8
10	Create System Architecture	7
11	Tentative UI Design	7,8,10
12	Implement Back-end	8,10,11
13	Initial Testing	8,10, 11, 12
	Milestone: Tentative System	
14	Finalize System Implementation	13
15	Complete Functional Testing	12, 13
16	Evaluate Test Results	15
17	Arrange Final Document and Presentation	14, 15, 16
	Tasks	Task Dependencies

4. Appendix

4.1 Appendix A - Project Schedule

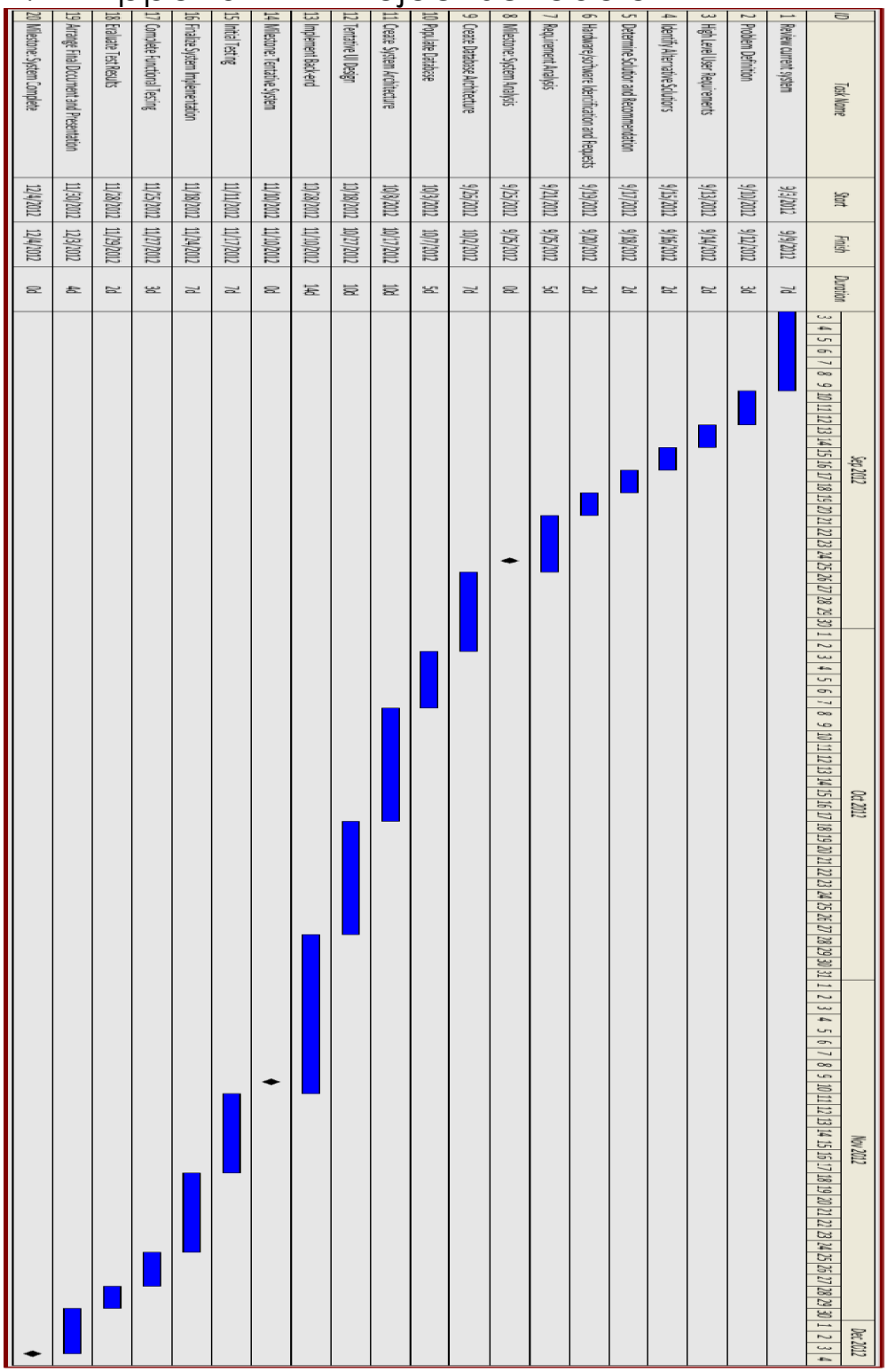


Figure 4-1 Gantt Chart

5. References