Software Requirements Specification

For

Dragon Course Scheduler

Prepared by

Nathan Gelfant

Kevin Huang,

Stan Kolakowski

Mark Scheid

At

Drexel University

SRS Revisions History

Date	Description	Version	Editors
1/28/13	Created the document.	1.0	Mark Scheid, Nathan Gelfant, Kevin Huang, Stan Kolakowski

January 25, 2013

Contents

1 Introduction

1.1 Purpose

This document specifies all the software requirements for Dragon Course Scheduler. These requirements directly relate to the features, functionality, performance, attributes, constraints and interfaces of the system. This program will address the lack of functionality in the current term master schedule, taking into account the users' major, concentrations, and course history.

1.2 Intended Audience

The intended audience for this document are Drexel faculty members, stakeholders, developers and testers.

1.3 Scope

The project's scope is to fill a need for additional features from the Drexel's term master schedule. It is implemented for Computer Science major students in Drexel to schedule their classes for multiple terms, taking in tracks information, however is designed to be expanded to include the entire student body. Through the scheduling process, the system will depend on user input data on class taking history and Drexel's term master schedule as well as Drexel Computer Science department tracks information.

1.4 Definitions, Acronyms and Abbreviations

1.4.1 Definitions

Browser: A piece of client side software, such as Firefox or Internet Explorer, capable of displaying web pages written in HTML, CSS and JavaScript code.

Class Selection Interface: a web page through which the user views the list of available classes for the selected term based upon their previously entered courses

Concentration: also named tracks, will be major specific information required or defined by Drexel

Database: A database is a structured collection of data that is organized for later retrieval.

Dependent classes: Classes which require the course in question as a prerequisite.

Prerequisites table: A table in the database that is populated based on Drexel Course Catalog, detailing what courses are necessary before one can register for a given class

Program: The java backend that manages data flow between the interfaces and the database

Selected Schedule Interface: A web page through which the user views their current plan of classes for the selected term.

Server: A content hosting system that will deliver content to end users and host aggregated information.

TMS Synchronizer: The server-side tool that automatically updates the database with available course information and major concentration definitions.

User Background Interface: A web page through which the user can input their major, concentrations, and the list of classes they have already taken.

User History Object (UHO) The user history object contains a data structure capable of holding information containing all previous classes completed by the user, as well as the currently selected list of classes for the scheduled terms.

1.4.2 Abbreviations:

DCS Dragon Course Scheduler
CRN Course Reference Number
JRE Java Runtime Environment
JVM Java Virtual Machine
TBD To be dated
TBA To be announced
TMS Term Master Schedule

2 Overall Description

2.1 Product Perspective

Drexel students need to schedule their classes carefully to ensure graduation in a timely fashion, and at times prerequisites and incomplete tracks may often slow down one's graduation plan. The program is intended to help student to schedule classes over multiple terms, helping make the process of generating and planning their schedule easy.. Once the schedule is done, the student can export the schedule into industry stand calendar format.

2.2 Product features

- Parse the term master schedule to fill a database with class information.
- Provide a class schedule based on term (Fall, Winter, Spring, and Summer).
- Interactively build your class schedule.
- Builds class list determined by user's entered class history.
- User specified filter highlights computer science tracks.
- Allow user to pick preference between day or night time classes.
- Display possible elective choices.
- Output a calendar of the class schedule.

2.3 User Characteristics

The users of the class schedule would primarily be computer science students. The targeted audience could easily navigate through the interface, but the user would have to be familiar with what classes they have previously taken and what track they plan to take in the future.

2.4 Requirements Apportioning

The priority levels for the requirements:

Priority	Description
1	The highest priority level; all requirements of this level must be completely satisfied and verified before the release of this software.
2	All requirements of this level are not expected to be verified. However, the unfulfillment of these requirements must not jeopardize the program. These requirements shall be met in the next minor update of the software.
3	These requirements are not expected to be completely met in the current release of DCS. They are however expected to be met and verified in the next major release of DCS.
4	The lowest priority level; all requirements of this level are not within the scope of the current design. These requirements are included to show where the DCS software will change in the future.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 SQL Database

3.1.1.1 Purpose: The database shall store class information which facilitates the program filtering based on the user history object to display relevant class options to the user interacting with the class selection interface.

The database will contain class information that acquired from Drexel's term master schedule. This database will then be used by several other classes including : history

3.1.1.2 Input: The database shall be generated by taking information from the Drexel term master schedule. As the user interacts with the program the updated user history object will generate a request for relevant classes.

3.1.1.3 Output: The output is a list of classes (String) to be parsed by the program and displayed to the class selection interface.

3.1.1.4 Data Format: Database interactions will take place in the form of SQL calls.

3.1.2 Java Virtual Machine (JVM)

3.1.3.1 Purpose: The JVM runs the program, facilitating the interaction between the database, the program, and the various interfaces of the program

3.1.2.2 Input: The JVM takes the program itself as its input

3.1.2.3 Output: The JVM, in running the program, generates the

3.1.2.4 Data Format: Data will be limited to the program, which the JVM shall load and run.

3.1.3 Server

3.1.3.1 Purpose: The Server presents the various interfaces to the user.

3.1.3.2 Input: The server takes input from a user interacting with a hosted web page through a browser to pass information to the website. It will likewise take input from the program to update the web page displayed.

3.1.3.3 Output: The server will output a web page for the user.

3.1.3.4 Data Format: The server will output a web page in standard HTML, CSS and JavaScript formats. The server will receive as input CSV format information for use in the displayed tables.

3.1.4 TMS Schedulizer

3.1.4.1 Purpose: The data collector will be used for collecting data from Drexel's TMS system.

3.1.4.2 Input: Raw data from Drexel TMS website

3.2 Functional Requirements

3.2.1 Entering User History

3.2.1.1 A drop down will allow the user to select the major they are pursuing at Drexel. As of this time, only the CS major is officially supported. **Priority 1**

3.2.1.2 A dialogue will allow the user to enter his history as a list of comma-separated course IDs. **Priority 1**

3.2.1.3 If a course has been passed, but received below a C-, it can be indicated by preceeding it with an asterisk to denote it cannot be used for prerequisites. **Priority 2**

3.2.1.4 Alternatively, entries in the same format can be uploaded as a file. **Priority 2**

3.2.1.5 The program shall retrieve information about the selected major from the database. **Priority 1**

3.2.1.6 Depending on the selected major, a list of concentrations will be displayed. **Priority 1**

3.2.1.7 The user can select which concentrations he is pursing by clicking on the desired choices. **Priority 1**

3.2.1.8 The user can click on a selected concentration to unselect it. **Priority 1**

3.2.1.9 The user will select which terms he wishes to plan for. **Priority 1**

3.2.1.10 The program will create a User History object, and use the database to generate a list of classes. **Priority 1**

3.2.1.12 This list will contain all classes the user is eligible for taking in the first term selected, and will be presented in the Class Selection Interface as a list of classes. **Priority 1**

3.2.2 Add Class

3.2.2.1 From the Class Selection Interface the user will be able to select a course by clicking the appropriate CRN. **Priority 1**

3.2.2.2 Upon adding a class, the program will automatically update the User History Object to include the class selected, thus allowing the user to add dependant classes in subsequent terms. **Priority 1**

3.2.2.3 Based upon this addition the program will update the existing list of classes, removing courses with the same ID or timeslot. **Priority 1**

3.2.2.4 Alternatively, this same update will highlight classes that share a timeslot with classes for this term. **Priority 2**

3.2.2.5 In this case, upon adding a class, the program will alert the user if the selected classes conflicts with the times of other classes. **Priority 2**

3.2.3 Remove Class

3.2.3.1 From the Class Selection Interface or Selected Schedule Interface the user will be able to select a course by clicking the appropriate time-slot on their schedule display. **Priority 1**

3.2.3.2 The program shall warn the user if removal of the class will interfere with selected courses for other terms. **Priority 2**

3.2.3.3 Upon removing a class, the program will automatically update the User History Object to remove the class selected, thus removing dependent classes from the available list in subsequent terms. **Priority 1**

3.2.4 Change Selected Term

3.2.4.1 Class Selection Interface

3.2.4.1.1 The user will select another available term from the menu. **Priority 1**

3.2.4.1.2 The program will update the current interface to display relevant information from the newly selected term. **Priority 1**

3.2.4.1.2 If the Class Selection Interface is in use, the program will use the User History Object to request a list of user-eligible classes from the database. **Priority 1**

3.2.4.1.3 If the Selected Schedule Interface is in use, the program will display the classes currently scheduled for the newly selected term. **Priority 1**

3.2.5 View Schedule

3.2.5.1 In the Class Selection Interface, the user will be able to select this option from the menu. **Priority 1**

3.2.5.2 Upon selection, the program will display the Selected Schedule Interface. **Priority 1**

3.2.6 Select Classes

3.2.6.1 In the Selected Schedule Interface, the user will be able to select this option from the menu. **Priority 1**

3.2.6.2 Upon selection, the program will display the Class Selection Interface. **Priority 1**

3.2.7 Output Schedule

3.2.7.1 Print Schedule

3.2.7.1.1 The user will select this option from the menu. **Priority 1**

3.2.7.1.2 The program will tell the browser to print the page, allowing the user to select a printer or file printer to save their selections. **Priority 1**

3.2.7.2 Export schedule to CSV

3.2.7.2.1 The program will output into a CSV file which can be used for importing to Google Calendar and Apple iCal. **Priority 2**

3.2.7.2.2 The program will output into a CSV file which can then be used in other workflow or applications. **Priority 2**

3.3 Software System Attributes

3.3.1 Availability: As web based product, the product is designed to be available around the clock, limited only by server and database stability.

3.3.2 Expandability: While the program is initially optimized for use by CS majors, it is designed such that simply expanding the prerequisites table and adding options to the majors list could allow its use to extend up to the entire Drexel student body.

3.3.3 Maintainability: The program is designed with high levels of encapsulation to ensure that changes to the code can be accomplished with minimum effort and side-effects.

3.4 Non-functional Requirements

3.4.1 Drexel TMS

3.4.1.1 The TMS synchronizer will use Drexel's TMS data on a term by term basis in order to acquire course availability, CRN, date information.

3.4.1.2 The TMS synchronizer will use data from Drexel TMS to determine multi-term schedule availability.

3.4.1.3 The TMS synchronizer will use data from Drexel TMS to build internal database for course selection and class filtrations.

3.4.2 Drexel Computer Science Department Track information

3.4.2.1 The TMS synchronizer will use Drexel Computer Science Department's Tracking information to map the concentration information in the software.

3.4.2.2 The track information will be put into a static database which only get updated once a year

3.4.2.3 The track information will be used to highlight courses as the subclasses of all available courses.

3.4.3 Referential Links

3.4.3.1 Drexel TMS https://www.**drexel**.edu/webtms

3.4.3.2 Computer Science Tracks information https://www.cs.drexel.edu/undergraduate/tracks

3.4.4 Internet connectivity

The client needs to have access to Internet in order to use the program.

3.4.4 Web Browser with Java JRE

The program requires the client to have a web browser with Java JRE capability.

3.5 Design Constraints

3.5.1 Privacy concerns: Due to the private nature of university records, the program requires the user to input data regarding their class history, rather than interfacing with such records directly and simplifying the scheduling system.

3.5.2 Access constraints: As an unofficial Drexel product, we do not have direct access to the database of class information. As such the product must be manually updated, and thus has the additional overhead of information gathering, and limitation on the accuracy of the data.4 User Interface

Below are primitive outlines of the proposed interface screens. Not included

4.1 User Background Interface

Dragon Course Scheduler

Please enter all classes you have completed with a C- or higher below:

CS171, CS172, CS260, CS265, CS270, MATH121, MATH122, MATH123

Please enter your major code (i.e. CS for computer science)

CS

Please select your tracks from the list below

Numerical analysis	Algorithms and Data Structures
Artificial Intelligence	Computer and Network Security
Human Computer Interaction	Computer Graphics and Vision
Game Development and Design	Numeric and Symbolic Computation
Programming Languages	Software Engineering
Systems	

Winter	Spring	Summer

(above are checkboxes)

4.2 Class Selection Interface

Dragon Course Scheduler

Concentrations selected: HCI, Game Design **Concentration requirements: CS337 CS38** **Winter Term Courses: CS350, CS281, CS121** **Spring Term Courses:** **Summer Term Courses:**	**Menu (Change term, export, view schedule, etc)**

Currently Planning: Winter term

CRN	Class	Title	Timeslot	Instructor	Prereq for
40123	**CS337**	**Human Computer Interaction**	**MWF 12-1**	**Salvage**	**CS338**

Winter Term Weekly Schedule

Time	M	T	W	R	F
8	x				
9	x				
10					
11		x			
12		x			x
1					
2				x	

4.3 Selected Schedule Interface

Dragon Course Scheduler

Concentrations selected: HCI, Game Design **Concentration requirements: CS337 CS38** **Winter Term Courses: CS350, CS281, CS121** **Spring Term Courses:** **Summer Term Courses:**	**Menu (Change term, export, view schedule, etc)**

Currently Viewing: Winter term

CRN	Class	Title	Timeslot	Instructor	Prereq for
41293	CS350	Software Design	MWF 2-3	Salvage	CS451
40288	CS281	Systems Architecture	R 6:30-9:20	Johnson	CS282
39882	CS121	Computation Lab	TR 12:30-2:00	Schmidt	CS122

Winter term Schedule

Time	M	T	W	R	F
8	x				
9	x				
10					
11		x			
12		x			x
1					
2				x	

5. Use cases

5.1 Enter User History

5.1.1 Enter major

5.1.1.1 A drop down will allow the user to select the major they are pursuing at Drexel. As of this time, only the CS major is officially supported.

5.1.2 Enter completed courses

5.1.2.1 A dialogue will allow the user to enter his history as a list of comma-separated course IDs.
5.1.2.2 If a course has been passed, but received below a C-, it can be indicated by preceding it with an asterisk to denote it cannot be used for prerequisites.
5.1.2.3 Alternatively, entries in the same format can be uploaded as a file

5.1.3 Enter Concentration Selection

5.1.3.1 The program shall retrieve information about the selected major from the database.
5.1.3.2 Depending on the selected major, a list of concentrations will be displayed
5.1.3.3 The user can select which concentrations he is pursing by clicking on the desired choices.
5.1.3.4 The user can click on a selected concentration to unselect it.

5.2 Add Class

5.2.1 From the Class Selection Interface the user will be able to select a course by clicking the appropriate CRN
5.2.2 Upon adding a class, the program will automatically update the Class Selection interface to remove other selections of the class, and to denote the timeslot they have filled.

5.3 Remove Class

5.3.1 From the Class Selection Interface or Selected Schedule Interface the user will be able to select a course by clicking the appropriate time-slot on their schedule display.
5.3.2 The program shall warn the user if removal of the class will interfere with selected courses for other terms.
5.3.3 Upon removing a class, the program will automatically update the Class Selection interface to update prerequisites for other classes.

5.4 Change Selected Term

5.4.1 Class Selection Interface

5.4.1.1 The user will select another available term from the drop down menu at the top right of the screen
5.4.1.2 The program will update the list of available classes based to those the user is elegible for in the selected term.

5.4.2 Selected Schedule Interface

5.4.2.1 The user will select another available term from the menu
5.4.2.2 The program will update the list of currently selected classes for the selected term.

5.5 View Schedule

5.5.1 In the Class Selection Interface, the user will be able to select this option from the menu.
5.5.2 Upon selection, the program will display the Selected Schedule Interface

5.6 Select Classes

5.6.1 In the Selected Schedule Interface, the user will be able to select this option from the menu.
5.6.2 Upon selection, the program will display the Class Selection Interface

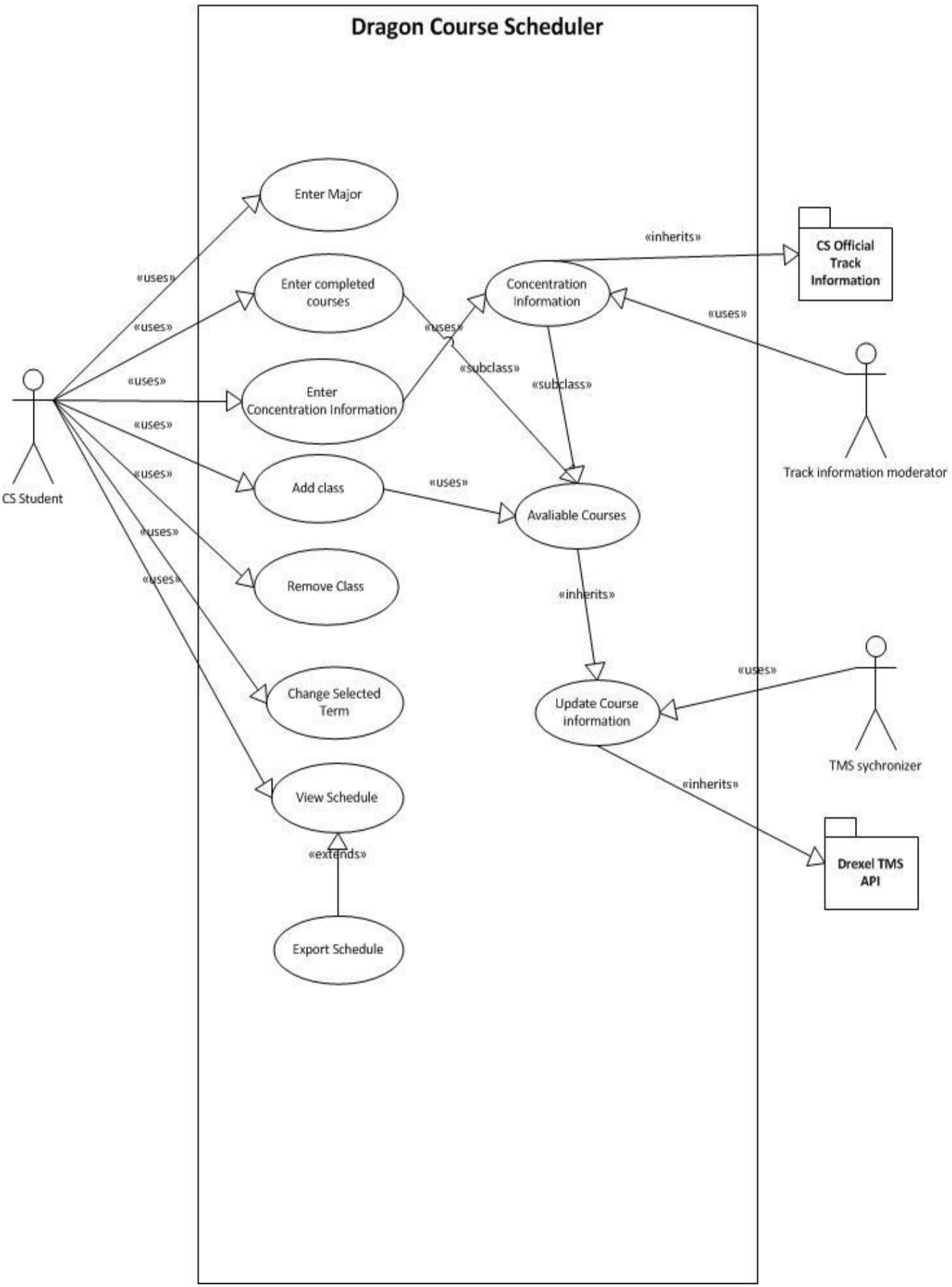
5.7 Output Schedule

5.7.1 Print Schedule

5.7.1.1 The user will select this option from the menu
5.7.1.2 The program will tell the browser to print the page, allowing the user to select a printer or file printer to save their selections.

5.7.2 Export schedule to CSV

5.7.2.1 The program will output into a CSV file which can be used for importing to Google Calendar and Apple iCal
5.7.2.2 The program will output into a CSV file which can then be used in other workflow or applications.


Dragon Course Scheduler
Enter Major
Enter completed courses
Enter Concentration Information
Add class
Remove Class
Change Selected Term
View Schedule
Export Schedule
Concentration Information
Avaliable Courses
Update Course information
CS Official Track Information
Drexel TMS API
CS Student
Track information moderator
TMS sychronizer
«uses»
«inherits»
«subclass»
«extends»