2016

Prepared by:

Amy Knowles

Steven Kundert

Yashwanth Reddy Muddireddy

Vaisali Namburi

2016 February 12

v.1.2

[Open Classes Project]

Requirements & Specifications





Contents

[1 Introduction/Overview 2](#_Toc445240774)

[1.1 Purpose 2](#_Toc445240775)

[1.2 Scope 2](#_Toc445240776)

[1.2.1 Main Objective 2](#_Toc445240777)

[1.2.2 Specific Goals 3](#_Toc445240778)

[2 System 3](#_Toc445240779)

[2.1 Development and Target Environment 3](#_Toc445240780)

[2.2 Who are the users? 4](#_Toc445240781)

[2.3 Functional Requirements 4](#_Toc445240782)

[2.3.1 Issues 5](#_Toc445240783)

[2.3.2 Major Subsystems or Functions 5](#_Toc445240784)

[2.3.3 Minor System Functions 5](#_Toc445240785)

[2.3.4 Major Classes Identified 5](#_Toc445240786)

[2.4 User Interface Specification 6](#_Toc445240787)

[2.5 Non-functional Requirements (constraints) 6](#_Toc445240788)

[2.5.1 Management 6](#_Toc445240789)

[2.5.2 Technical 6](#_Toc445240790)

[2.5.3 Performance 7](#_Toc445240791)

[2.5.4 Security 7](#_Toc445240792)

[2.6 System Evolution/Maintenance 7](#_Toc445240793)

[3 Other Deliverables 7](#_Toc445240794)

[4 Risks 7](#_Toc445240795)

[5 Glossary 7](#_Toc445240796)

[6 References 8](#_Toc445240797)

[7 Consultants 8](#_Toc445240798)

# Introduction/Overview

## Purpose

The purpose of this document is to serve as an agreement between the developer and customer on what the software will do. This document outlines the scope, application overview, and functional requirements for the open classes project.

## Scope

The customer advises undergraduate students in order ensure that the students are enrolled in the correct classes at the proper time in order to meet the requirements of either the B.A. or B.S. degree plan, depending on the student’s current degree track. In addition to these advising duties the customer is a Professor of Computer Science which includes a full course load and individual research projects. Thus the customer requests a more efficient and simplistic method of determining which classes are available for enrollment and would fit into the advisee’s schedule.

The current system for accessing a list of open classes is to utilize the campus registration database – Banner / Webworld. The current system allows the user to search the database for a given college, and then drill down into a specific class within that college. Information returned in the search includes the CRN, subject, course number, course section, number of credit hours, days of the week on which the course falls, time of the course, cap, current enrollment, open slots, instructor, and location of the course.

The current system is slow, and painstakingly tedious due to the drill down method of locating classes. Also, the system resets all search parameters for each class search. The customer wants a system which will be easy to use and graphical in nature. While the current system requires the user to drill down through a considerable amount of excess data, the customer envisions a system that accomplishes the same task using radio buttons, pull-down menus, and other GUI characteristics.

In addition, the customer would like to add several features to the system. The customer would like to be able to view all open core classes and/or open elective classes available at a given time on a single screen. They also want to be able to differentiate between the two computer science degree plans, as these plans differ significantly between core requirements. Furthermore, they would like to be able to see a list of all core classes, whether those classes are currently full or have available spaces.

As the customer is a professor of computer science, this application will focus on the two computer science undergraduate degree plans, and, if time permits, additional degree plans could be added. This design team will concentrate on having this application working for a Windows 7 / 10 environment, and, if time permits, an OS X environment.

### Main Objective

The main objective is to produce an application that the customer can use for advising beginning Summer 2016. The application will provide a user-friendly, graphical user interface that works on top of a mini-database generated by reading in a text file that was converted from an available PDF of the course catalog and should demonstrate enough features that the customer can successfully utilize the GUI to access open core and elective courses during the advisement period. The development phase will take place over a twelve-week period during the spring 2016 semester.

### Specific Goals

The customer requires a system that is graphical in nature. Radio buttons, use of a mouse and drop-down menus will be implemented. A PDF file containing the current class availability will be extracted from the MSU website, and translated into a text file via a utility, if a reliable tool can be located. This text file will then be used to create a temporary database of the current class availability which can be queried for the following information: B.S. / B.A. core classes, class availability, registration information, as well as days and times.

# System

The remainder of the document is intended to inform the customer, as well as the development team, of the intended system. Hardware and software requirements, major users, both major and minor functions, constraints, and intended user interface are described.

## Development and Target Environment

The development environment will be as follows:

**Hardware** will include a personal (desktop or notebook) computer with the following specifications:

* an Intel i5 microprocessor running at 2.0 GHz or above
* a minimum of 128 megabytes of system RAM
* a CD-RW drive
* a v.92 compliant 56K modem
* a hard drive with at least 2 gigabytes of free space
* an XGA screen display 14” or larger
* a graphics card with 16 megabytes or more of RAM

**Software** utilized will include:

* Windows 7 / 8 / 10
* Eclipse IDE
* PDF to TXT converter
* Microsoft Project
* MySQL Workbench

The **application** developed will utilize the following:

**Hardware** – a personal notebook computer with the following specifications:

* an Intel i5 microprocessor running at 2.0 GHz or above
* a minimum of 128 megabytes of system RAM
* a CD-RW drive
* a v.92 compliant 56K modem
* a hard drive with at least 2 gigabytes of free space
* an XGA screen display 14” or larger
* a graphics card with 16 megabytes or more of RAM

**Software:**

* Windows 7 / 8 / 10
* OS X Yosemite 10.10.5
* Executable file developed the project team
* PDF to TXT software
* MySQL Workbench

## Who are the users?

The users of the open classes software will be professors, primarily from the computer science department but should also include other faculty and staff tasked with advising undergraduate students majoring in computer science. If other degree plans were added at a later date to this software, then users could include all faculty and staff tasked with advising undergraduate students in any major for which a degree plan has been included. The students being advised would be considered stakeholders as well.

The respective values that each of these entities receives from the system are as follows:

**Computer Science Professors / Faculty / Staff:**

* efficient access to open a list of classes when advising undergraduates majoring in computer science
* an adaptable system which will allow degree plans to be updated to reflect future requirements

**Advising Faculty / Staff:**

* efficient access to open a list of classes when advising undergraduates majoring in computer science
* an upgradeable system to which multiple degree plans could be added in order to aid faculty and staff advising undergraduates in any major
* an adaptable system which will allow degree plans to be updated to reflect future requirements

**Undergraduate Students:**

* efficient display showing which required courses fit into their proposed schedule

## Functional Requirements

While the final version of the proposed software has many requirements, the prototype will focus on the two undergraduate degree plans within the computer science department and the search for a reliable PDF to TXT software.

### Issues

The issues are shown below:

* adapt reliable PDF to TXT software
* select between the two different computer science degree plans
* sort the classes into Major Core and Electives, and Fundamental Core and Electives
* search for open classes based on the type of course required by the student’s degree plan
* display class schedule information based on input search parameters

### Major Subsystems or Functions

* PDF to TXT converter: locate software or merger that will generate a reliable TXT document from the PDF course data available from the MWSU website.
* Computer Science degree plans: files containing information for the B.A. and B.S. degree plans
* Class search or query: a system by which open classes can be searched depending on degree plan and type of course required

### Minor System Functions

* Open classes search: option to return ONLY open classes or all classes, independent of availability, based on type of course selected
* Fundamental Core Classes search: option to return classes under the fundamental core umbrella
* Fundamental Elective Classes search: option to return classes under the fundamental electives umbrella
* Major Core Classes search: option to return classes required by the computer science department for all undergraduate computer science students
* Major Elective Classes search: option to return computer science courses that can be selected on a per student basis that would count toward the computer science undergraduate degree requirement of 18+ hours in advanced topics related to the computer science field.

### Major Classes Identified

This section contains a list of the major classes, denoted by a solid bullet point, and the member attributes for each class, denoted by an open bullet point.

* Degree Plan
  + B.A.
  + B.S.
* Course Category
  + Computer Science Core
  + Computer Science Electives
  + Additional Departmental Requirements
  + Fundamental Core Requirements
  + Social & Behavioral Sciences
  + Creative Arts
  + Language, Philosophy & Culture
  + Cultural & Global Understanding
  + Undergraduate Inquiry & Creativity
  + Science
  + Foreign Language
  + Literature - Humanities
  + Other Electives
* Course
  + CRN
  + Subject
  + Course
  + Section
  + Title
  + Days
  + Time
  + Cap
  + Enrolled
  + Remaining
  + Instructor
  + Location

## User Interface Specification

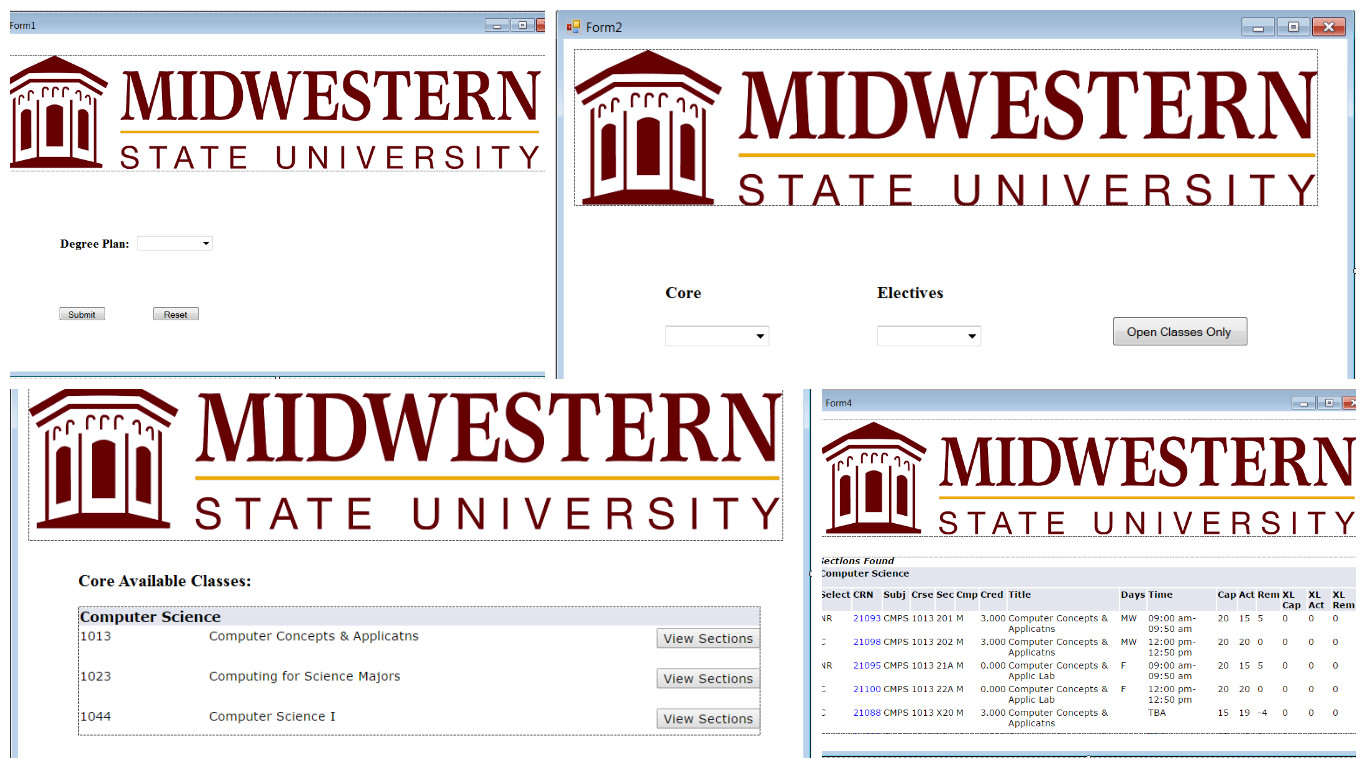
The customer’s idea of the preliminary user interface is shown below. From top left to bottom right the figure below shows the introduction screen, the core/ electives selection screen, available classes screen, and the final drill down screen showing individual course sections. 

Figure 1. GUI Design.

In accordance, the preliminary user interface should consist of at least the following:

* main window with system name and management buttons
* main menu bar (minimize, maximize, help, close)
* child window that will show search results
* drop down menu for degree plan choices
* selection buttons for search parameters
* search execution button

## Non-functional Requirements (constraints)

### Management

Due to this being a student-developed project, there will be no costs associated with this project. This project will be developed during the 2016 Spring Semester with a completion date during 2016 April. Periodic reviews and testing of the listed functions should be conducted to ensure that each phase of the prototype meets the requirements.

### Technical

The project must be run on a standard Windows 7 / 10 system. The development environment will consist of a standard Windows 7 / 10 system and a standard OS X system. The project will be written in Java. MySQL Workbench will be used to handle the temporary database.

### Performance

Efficiency and reliability are relatively important for this project and are thus at minimum a minor concern. Thus there should be a reasonable effort to the ensure that the temporary database is accurate to the PDF file from which it is created and that there are no major errors in the database query system.

### Security

Security is not a major concern for this prototype as the original document is freely available on the internet.

## System Evolution/Maintenance

This project should be written with expansion in mind. This prototype will be designed in such a way as to make adding degree plans a relatively simple action so that this software can be utilized by advising faculty in other departments within the university. The application will also be designed in such a way as to allow portability to various platforms.

# Other Deliverables

At the end of the semester (April 2016) a project plan, requirements specification document, test plan, test report, prototype, user manual, design diagrams, and a final report will be delivered to the customer.

# Risks

The risks involved in the development of this system include illness of team members, procrastination, changes in requirements, compatibility constraints, learning curve for use of development applications, and a lack of communication. Students in this development team are also involved in other research groups and regular meeting times could create scheduling conflicts. An effort by team members to keep other lines of communication open, such as utilizing phone applications like What’s App and e-mail, will reduce these risks.

# Glossary

B.A. Degree Plan – a degree plan within the computer science department that allows students to take less rigorous mathematics courses in exchange for a number of credit hours in humanities/literature courses and fourteen semester hours of a foreign language.

B.S. Degree Plan – a degree plan within the computer science department that requires students to take more rigorous mathematics courses.

Course – an available college course during a given semester; can be either traditional or on-line.

Degree Plan – a roadmap of the required courses for a particular major within a collegiate institution.

Fundamental Core / Electives – courses required of all university students independent of major.

GUI – the graphical user interface that allows a person to interact with the software easily by utilizing a mouse and keyboard.

Major – a student’s desired field of study within the university.

Major Core / Electives – courses required by a specific department for students graduating with a degree from that department.

On-line Course – college class in which students access course information through an on-line portal, such as D2L, and are not required to present to a physical location.

Traditional Course – college class in which students are required to attend a lecture on specific days and times in a physical location.

# References

1. Sommerville I., Software Engineering 9th edition, Delhi, Pearson Education, Ltd, 2011, 84-88.
2. Uyttewaal E., “Microsoft Project: Plan Better with Microsoft Project”, New York, TechNet Magazine, February 2012.

# Consultants

1. Stringfellow, Dr. Catherine, Customer, Open Classes Project, MSU Wichita Falls, Fall 2015.
2. Kundert, Carley, Possible Future Customer, Open Classes Project, MSU Wichita Falls, Fall 2015.