Software Requirements Specification

For

UWI SMART COURSE CHOICE

Version 1.0

Prepared by

Group Name: R.A.J.J

|  |  |  |
| --- | --- | --- |
| Ritesh Reddy | 620059221 | ritesh\_r\_reddy@hotmail.com |
| Ashle | 620056706 | ashlehenriques@gmail.com |
| Jordanne | <student #> | <e-mail> |
| Jeremy | 620057694 | jeremy\_df9@hotmail.com |

|  |  |
| --- | --- |
| Instructor: | Mr. Eyton Ferguson |
| Course: | COMP2140 |
| Lab Section: | *<place your lab section here>* |
| Teaching Assistant: | *Dean Jones* |
| Date: | <place the date of submission here> |

Contents

Contents ii

Revisions iii

1 Introduction 1

1.1 Document Purpose 1

1.2 Product Scope 1

1.3 Intended Audience and Document Overview 1

1.4 Definitions, Acronyms and Abbreviations 1

2 Overall Description 2

2.1 Product Perspective 2

2.2 Product Functionality 2

2.3 Users and Characteristics 4

2.4 Operating Environment 4

2.5 Design and Implementation Constraints 4

2.6 User Documentation 4

2.7 Assumptions and Dependencies 5

3 Specific Requirements 6

3.1 External Interface Requirements 6

3.2 Functional Requirements 6

3.3 System Model 9

4 Other Non-functional Requirements 12

4.1 Performance Requirements 12

4.2 Safety and Security Requirements 12

4.3 Software Quality Attributes 12

Appendix A - Group Log 13

Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| V.1.0 | Ritesh Reddy  Ashle Henriques  Jordanne Elliott  Jeremy-Dane Ferguson | Initial Version. Requirements Complete | 01/10/2013 |

# 

# Introduction

This document will provide an overview and in depth analysis of the requirements of the UWI Smart Course Choice System (USCC), a system to help the University students choose courses to fulfill programme requirements.

## Document Purpose

This Software Requirements Specification (SRS) document is intended to detail the UWI Smart Course Choice system (USCC). The document specifies the requirements of the USCC and the constraints under which it should operate.

## Product Scope

The UWI Smart Course Choice system is an automation of a manual and tedious process that students of the university have to do. It eliminates the need for students to consult multiple, dense sources of information manually and choose courses to take at the university. The process presents a major hurdle for students in pursuing the degrees that they desire in the shortest time possible and therefore, due to mistakes in course choices, students spend longer than expected at the university without graduating.

The USCC will eliminate the need for the manual work on part of the students, and easily allow them to make various plans based on their interest and assess their feasibility in terms of time required to complete a programme’s requirements.

## Intended Audience and Document Overview

This SRS is intended for both a technical and non-technical readership. The document includes a section on the overview of requirements for the non-technical readers and is intended to establish the user requirements for the system. Furthermore, the document includes a section on specific system requirements for technical readers to benefit from a break down and detailed analysis of the system requirements.

## Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| Term | Definition |
| USCC | UWI Smart Course Choice |
| Efficiency & Effectiveness of time spent at the university | This refers to the minimization of the time it took a student to successfully complete the desired major with the least amount of time. |

# Overall Description

## Product Perspective

The University of the West Indies offers a wide variety of programmes and courses that overwhelm first year students who are not accustomed to the university's environment. Albeit the University clearly setting out all the required information on all courses and programmes in the respective faculty's handbooks, students still find it difficult to navigate through the many pages in the handbooks laboriously and end up making uninformed decisions and course choices. These decisions have an adverse effect on the time they spend at the university in pursuit of their degree, as they might have to spend additional time having to satisfy prerequisites for courses that they did not intend to do initially.

In order to address this issue and maximize the student's efficiency and effectiveness of the time they will spend at the university, there is a need for a system that will help students figure out all the university, faculty and course requirements and the time it will take for each of their desired majors and/or minors without much effort on their part. This is exactly what the Course Planner system will accomplish: allow students to determine what courses need to be taken and in which order to take them such as to satisfy all requirements within the university in order to graduate with the desired degree.

## Product Functionality

1. The system shall allow enrolled users to login to the system.
2. The system shall allow users to select the major that they wish to pursue.
3. The system shall allow users to select any **one** of the following:
   1. A second Major
   2. A Minor
   3. Two Minors
4. The system shall allow students who still have course space to select an area of interest in order to fill those spaces
5. The system shall allow the users to view courses for the areas of interest and allows the users to assign them to the semesters.
6. The system shall allow users to view the courses required in each semester in order to achieve their desired major and (optionally) minors.
7. The system shall allow users to rearrange courses into different semesters and years, provided that the new configuration is still valid.
8. The system will advise the user on school policies regarding courses such as faculty requirements and programme requirements.
9. The system shall allow users to generate the course plan for the desired area of study.
10. The system shall ensure that the final course plan will be valid such that every course's prerequisite will be satisfied before the course has been taken.
11. The system shall solicit data from the UWI course and programme management system.
12. The system shall gather the user’s passed course list from UWI’s student management system.

## 

## 

## Users and Characteristics

The system is designed to maximize efficiency and effectiveness of the time spent pursuing education at UWI and as such the system shall ideally cater to the followings sets of users who have not yet spent any time at the university:

1. First year UWI students
2. Students who wish to attend UWI in the near future

The system shall also be useful, but to a lesser extent to non-first year UWI students as they would have already spent some time at the university and as such it is not possible to guarantee maximum efficiency of their time spent at UWI.

Further stakeholders in the system include:

* + - 1. **The administration of UWI**

The administration of UWI dictates the various security and administrative policies that the system must adhere to in order for the system to be deployed by the University.

* + - 1. **The deans of various faculties**

The deans of the faculties set out requirements at the faculty level that the system shall consider in validating its results. The dean has the final say in faculty related matters and as such any information generated by the system must be consistent with his/her decisions.

* + - 1. **The academic department heads**

The department heads are responsible for setting out course and programme requirements that the system shall consider in validating its results. The department head has the final say in course and programme related matters and as such any information generated by the system must be consistent with his/her decisions.

## Operating Environment

* The system shall be implemented as a web based application to allow users to access if via the internet.
* The system shall be compatible with a UNIX based platform.
* The system shall be able to communicate with existing systems for user authentication.

## Design and Implementation Constraints

* The system must adhere to student confidentiality policies.
* The system must not consume excessive resources.
* The system must adhere to the university's policies.
* The system must be able to run in a UNIX environment.
* The system must be secure.
* The system must not produce results that are inconsistent with other university sources.

## User Documentation

The system shall be accompanied with:

* Quick-start guide
* A screen-cast showcasing the major features.

## Assumptions and Dependencies

1. Information on courses offered at UWI is available in a digital format.
2. Information on programme requirements at UWI is available in a digital format.
3. All information is up-to-date.
4. The users are sure of at least one major choice.

# Specific Requirements

## External Interface Requirements

### User Interfaces

1. The system shall provide the user with the following screens:
   1. Welcome Screen - to provide an overview of the system
   2. Login Screen - for enrolled students to login
   3. Major/Minor selection screen
   4. Course Plan screen
2. The Course Plan screen is the most important screen and will also be the most interactive screen. It will allow users to interact (re-arrange courses) with the system and provide immediate feedback. The feedback can take the following forms:
   1. Information on the validity of the current course plan selected.
   2. Information specifying deficiencies in the current course plan, i.e., the difference between required courses for a desired programme and current plan.
   3. Advise users on faculty and university requirements for the particular programme chosen.

The feedback can be in the form of popups for important information that require user confirmation of notice or in the form of highlighted text for less vital information.

### Hardware Interfaces

1. There are no specific hardware interfaces for this system.

### Software Interfaces

1. The system shall be compatible with UNIX based operating systems.
2. The system shall be able to interoperate with the University's user authentication system.
3. The System shall interact with the University's course management system: Banner.

### Communications Interfaces

1. The system shall communicate with the University's authentication system using LDAP.
2. The system shall communicate with banner through HTTP RESTfull web services.

## Functional Requirements

3.2.1

|  |  |
| --- | --- |
| **Use Case Name:** | Login |
| **Priority** | Auxiliary |
| **Trigger** | Login Form - Submit button |
| **Precondition** | User is an enrolled student |
| **Basic Path** | 1. Allow user to enter ID number and password. 2. Encrypt ID number and password and transmit to server for authentication. 3. Authenticate user using LDAP. 4. On successful authentication, redirect user to Major/Minor Selection (Refer to use case Major/Minor Selection SRS 3.2.2) |
| **Alternate Path** | 1. On unsuccessful authentication, redirect user to login screen again. |
| **Post condition** | The user is logged in. The user's course records have been collected. |
| **Exception Path** | None |
| **Other** |  |
| **Reference** | SRS 2.2.1 |

3.2.2

|  |  |
| --- | --- |
| **Use Case Name:** | Major/Minor Selection |
| **Priority** | Essential |
| **Trigger** | Major/Minor Selection Form |
| **Precondition** | - |
| **Basic Path** | 1. User selects a Major 2. Users select any **one** of the following:    1. A Second major    2. A Minor    3. Two Minors 3. If users do not select any one of the above: they will be allowed to select an area of interest. |
| **Alternate Path** | - |
| **Post condition** | User is redirected to Course Planner Screen (Refer to SRS 3.2.3) |
| **Exception Path** | None |
| **Other** |  |
| **Reference** | SRS 2.2.3, 2.2.4 |

3.2.3

|  |  |
| --- | --- |
| **Use Case Name:** | Course Planner |
| **Priority** | Essential |
| **Trigger** | User selects desired programme(s) and (optionally) areas of interest. |
| **Precondition** | User selects at least one major. |
| **Basic Path** | 1. Displays a preliminary course plan. 2. Allows users to view courses from the chosen areas of interest. 3. Allows users to rearrange courses as desired. 4. After all rearrangements the system returns feedback on the validity and other requirements to obtain the desired programme(s) is given to the user. (Refer to SRS 3.2.4) |
| **Alternate Path** | - |
| **Post condition** | A Course Plan is generated |
| **Exception Path** | Given the combination of programme(s) it is not possible to generate a course plan due to conflicts. |
| **Other** |  |
| **Reference** | SRS 2.2.5, 2.2.5, 2.2.6, 2.2.9 |

3.2.4

|  |  |
| --- | --- |
| **Use Case Name:** | Course Plan Validator |
| **Priority** | Critical |
| **Trigger** | User re-arranges courses. |
| **Precondition** | A course plan exists. |
| **Basic Path** | 1. Validates course plan based on the course information. 2. Validates course plan based on prerequisites. 3. Validates course plan against programme requirements. 4. Validates course plan against faculty requirements. 5. Validate course plan against administrative policies. |
| **Alternate Path** | - |
| **Post condition** | Course plan is validated or invalidated. |
| **Exception Path** |  |
| **Other** |  |
| **Reference** | SRS 2.2.7, 2.2.8, 2.2.10 |

3.2.5

|  |  |
| --- | --- |
| **Use Case Name:** | Information Aggregator |
| **Priority** | Essential |
| **Trigger** | Course Validator or Course Selector Engaged |
| **Precondition** | User is authenticated |
| **Basic Path** | 1. The system shall solicit course information from UWI’s course management system. 2. The system shall solicit programme information from UWI’s course management system. 3. The system shall solicit a user’s passed course list to aid in pre-requisite checking. |
| **Alternate Path** | The user is not logged in, and only non-student specific records are solicited. |
| **Post condition** | Information for validation is aggregated. |
| **Exception Path** |  |
| **Other** |  |
| **Reference** | SRS 2.2.11, 2.2.12 |

## System Model

### Use Case View

#### Login Use Case

The login use case is activated when the user clicks login, enters his/her ID number and password. The login then can either be submitted by pressing submit or discarded by clicking cancel.

#### Fetch Student Course Pass List Extension

The fetch student course pass list extension is initiated from with the login use case. If logged in successfully, the system will query the university’s student administration system and retrieve the student passed course list.

#### Select Programme Use Case

The select programme use case is initiated when the user chooses select major from the options presented on the main screen and allows the user to choose majors and/or minors.

#### Select Major Extension

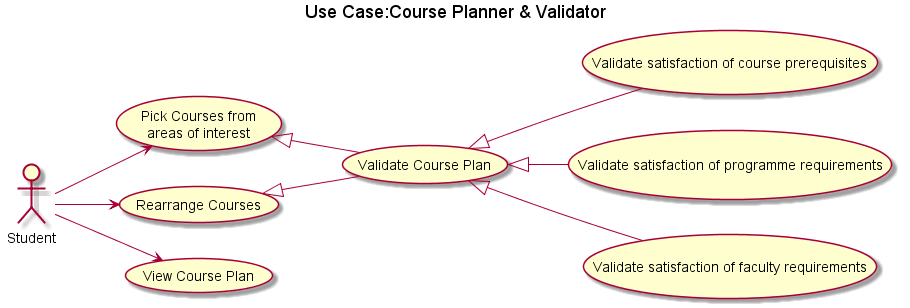
The select major extension is activated from within the select programme use case. This use case allows the user to select one or two majors that are offered by the University.

#### Select Minor Extension

The select minor extension is also activated from within the select programme use case. This use case allows the user to select one or two minors that are offered by the University.

#### Choose area of interest extension

The Choose area of interest extension is reached through the Select Programme Use case and allows the user to choose an area of interest in order to pick electives from.



#### Rearrange Courses Use case

The rearrange courses use case is activated when the user is presented a course plan and changes the sequence of courses by transposing the semester in which a course appears to another.

#### Pick Courses from Areas of Interest Use Case

The pick courses from areas of interest use case is initiated when the user chooses a course to fill gaps in the course plan from a list of courses from their desired area of interest.

#### View Course Plan Use Case

The view course plan use case is initiated by the user when he clicks “Generate Course Plan” button on the main screen. It displays the new course plan and its validity.

#### Validate Course Plan Extension

The validate course plan extension is invoked through Rearrange Courses use case and Pick courses from areas of interest use case. When invoked it validates the course plan given and provides feedback to the different modules.

#### Validate Satisfaction of Course Pre-requisites Extension

This extension is reached through the validate course plan extension and tests whether for each course the prerequisites have been satisfied before taking that course.

#### Validate Satisfaction of programme requirements extension

This extension is reached through the validate course plan extension and checks whether the given course plan includes all the courses required for the user’s desired major(s) and/or minor(s).

#### Validate Satisfaction of Faculty Requirements

This extension is also invoked via the Validate Course plan extension and checks to ensure that the course plan also complies with the faculty requirements.

### Class View

### 

# Other Non-functional Requirements

## Performance Requirements

### The system has to be responsive, i.e., the system should respond to each user interaction except the generation of course plan within 2 seconds.

### The system has to generate a course plan within 3 minutes of starting to generate a plan.

### Each validation test should not take more than 5 seconds at most.

## Safety and Security Requirements

### The system should encrypt user IDs and passwords in order to ensure safety. It should not save this information at any point in time, i.e., must be discarded immediately after use.

### The system should not save the users’ records used for prerequisite checking at any point in time, i.e., must be discarded immediately after use. Furthermore, this data should not leave the server side at any point in time.

## Software Quality Attributes

### Reliability

* The system has to have a 99.9% uptime before registration of courses begins every semester. Failure to do so will render the system useless.
* The system has to provide accurate results which are consistent with the faculty plan for each individual programmes.

### Usability

* The system must be user friendly to both enrolled students who are familiar with the university’s policies and environment and to students who are just planning for the future(students yet to enroll into the university)
* The system should provide relevant user documentation to help get the user familiar with the system and troubleshooting guides.

### Maintainability

* The system must be designed with industry standards and must follow a tested and tried architecture.
* The system must have sufficient technical documentation (both external and internal to the code) such that any competent member can understand the design and be able to implement changes or troubleshoot the system.
* The system must be designed to allow changes in the future in terms of accommodating more information with regards to courses such as course costs for example, course timetables, etc.