**Software Requirements Document**

**By**

Erica Castro-Morrison

Jonathan Mahoney

Akeem Wilkins

Norfolk State University

March 25, 2019

**Table of Contents**

Table of Contents..........................................................................................1

Revision History............................................................................................2

1. Introduction......................................................................................... 3

1.1 Purpose.............................................................................................. 3

1.2 Intended Audience and Reading Suggestions................................... 4

1.3 Product Scope.................................................................................... 4

1.4 References..........................................................................................4

2. Overall Description............................................................................. 5

2.1 Product Perspective........................................................................... 5

2.2 Product Functions...............................................................................5

2.3 User Classes and Characteristics……………………………............... 7

2.4 Operating Environment....................................................................... 7

2.5 Design and Implementation Constraints………………………………. 7

2.6 User Documentation........................................................................... 8

3. External Interface Requirements........................................................ 8

3.1 User Interfaces................................................................................... 8

3.2 Hardware Interfaces........................................................................... 9

3.3 Software Interfaces............................................................................. 9

3.4 Communications Interfaces.............................................................. 10

4. System Features...............................................................................10

4.1 System Feature 1............................................................................. 10

4.2 System Feature 2 (and so on)...........................................................11

5. Other Nonfunctional Requirements...................................................19

6. Other Requirements..........................................................................20

6.1 Legal Requirements…………………………………………………...…20

6.2 Database Requirements………………………………………………...21

Appendix A: Glossary................................................................................. 22

Appendix B: Engineering Flowchart…………………………………………..23

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

**1. Introduction**

In today’s day and age there are numerous ways to keep track of statistics, grades, and inventory. The most important thing for college students, is their course curriculum for their specific major and concentration or track. Many schools have different ways of keeping track of their students’ evaluations of their courses. The key is to make sure that the most simple and accurate way is used to keep track of this data. Currently, there is an application being developed that is user friendly and that helps to make this process run smoothly and more efficient. Faculty and students were asked to participate in surveys and interviews so that the feedback received from both parties could be used to resolve this issue together. This process should be simple and well organized, especially with the technology that is readily available.

## **1.1** **Purpose**

The purpose of this application is to present a detailed description and to develop an electronic fully functional interactive tool of the CSC Flowchart currently being created by CSC faculty member Dr. Rasha Morsi. This tool will enhance the user’s experience and help to solve the case problem: “Most freshmen and sophomore students do not make use of the Curriculum Track Sheets to guide them through registration each semester and to plan for future semesters”. This document will provide an overview of the Electronic CSC Flowchart Application; its parameters, goals, user interface, hardware and software requirements, and its functionality.

## **1.2** **Intended Audience and Reading Suggestions**

The target audience will be Norfolk State University’s Computer Science Department Chair, faculty, students, IT staff, and other Norfolk State University administration that would be involved in the decision making process in regards to curriculum and registration processes.

## **1.3** **Product Scope**

The Electronic CSC Flowchart Application is designed to run locally and to allow students to plan, create and track their curriculum, update or modify an existing curriculum plan, and engage with their advisors. The electronic plans will be more convenient and allow students to keep on track with graduation timelines while having personal curriculum more transparent as to what is needed to stay on course. The data will be held in an database with access being given to advisors and administrators to help keep students aligned with their graduation goals from freshman to senior year.

## **1.4** **References**

Butler, C., Ing, J. and Lauk, E. (2019). [online] Available at: https://nsu.blackboard.com/bbcswebdav/pid-2448758-dt-content-rid-39172596\_1/courses/CSC-380-01-182/CSC380RequirementsDocument.pdf [Accessed 30 Mar. 2019].

Morsi, R. (2019). [online] www.Nsu.edu. Available at: https://www.nsu.edu/getattachment/Academics/Faculty-and-Academic-Divisions/Schools-and-Colleges/College-of-Science-Engineering-and-Technology/Departments/Engineering/Programs/B-S-Electrical-Electronics-Engineering/EEE\_curriculum-CalReady\_2019.pdf.aspx?lang=en-US [Accessed 30 Mar. 2019].

Smith, C., McFall, C. and Dodson, A. (2019). [online] Available at: https://nsu.blackboard.com/bbcswebdav/pid-2448758-dt-content-rid-39172595\_1/courses/CSC-380-01-182/CSC380ProjectRequirementsDocument.pdf [Accessed 30 Mar. 2019].

**2. Overall Description**

## **2.1** **Product Perspective**

The product deliverable will be a desktop application and database connected to a local server. This will be an application that initially will be stand alone until it can be implemented into the Norfolk State University registration process. Future work will be a mobile application that will eventually be connected to the Norfolk State University department server as a supplemental application to be integrated into the the department’s registration process.

## **2.2** **Product Functions**

File:

* User Registration: Users registers for an account.
* Login: User logins with username and password.
* New Plan: Creation of a new curriculum plan.
* Open: Loads an existing plan.
* Open Recent: Loads one of the displayed, recently opened files.
* Close Project: Closes the currently open plan.
* Properties: Displays some properties of the plan(such as the major, track, previous courses, or gpa).
* Notes: Allows user to add notes to plan.
* Import Database -> Edges List: Loads the students information from the Database (MySQL).
* Generate -> Flowchart: Creates a Dynamic Flowchart from saved plan.
* Generate -> Report: Creates a report from saved plan.
* Email: Allows the user to email report or flowchart.
* Rename: Renames the curriculum plan.
* Save: Saves the curriculum plan without changing its name or directory.
* Delete: Deletes the current plan.
* Exit: Exits the application.

Help:

* Check for Updates: Displays the plugins that can be updated to newer versions
* About: Displays the developers, their contact information and other info.
* Options: Here the user can change settings.
* Quick Start User Guide: Redirects the user to quick start document

## **2.3** **User Classes and Characteristics**

* Typical Users: Students who want to plan, modify, and track their curriculum.
* Advanced Users: Faculty and Administrators who have role privileges to track students progress, advise changes, and override decisions.
* Professional Users: IT professionals with developer and security administrator privileges who can troubleshoot issues, reset the application, and do maintenance.

## **2.4** **Operating Environment**

The application will run on machines with Windows Operating Systems initially and later expansions will be ported to a mobile application to run on both android and apple devices.

## **2.5** **Design and Implementation Constraints**

The original design will be limited to Windows based desktop applications without any functionality for Mac OS or Linux Operating Systems. There also will not be a way for the application to be used on mobile devices until the application can be ported in the future works portion of the design. Having the application on a local database will also limit updates as well as how user information can be accessed by faculty and administration. For now the user will be able to save their curriculum plans and possibly be able to download and email the reports and flowcharts to their advisor or print out hard copies to use in joint planning.

## **2.6** **User Documentation**

A quick start guide will be added into the help function of the application as well as added to the appendix of this Software Requirements Document to help users navigate the system.

# **3.** **External Interface Requirements**

## **3.1** **User Interfaces**

The electric CSC flowchart application will be very similar to the appendix image of the engineering departments plan. The ideal graphical user interface will be able to generate an automated version of this for Computer Science. Each course shown on the map will be a clickable component to show the needed prerequisites. For the different tracks this graphic will be displayed in a specific tab, there will be multiple tabs available so the user can flip through the given tracks and see which one they are most on pace to complete. The loading of course for this prototype will be done through checkboxes on an initial page. These checked courses will then be put into the flowchart format. The courses taken will be shaded and the course that needs to be taken next will be indicated in some format most likely a highlighted approach. Other features for the graphical interface are unclear at the moment, but could include line colored routes, other button components such as a print, a text area and more color in general.

## 

## **3.2** **Hardware Interfaces**

The hardware devices used to run the software will be a PC. The application will be runnable from any Windows desktop. The user will click the given buttons on the screen with the mouse. This data will then be collected and fed into a database for storage. Commonly for the software to change pages or quit the user will either click an enter or exit button, though in some case may hit the return button on the keyboard. The initial prototype will not be a mobile application, but future works would ideally head in that direction.

**3.3** **Software Interfaces**

The product is to be built in java on eclipse. The libraries of windowBuilder and javafx shall be imported to optimize the creation. The java application will have a direct connection to a mySQL database. The application will share the data collected from the java GUI application with the database for storage. A save component will be built so that the user can save their given data to the database. This data will contain their full name, student ID, a registered password, and all of their collected course data. The components of the interface will most likely be global variables which can have its hazards. There will be objects created such as a student object that contains the data being collected. The variables of these objects will be private and use the proper methods to access their data. A course may also be its own object with attributes of name and course number. The student object will contain a list attribute of class objects.

## **3.4** **Communications Interfaces**

The only line of communication is that of the interface to the SQL database. Future works may be done online or the need of email addresses for professors to communicate to the student through the application. The data will be saved to the database in Last Name, FIrst Name, student ID, course list format.

**4. System Features**

**4.1 Plan courses per year and semester**

**4.1.1 Actors**

Customers, System admin

**4.1.2 Description**

This will give a general idea of what courses are in the near future. This will also give a general idea of the expected graduation date. The user can use this data to plan all of their course until the graduate or plan their courses for each semester by looking at the courses that are completed and or course that are needed to be taken.

**4.1.3 Data**

Available courses for core/major needed for each year and semester.

**4.1.4 Stimulus/Response Sequence**

In the instance that the user wants to know what courses are needed to complete for each semester and year to plan. The user will be able to select from freshman and senior year. The user could also select by fall and spring semester.

**4.1.5 Functional Requirement**

REQ-1: User will log in using their student number and password.

REQ-2: The user will select what semester/year they are planning their courses.

REQ-3: The application will display the courses that are required to take for that specific semester.

REQ-4: If user tries to select a different course for that semester/year, the application will notify they need this permission overridden by the department/advisor to complete this request.

REQ-5: If the user continues with the selected courses the user has the option to email/print/save the selected courses that are enrolled for that semester/year.

REQ-6: Once the user has completed the task they can select the tab to logout from the application.

**4.2 Bring up previous courses taken (inputted from user until can be uploaded from server)**

**4.2.1 Actors**

Customers, System admin

**4.2.2 Description**

This feature should input all courses that are transferred from all colleges and universities. The original name of the course you took at your previous campus and the current course that is similar to the current university you are attending. This should combine the credit hours and curriculum from all insitutitions.

**4.2.3 Data**

This data will display the course that are taken from previous semesters and also display courses that are transferred from other institutions.

**4.2.4 Stimulus/Response Sequence**

If user would like to see what courses they have completed the applications will display courses that are taken at their institution. It will also show what courses are transferred from other institution and substituted to similar courses at the current institute.

**4.2.5 Functional Requirement**

REQ-1: User will log in using their student number and password.

REQ-2: The user will select what semester/year they are planning their courses.

REQ-3: The application will display the courses that have been completed this will also include the transferred courses.

REQ-4: The user can select to print/save/print this output of the application of the completed courses.

REQ-5: Once user is complete they can click on the tab to logout from the application.

**4.3 Visually appealing and easy to follow pathway from freshman to senior year**

**4.3.1 Actors**

Customers, System admin

**4.3.2 Description**

The students/faculty should have a general idea of the path they are taking from freshman and senior. This should take stres

**4.3.3 Data**

The application display will a digital image that is visually appealing and easy to follow pathway that will guide the user from freshman to their graduation semester.

**4.3.4 Stimulus/Response Sequence**

If the user would like to see the proposed path of their college career, the application will display an actual graphical image that will show where the current progress and show what courses remain by highlighting the path that is left.

**4.3.5 Functional Requirement**

REQ-1: User will log in using their student ID number and password.

REQ-2: The application will display the current progress of where the user is in their curriculum path. This will also display the rest of the path of courses needed complete their degree program.

REQ-3: Once user is complete they can click on the tab to logout from the application.

**4.4 Show remaining/taken credit hours and GPA**

**4.4.1 Actors**

Customers, System admin

**4.4.2 Description**

This function will show students their remaining credit hours that are left to graduate. The application will also provide the overall GPA calculation from your grades at your institution.

**4.4.3 Data**

The application will display the remaining/taken credit hours and will also show a calculated GPA from the courses that have been taken and graded.

**4.4.4 Stimulus/Response Sequence**

If the user would like to see the remaining/taken credit hours the user can select this option. Then the application will display this information. Also if the user would like to see their current GPA they can also select this option and the application will display the current GPA.

**4.4.5 Functional Requirement**

REQ-1: User will log in using their student ID number and password.

REQ-2: The user can select credit hours. This will display the remaining/taken credit hours

REQ-3: Once user is complete they can click on the tab to logout from the application.

**4.5 Separate major/core courses from electives**

**4.5.1 Actors**

Customers, System admin

**4.5.2 Description**

This application will help divide your major and core courses from the electives that are required for the student to graduate.

**4.5.3 Data**

This data will include the major/core courses from electives that are required for each major.

**4.5.4 Stimulus/Response Sequence**

If the user would like to see the difference of desired majored and core courses that are required to satisfy to graduate, the application will display the path and required courses that is needed for the user to complete and graduate within the designated time.

**4.5.5 Functional Requirement**

REQ-1: User will log in using their student number and password.

REQ-2: The user will select what semester/year they are planning their courses.

REQ-3: The application will display the courses that are needed for their desired major and core courses need to complete as well.

REQ-4: The user can select to print/save this output of the application of the completed courses.

REQ-5: Once user is complete they can click on the tab to logout from the application.

**4.6 Be able to see which electives you qualify**

**4.6.1 Actors**

Customers, System admin

**4.6.2 Description**

This will display your core course needed to be taken. The core courses are your English, Math, Science, Humanities, and cultural elective courses. Each section will display a list of classes that qualify for each subject.

**4.6.3 Data**

The application will display the qualified courses that are desired to complete for desired major.

**4.6.4 Stimulus/Response Sequence**

The user will be able to select their major, then the application will output the electives that the user is qualified to take and complete to satisfy their completion to graduate from there institute.

**4.6.5 Functional Requirement**

REQ-1: User will log in using their student number and password.

REQ-2: The user will select what is their desired major.

REQ-3: The application will display the electives the user qualify to take.

REQ-4: The user can select to print/save/print this output of the application of the completed courses.

REQ-5: Once user is complete they can click on the tab to logout from the application.

**4.7 The system will show the prerequisites before taken the selected course.**

**4.7.1 Actors**

Customers, System admin

**4.7.2 Description**

Some 200 or 300 level course require you to take a intro course before taking the more advanced course. This will prevent the professors from repeating or reteaching information that students should’ve learn in the intro level courses. The application will display the courses that are needed before registering for the higher level courses. Also the student will not be able to register for that course unless the advisor/department overrides it.

**4.7.3 Data**

This data will be available once the user selects and 200 or above level course. The application will display the desired prerequisites need to move to the next course that was selected.

**4.7.4 Stimulus/Response Sequence**

Once the user have selected the course to enroll for the selective semester/year. The application will output the prerequisites that is needed before enrolling to the selected course.

**4.7.5 Functional Requirement**

REQ-1: User will log in using their student number and password.

REQ-2: The user will select what semester/year they are planning their courses.

REQ-3: The user will select the course they are looking to enroll

REQ-4: The application will output the prerequisites courses that is needed before enrolling into the course.

REQ-5: If the user has not completed the prerequisites, the application will notify the user that they must get this course overridden by the department/advisor to continue to take this course.

REQ-6: If the user meets the prerequisites, the user will be able to select the course for the selected semester/year.

REQ-4: The user can select to print/save this output of the application of the selected courses or the prerequisites need to complete to advance to the next course.

REQ-5: Once user is complete they can click on the tab to logout from the application.

**4.8 Username/password needed to logon**

**4.8.1 Actors**

Customers, System admin

**4.8.2 Description**

This simple feature is basically needed to protect students and the institution from outside parties who are not authorized to gain information. This is basically a unique security feature for the user to create their own password and username. The application will recommend you keep passwords and username to yourself. Also the application will advised on the strength of your password and provide ideas to create unique passwords that are not easy for outside user to solve.

**4.8.3 Data**

The application should have the password and login information saved in a secured location. So the user can gain access to their courses and data.

**4.8.4 Stimulus/Response Sequence**

In order for the user to gain access to the application of their own data. They must login with their username/password that they created. This information was created when they first created their profile on this application.

**4.8.5 Functional Requirement**

REQ-1: If the user never login to this application. The user will select the tab to create an account.

REQ-2: The user will create their username/password after completing the general information

REQ-3: If the user has completed this general information before they would just need their username/password to gain access to their data on the application.

REQ-4: Once user is complete they can click on the tab to logout from the application.

**5. Other NonFunctional Requirements**

**5.1 Log out after 10 minutes inactivity.**

The system will set a time limit that will automatically time and log you out due to inactivity on the application. To protect the user from leaving their data open to a wandering eye.

**5.2 Legend of different colors of routes of path to graduation.**

The legend will have different colors that will guide and make it more user friendly. So the user can understand and see the actual routes of path to your actual graduation completion.

**5.3 Highlight courses of completion.**

*Separated by (planned, pre-registered, taken)*

The application will highlight the courses of completion. The application will have different colors of highlight for planned, pre-registered, and taken. There will be a key provided that define the different colors if the user forgets the different colors for each category.

**5.4 Department/advisor contact information.**

Basic contact information if the student needs to speak to the department or to the advisor directly.

**5.5 A way to put notes for user.**

The user will have away to add notes throughout the entire application. Also the notes are helpful to keep track of notifications to students and advisor.

**5.6 The system should be user friendly**

This system is simple and straightforward. The system will take the user less than 24 hours to understand the entire application using the quick start guide found in the help section.

**5.7 Should be able to keep track of user’s courses**

The application will track of the user’s courses. This will help the system to be more user friendly and prevent any confusion of application.

**6. Other Requirements**

The other requirements will discuss the necessities regarding the handling of the confidential student information, their privacy rights and the maintenance of the database.

**6.1 Legal Requirements**

All functions of the system will follow and abide by the stipulations set forth by university, state and federal law in accordance to FERPA (Family Education Rights and Privacy Act).

**6.2 Database Requirements**

The amount of storage for the database is one of the most essential elements to consider. The database will be pre constructed to hold the given objects and data that is embedded within them.

**Appendix A: Glossary**

CSC: Computer Science Course.

Curriculum: The courses and requirements for a given degree.

Database: An organized collection of data on a computer.

IT: Information Technology.

JAVA: an object-oriented programming language.

Linux : Open source operating system.

Mac- OS: Operating system created by Apple.

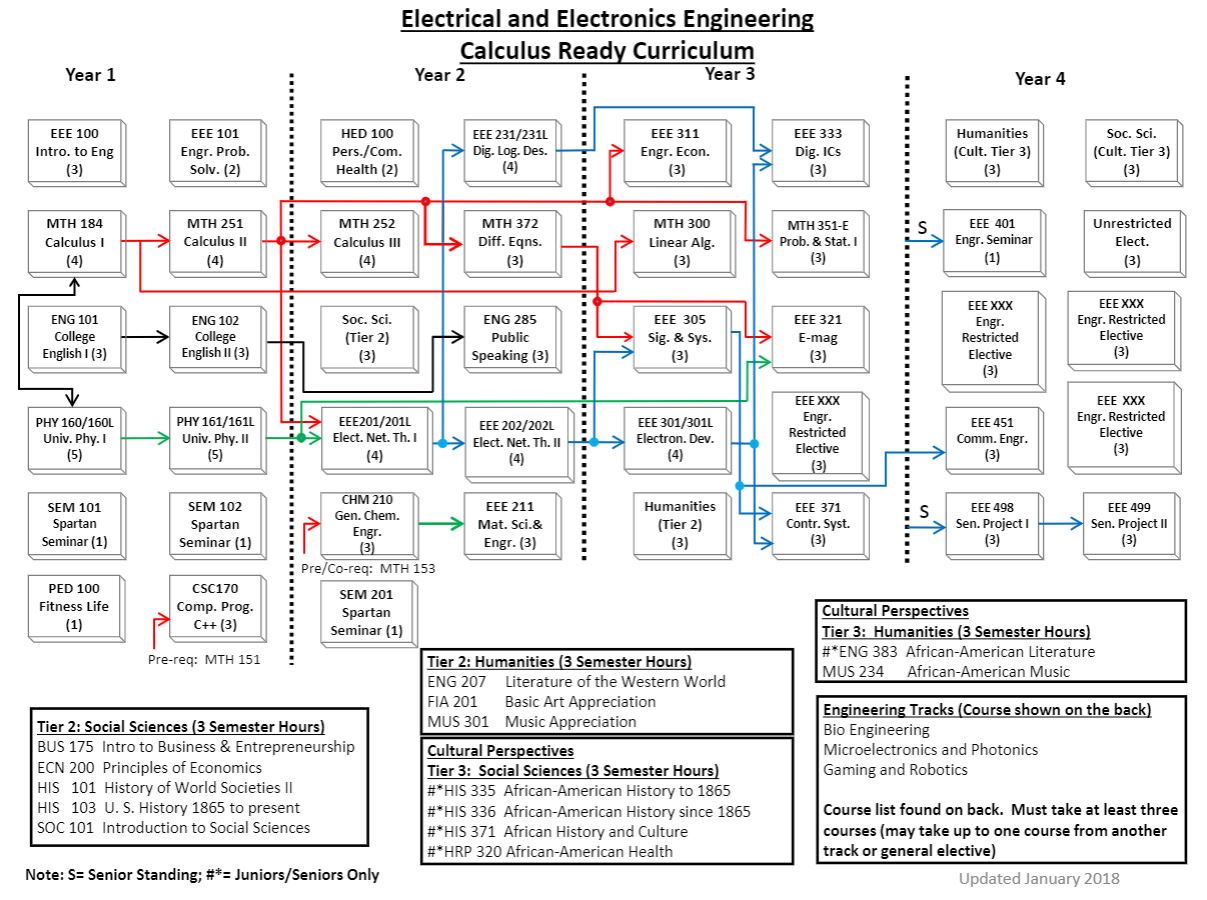
MySQL: Databasing language.

Object: Datatype formulated by the software engineers.

Prerequisite: Baseline coursework needed prior to taking degree specific coursework.

Prototype: A mock draft of initial software.

Track: Degree driven outline of courses.

**Appendix B**

Example Engineering Department Flowchart