|  |
| --- |
| Group 1 |
| CIS – 4050 002 |
| Project 1 |

|  |
| --- |
| Richard Brookman, Tayven Bigelow, Adam Hill, and Bryan Bengoechea |

Contents

[Memorandum 1](#_Toc413086028)

[Executive Overview 2](#_Toc413086029)

[Description 2](#_Toc413086030)

[Analysis 2](#_Toc413086031)

[Development 2](#_Toc413086032)

[Recommendations and Conclusions 3](#_Toc413086033)

[Use Case Diagram 4](#_Toc413086034)

# Memorandum

To: Metro State School of Business

CC: Professor Moldanado

From: Group 1

Date: 3/2/2015

Re: Major Class Requirements

To whom it may concern:

It has come to the attention of the school of Business at Metro State that there is a problem with the registration procedures of the students. There is no process for a student to map out the course requirements for the major they are in. There are course descriptions and the requirements for those classes, however if a student requires any outside influence for registration direction, they must make an educated guess on the next set of classes they must register for. This is a severe hindrance for an apt registration process.

A group has been tasked with the creation of a program that will allow an advisor to meet with a student and prepare a detailed course path for him to follow for the next several semesters based upon his major and classes taken. With this program, the student will feel more at ease with the direction he needs to go for his program. This will also reduce the amount of time advisors will spend on each student by having his class path laid out several semesters in advance.

Success will be measured by the functionality of the program and how versatile and error-free it will be. It will be measured by the amount of students that use the program with the advisor, and how satisfied those students are with the program. This project will go from inception to working model within the next year.

-Group 1

# Executive Overview

This report details the initial problems of the class registration procedures, the process by which the group creates a program to solve those problems, and a description and final analysis of the program after it is done. This projects was assigned by Dr. Moldanado on January 21, 2015. All information in this report was created by Group 1, which includes Richard Brookman, Tayven Bigelow, Adam Hill, and Bryan Bengoechea.

## Description

Metro State University school of Business has had concerns that students are not using the advising office to help them in their academic pursuits. One of the problems they have addressed is the lack of an adequate advising on the path that prospective students must take to satisfy the course requirements for each of their majors. A survey was emailed to all business students via email as to the benefit of having such a program that would outlay a course for the students to follow. Of the over ten thousand emails sent out, 75% of the respondents favored a program that allowed them to map out the class requirements for their major based on previous classes taken. The advising center was also in agreement that having such a program would benefit the students and the staff. Therefore, a request was made of Dr. Moldanado to develop a program that allowed an advising staff member to meet with a student and display a class path for the student for a future two semesters. This project was then tasked to the group via Dr. Moldanado.

## Analysis

The first step of the group was to develop a problem analysis matrix to brainstorm all the problems that may occur during the creation of the program. The second step was to create a use-case scenario on how the users will interact with the program. Then several diagrams will be created to provide a blueprint of how the project will be programmed, how it will interact with all outside influences, and all the requirements that will be needed.

The problem analysis matrix details the problems associated with the project. Referring to the Appendix, the problems in the matrix include the storage and recall of all the data for each of the students in the school of business. This is caused by the lack of a database for the express purpose of holding student information with the requirements and pre-reqs of all classes in the school of business. There is a problem of the database connection to the school's database. This will cause the information for the student's classes that they have taken to be inadequate or false unless another avenue is explored. A third problem may be the major structure itself, and if that structure is concrete, or may have to be tailored to accommodate changes. These and other problems pose some difficulties that must be overcome to develop the program satisfactorily.

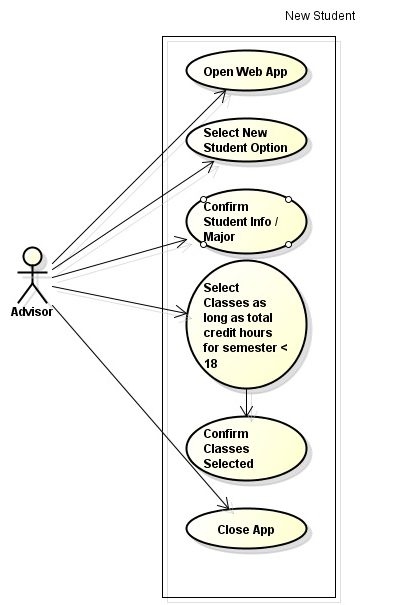
## Development

The development phase will include the creation of the database that will house the student's information. This database will be developed using ERD diagrams to show the relationships between the student, their classes they've taken, the class requirements for the corresponding major, and the pre-requisites for those classes. Use-case models will be used to show the interactions between the user of the program and how the program responds. A data flow diagram, a functional decomposition diagram, and a program design of primitive processes will be used to determine the processing requirements of the project. The problem analysis matrix ( included in the Appendix ) will detail the problems that are present, their causes and effects, the system objective and constraints, and the proposed solutions for each problem

## Recommendations and Conclusions

At this point in the process, a working model of the program, the database, and the interactions between them should be created to give an adequate representation of how this model works before applying it on a full scale to the school of business' student database. This will allow the evaluating of potential problems that may occur during the creation and implementation without affecting the existing database for the school. Once the model is out of its testing phase and can operate without any problems, then it shall be integrated into the system. At this time, the advising office will use the program and provide feedback to Dr. Moldanado for any proposed changes to the system.

# Use Case Diagram: New Student



# Use Case Narrative: New Student

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use-Case Name:** | Class Registration Application | | | |
| **Use-Case ID:** | 1 | |  | |
| **Priority:** | High | | | |
| **Source:** |  | | | |
| **Primary Business Actor:** | Student | | | |
| **Other Participating Actors:** | Advising staff | | | |
| **Other Interested Stakeholders:** |  | | | |
| **Description:** | This use case describes the events of using an application to help students register for classes based on students’ course records | | | |
| **Precondition:** | Application must be installed in all advising staff computers. Computers must have access to the intranet. | | | |
| **Trigger:** | This use case is initiated when the student approaches an advising staff for help | | | |
| **Typical Course of Events:** | **Student** | **Advising Staff Action** | | **System Response** |
| **Step 1:** Student seeks help from advising office. | **Step 2**: Advisor opens Application and enters student’s information. | | **Step 3:** System displays current student history. |
| **Step 4:** Student provides updated record of classes completed. | **Step 5:** Advisor enter completed classes into database. | | **Step 6**: System confirms classes updated. |
|  | | | **Step 7:** System displays projected classes |
|  | **Step 8:** Advisor prints a copy of the classes for student | |  |
| **Step 9:** Student takes the copy of the classes and leaves the office |  | | |
| **Alternate Courses:** | **Step 1a:** No advisors are available for help and student leaves the office. | | | |
| **Conclusions:** | Use case concludes when the student leaves the office with a copy of the classes to take for next semester | | | |
| **Post Conditions:** | Student is now ready to register for next semester | | | |
| **Business Rules:** | 1. Advising staff can select courses for students for as long as credit hours are less than 18 | | | |
| **Implementation Constraints and Specifications:** | N/A | | | |
| **Assumptions:** | Student has a declared major  Student will register for next semester | | | |
| **Open Issues:** | N/A | | | |

# Database Requirements



# Processing Requirements

## Functional Decomposition Diagram



## Data Flow Diagrams

### Context Diagram



### Zero Level

# Program Design

## Structured English

**1. Student Info Query Subsystem**

Student 900# is inputted

If Student exists then query student information and display it

Else set up as new student

1) Input additional information like Name and Major

2) Save Student information in database

3) Display student information

**2. Class History Query Subsystem**

Use student information (900#) to query class history

1) If it exists, display the class history

2) If it does not exist allow Advisor to input it if student has a class history

Query the required class list for the student’s selected major and display it

**3. Class Comparison Subsystem**

Compare the class history list to the required class list for the major

1) If a class appears in both lists then exclude it

2) Display list of remaining classes (untaken required classes)

**4. Class Prerequisite System**

Query pre-requisites for untaken required classes

Display pre-requisites for untaken required classes

**5. Eligible Classes Subsystem**

Compare pre-requisites for the untaken required classes to class history and student credit level

1) If the student does not meet minimum credit level then exclude that class

2) If the student has not taken a pre-requisite class then exclude that class

Display the remaining classes in the list for which the student meets the pre-requisites

**6. Class Saving Subsystem**

If semester credit level is less than 18 credit hours then allow advisor to select classes from eligible list

Else do not allow class selection

Save class selection to class history with the semester name (spring 2014 etc)

Re-run subsystems 2-5 to update eligible class list

Re-execute Class Saving subsystem until student meets all major requirements

Display full selected class list organized by semester and year

# Appendix:

## Problem Analysis Matrix:

## Questions Used:

## Project Timeline:

## Database (create table statements):