CS 411W Lab II – COSRICA Prototype Product Specification

Team Red

Bitaseme Mboe

CS 411W

Professor Janet Brunelle

November 23, 2014

Version 2.0

**Table of Contents**

[**1** **Introduction** 3](#_Toc404642631)

[**1.1 Purpose** 4](#_Toc404642633)

[**1.2 Scope** 5](#_Toc404642635)

[**1.3 Definitions, Acronyms, and Abbreviations** 8](#_Toc404642637)

[**1.4 References** 10](#_Toc404642638)

[**1.5 Overview** 11](#_Toc404642639)

[**2      General Description** 11](#_Toc404642640)

[**2.1 Prototype Architecture Description** 11](#_Toc404642641)

[**2.2 Prototype Functional Description** 13](#_Toc404642643)

[**2.3 External Interfaces** 18](#_Toc404642646)

[**2.3.1    Hardware Interfaces** 18](#_Toc404642647)

[**2.3.2    Software Interfaces** 18](#_Toc404642648)

[**2.3.3    User Interfaces** 18](#_Toc404642649)

[**2.3.4    Communications Protocols and Interfaces** 18](#_Toc404642650)

[**3      Specific Requirements** 19](#_Toc404642651)

[**3.1 Functional Requirements** 19](#_Toc404642652)

[**3.2 Performance Requirements** 19](#_Toc404642653)

**List of Figures**

[***Figure 1*. Current Registration Process 5**](#_Toc404642632)

[***Figure 2.* Improved Registration Process with CORSICA 6**](#_Toc404642634)

[***Figure 3*. Prototype Major Functional Component Diagram 13**](#_Toc404642642)

**List of Tables**

[***Table 1*. Real World Product and Prototype Comparison 8**](#_Toc404642636)

[***Table 2*. Description of Algorithms 17**](#_Toc404642644)

[***Table 3*. CORSICA Access Control Table 18**](#_Toc404642645)

**1 Introduction**

Course registration each semester in universities can make students anxious. There are a limited amount of seats available for courses, and the problem is that there are usually more students intending to register than there are course seats. Old Dominion University (ODU) is one school that experiences this problem. The National Center for Education Statistics states for postsecondary education, “Between 2001 and 2011, enrollment increased 32 percent, from 15.9 million to 21.0 million. Much of the growth between 2001 and 2011 was in full-time enrollment; the number of full-time students rose 38 percent, while the number of part-time students rose 23 percent.” (“Digest of Education Statistics: 2012”, 2012, para. 4). ODU’s current class enrollment system is insufficient when course enrollment reaches capacity.

ODU has more students registering for courses than there are course seats; the current system is also confusing, time consuming, and unfair. This affects faculty, advisors, schedulers, administrators, and of course, students. When a course has reached capacity, students will contact advisors for an override. Advisors have to contact the respective faculty to see if they will allow an additional student. All of this results in more time taken out of the advisor’s and faculty’s schedules to help a student get into a course, and the student would have to make-up any work missed if the student registered for the course after its start date. Figure 1 shows the current registration process.



### *Figure 1*. Current Registration Process

## 1.1 Purpose

The system that is being implemented to remedy this issue is called COmputeR ScIenCe wAit-list (CORSICA). The purpose of CORSICA is to solve ODU’s current class enrollment system inefficiency when course enrollment reaches capacity. With CORSICA, a student trying to register for a computer science (CS) course will be put into CORSICA’s wait-list system, and the student will be able to monitor enrollment status and receive notifications of their individual spot on the wait-list. Figure 2 shows an improved process flow with CORSICA.



### *Figure 2.* Improved Registration Process with CORSICA

**1.2 Scope**

The CORSICA prototype will use a test harness and have limited capabilities for the purpose of providing proof of concept. By providing proof of concept, the prototype will show mitigation of risks. The prototype will have operations for Administrators, Schedulers, Advisors, Students, and Visitors.

The prototype will have the following features: operation as a mainly automated system, fair enrollment process, notification system, wait-list statistics analysis, and wait-list seat reservations. CORSICA’s notification system will be automated and send emails to wait-listed students about their wait-list position status. Seats will be reserved for Students on a wait-list to demonstrate fairness. Table 1 shows a comparison between the real world product and the prototype.

|  |  |  |
| --- | --- | --- |
|  | **Real World Product** | **Prototype** |
| **Environments for all Users:** | **Yes** | **No**· **Will demonstrate student, admin, and scheduler users** |
| **Notification System** | **Yes** | **No**· **Will be simulated with text box** |
| **Check for available seats** | **Yes** | **Yes** |
| **Add Student to Wait-list** | **Yes** | **Yes** |
| **Drop Student from Wait-list** | **Yes** | **Yes** |
| **Fair process** | **Yes** | **Yes** |
| **Mostly automated** | **Yes** | **No**· **Will rely heavily on user interaction** |
| **Link to Banner** | **Yes** | **No**· **Will be loaded with data.txt files instead** |
| **Link to Leo-Online** | **Yes** | **No**· **Will be simulated with command box menu** |
| **GUI** | **Yes** | **Very Basic (Text System)** |
| **Seat Analysis System** | **Yes** | **No** |

### *Table 1*. Real World Product and Prototype Comparison

(This space is intentionally left blank.)

**1.3 Definitions, Acronyms, and Abbreviations**

**Algorithm -** A set of steps that are followed in order to solve a mathematical problem or to complete a computer process.

**Banner -** Old Dominion University's centralized academic and administrative records system.

**Browser -** A computer program that is used to find and look at information on the Internet.

**C++ -** A general purpose programming language that is free-form and compiled.

**Cascading Style Sheets (CSS) -** A style sheet language used for describing the look and formatting of a document written in a markup language.

**Computer -** An electronic machine that can store and work with large amounts of information.

**Database -** A collection of pieces of information that is organized and used on a computer.

**E-mail -** A system for sending messages from one computer to another computer.

**Graphical User Interface (GUI) -** A program that allows a person to work easily with a computer by using a mouse to point to small pictures and other elements on the screen.

**HyperText Markup Language (HTML) -** A computer language that is used to create documents or Web sites on the Internet.

**Internet -** An electronic communications network that connects computer networks and organizational computer facilities around the world.

**Javascript -** A dynamic computer programming language, used as part of web browsers, whose implementations allow client-side scripts to interact with the user.

**Laboratory -** A room or building with special equipment for doing scientific experiments and tests.

**Lecture -** A talk or speech given to a group of people to teach them about a particular subject.

**MySQL -** A database management system.

**Notification -** The act of notifying someone.

**ODU** - Old Dominion University, a public 4-year university in Norfolk, Virginia.

**PHP -** A server-side scripting language designed for web development.

**Prototype -** An original or first model of something from which other forms are copied or developed.

**Recitation -** A class period especially in association with and for review of a lecture.

**Server -** The main computer in a network which provides files and services that are used by the other computers.

**SQL -** A programming language designed for managing data held in a relational database management system.

**University Identification Number (UIN) -** A unique identification number given out to students at Old Dominion University.

**Wait-list -** To be put on a waiting list.

**1.4 References**

Author Unknown. (2013). “Digest of Education Statistics: 2012.” Retrieved September 18,

2014.

Lab 1 - CORSICA Product Description. Version 2. (2014, October). CORSICA. Red Team. CS411W: Bitaseme Mboe

(This space is intentionally left blank.)

**1.5 Overview**

This product specification provides a description of both the CORSICA product itself and the prototype. Furthermore, the product specification details the hardware and software configurations and key features of the CORSICA prototype. The prototype will have limited capabilities compared to the actual product. This product specification will also provide a list of specific requirements that the prototype must satisfy.

**2      General Description**

The CORSICA prototype is necessary in order to provide proof of concept. The proof of concept will have enough functionality to show mitigation of risks. Also, the customer will be able to provide feedback on the prototype. This will allow the CORSICA team to do experiments with the customer in order to improve the product.

**2.1 Prototype Architecture Description**

The prototype will have limited functionality compared to the real world product. As defined in Table 1, the prototype will have the following features: operation as a mainly automated system, fair enrollment process, notification system, wait-list statistics analysis, and wait-list seat reservations.

Figure 3 shows the prototype’s five major components: course database, back-end algorithms, notification system, test harness, and user interface. Prototype interaction will be done through the user interface. The course database will utilize a test harness that contains course names, course sections, course capacity, number of students enrolled in the course, and number of available seats.



### *Figure 3*. Prototype Major Functional Component Diagram

|  |
| --- |
|   |
|   |  |

**2.2 Prototype Functional Description**

The prototype’s functionality is comprised of back-end algorithms and user access levels. The back-end algorithms in the prototype include Load Enrollment Data File, Open Course, Check for Open Seats, Add Student to Wait-list, Notification, Drop Student from Wait-list, and Close Course. Table 2 contains the algorithm descriptions.

(This space is intentionally left blank.)

|  |  |
| --- | --- |
| **Algorithm** | **How it functions** |
| **Load Enrollment Data Files** | **Course data files are loaded into CORSICA.****Files contain Course Capacity, Number of Enrollments, and Available seats.** |
| **Open Course** | **An Administrator or Scheduler user logs into Banner and opens a course for students to enroll in.****Banner database is updated****CORSICA database is notified of change and is updated** |
| **Check for Open Seats** | **Once a course becomes full, a wait-list is activated for it by CORSICA****CORSICA will continually reference the current course capacity and amount of students enrolled.****If the amount of students enrolled is less than course capacity, a seat has become available.****CORSICA database updates****Calls notification algorithm** |
| **Add Student to Wait-list** | **Student X wishes to enroll in Course Y’s wait-list****CORSICA receives this request and adds Students X to wait-list queue****Course Y’s wait-list is updated** |
| **Notification** | **The check for open seats algorithm completes and returns true for an available seat****All students on the wait-list queue are notified of opening****Students respond** |
| **Drop Student from Wait-list** | **Student X wishes to be dropped from Course Y’s wait-list or the time window for that student has expired****CORSICA receives this request and removes Student X from the wait-list queue****Course Y’s wait-list is updated** |
| **Close Course** | **An Administrator or Scheduler user logs into Banner and closes a course as an available option for students to enroll in****Banner database is updated****CORSICA database is notified of change is updated** |

### *Table 2*. Description of Algorithms

User access levels are determined at the time of log-in. The different types of users are categorized into Administrators, Schedulers, Advisors, Students, and Visitors. Table 3 details all five user access levels. The prototype will only have Administrator, Scheduler, and Student use cases.

(This space is intentionally left blank.)



### *Table 3*. CORSICA Access Control Table

##  2.3 External Interfaces

The prototype requires software and user interfaces. The prototype can function independent of hardware interfaces and communication protocols.

**2.3.1    Hardware Interfaces**

N/A

**2.3.2    Software Interfaces**

For software components, the database will contain all of CORSICA’s information. This database would be coded in MySQL and manipulated using PHP, SQL, JavaScript, and HTML. PHP, SQL, JavaScript, and HTML are also what the GUIs would be coded in. Banner would be the only software that CORSICA needs to interact with for the real world product, but for the prototype, a test harness will be used.

**2.3.3    User Interfaces**

A mouse and keyboard will be required for the user to interact with CORSICA. The prototype has a user interface that includes going to a search field and typing in a search term. CORSICA would then return a filtered list of all items that match the search term.

**2.3.4    Communications Protocols and Interfaces**

N/A

**3      Specific Requirements**

Specific requirements are comprised of two categories: functional requirements and performance requirements. CORSICA will only have functional requirements. The functional requirements in section 3.1 are located in a separate document.

**3.1 Functional Requirements**

These requirements are located in a separate functional requirements document.

**3.2 Performance Requirements**

N/A