**Lab II – CORSICA Prototype Product Specification**

**Team Red**

**Anthony Baron**

**CS411W**

**Janet Brunelle**

**November 3, 2014**

**Version II**

**Table of Contents**

[**1** **Introduction** 3](#_Toc402804380)

[**1.1** **Purpose** 4](#_Toc402804381)

[**1.2 Scope** 4](#_Toc402804382)

[**1.3 Definitions, Acronyms, and Abbreviations** 7](#_Toc402804383)

[**1.4 References** 9](#_Toc402804384)

[**1.5 Overview** 9](#_Toc402804385)

[**2** **General Description** 9](#_Toc402804386)

[**2.1** **Prototype Architecture Description** 10](#_Toc402804387)

[**2.2** **Prototype Functional Description** 14](#_Toc402804388)

[**2.3** **External Interfaces** 14](#_Toc402804389)

**List of Tables**

[Table 1 Real World Product vs. Prototype 10](#_Toc402804361)

[Table 2 Description of Algorithms 14](#_Toc402804362)

**List of Figures**

[Figure 1 – Current Process Flow 5](#_Toc402804363)

[Figure 2 – New Process Flow 6](#_Toc402804364)

[Figure 3 – Major Functional Components Diagram 11](#_Toc402804365)

# **Introduction**

Earning a college degree has become a necessity to gain entry into the professional workforce. Individuals who have forms of higher education earn larger incomes and have a lower unemployment rate compared to those who have no higher education (Bureau of Labor Statistics 2014). It is expected that between 2011 and 2021, college enrollment will increase by 13 percent (U.S. Department of Education 2013). With this rise in attendance, universities are struggling to establish appropriate course capacities.

Old Dominion University (ODU) is currently one of those universities having issues with deciding course capacity limits. Today, students who wish to enroll in a course will log into LEO online, ODU’s course enrollment system, and enroll in a course through that interface. Often, enough spaces in a course are not allocated to meet the demand of student enrollment needs. While the enrollment process is seamless, when course capacities are not met, ODU’s current enrollment system lacks an efficient wait-list process. When a course’s capacity is reached, there is no method in place to facilitate notifications of newly available seats. Due to this inefficiency, problems arise for many individuals at the university which are as follows: the student, faculty, advisors, and schedulers. When a course reaches max capacity, students communicate with faculty and their advisors in attempts to receive an overload so they may enroll. In addition, students will spend time monitoring LEO to find an opportunity to enroll in a course that has reached capacity.

Due to limited resources and faculty, a wait-list system must be implemented in order to streamline a fair enrollment process. The COmputeR ScIenCe wAit-list (CORSICA) system will be prototyped for the Computer Science (CS) department to demonstrate a wait-list benefit to ODU.

## **Purpose**

CORSICA was developed to address ODU’s currently inefficient wait-list process. The prototype will demonstrate the following functional goals and objectives: operate as a mainly automated system, make efficient use of current notification systems (will be simulated), reserve seats for student users on course wait-lists, ensure a fair enrollment process, and trigger alerts. CORSICA will operate as an automated system while still accepting manual user requests / actions. Allowing the system to be mainly automated ensures the greatest ease of use for all users.

CORSICA’s notification system will utilize text and email features. Students are required to opt into the system by providing their contact information. In an effort to ensure fairness during course enrollment periods, CORSICA will reserve seats for student users on a course wait-list and block sign up attempts from students who are not. Finally, alerts will be triggered, by means of a pop up textbox, when an Administrator user or Scheduler changes course information to ensure all changes are intentional.

## **1.2 Scope**

The idea for the project was originally proposed by Professor Brunelle, who is the Assistant Chair of the Computer Science Department, Chief Departmental Advisor, and Senior Lecturer. Too often, courses reach capacity, and student enrollment needs are not met. When this occurs, students begin to engage University personnel with email and calls in hopes of being able to enroll in a full course. Figure 1 details the problematic situation before CORSICA’s implementation.

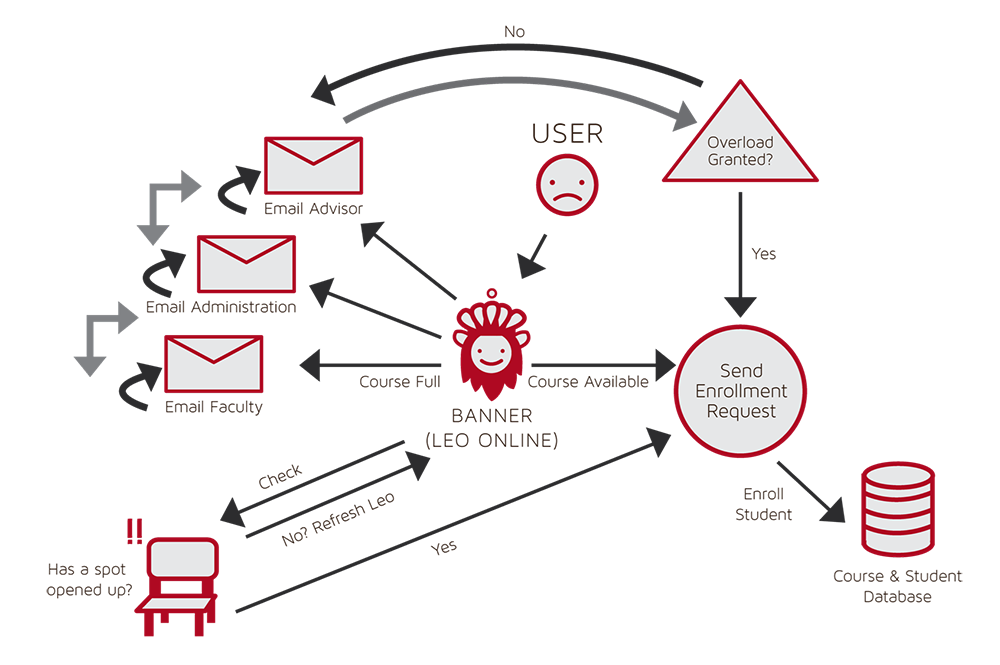


Figure 1 – Current Process Flow

Understanding the great need for an effective wait-list process, the CS 411W Red Team took the idea and turned it into what is now CORSICA. CORSICA is designed to be used primarily by University students, advisors, faculty, and course schedulers. The wait-list system will enable a more streamlined and effective course enrollment process for closed courses. Figure 2 describes how CORSICA will improve the current process.

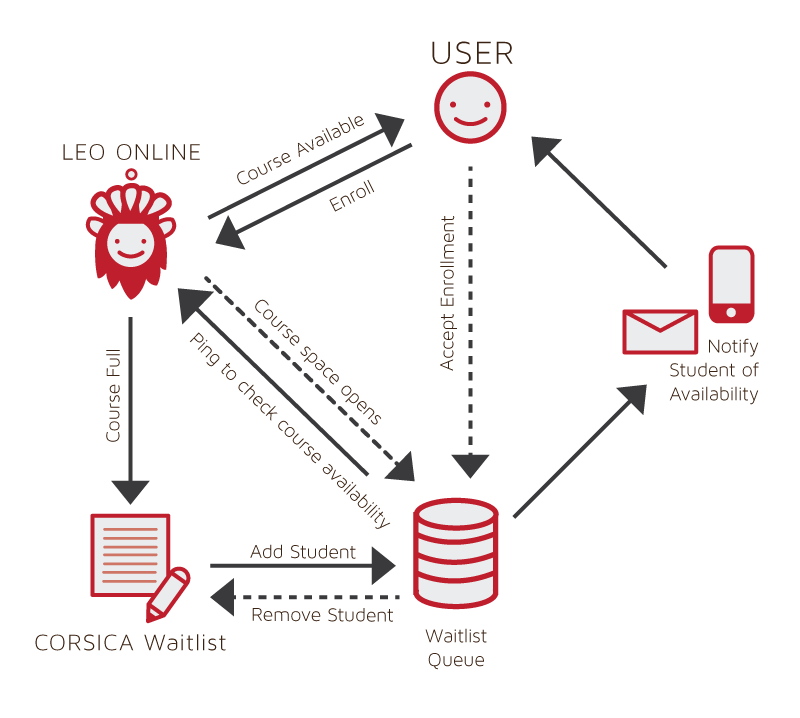


Figure 2 – New Process Flow

## **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.

**\*CORSICA Database (CDB)** – A collection of information required by CORSICA to be functional.

**\*Course Registration Number (CRN)**- A unique number given to each course section for identification purposes.

**\*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.

**\*Text Message -** A short message that is sent electronically to a cell phone or other device.

**\*\*\*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.

**\***Found at http://www.merriam-webster.com/

**\*\*** Found at <http://en.wikipedia.org/wiki/>

**\*\*\*** Found at https://www.odu.edu

## **1.4 References**

1. Anthony Baron, (September 30th, 2014)*. Lab I – CORSICA Product Description*. Old Dominion University

2. (March 24, 2014). Employment Projections.  *Bureau of Labor Statistics.* Retrieved May 1, 2014, from <http://www.bls.gov/emp/ep_chart_001.htm>

1. Yu, R. (2012, September 3). Voice mail in decline with rise of text, loss of patience. *USATODAY.COM*. Retrieved April 27, 2014, from <http://usatoday30.usatoday.com/tech/news/story/2012-09-03/voicemail-decline/57556358/1>
2. (December, 2013). Digest of Education Statistics: 2012. Retrieved September 17, 2014, from http://nces.ed.gov/programs/digest/d12/ch\_3.asp

## **1.5 Overview**

This prototype product specification document first provides a high-level look at the goals and structure of CORSICA’s prototype. The purpose of the project is established, and how that purpose will be met is described. This document also describes the hardware and software architecture involved which enable the prototype to have functional capabilities. Software interfaces and external components are also detailed. Finally, the document lists specific requirements that need to be met by the prototype.

# **General Description**

The product will be prototyped in order to provide proof of concept and mitigate risk. CORSICA is web-based notification software that is intended to alert students on a wait-list when an opportunity to enroll in a full course becomes available. At the user level, CORSICA consists of five GUIs for Administrator, Advisor, Faculty, Student, and Guest users. These GUIs allow each of the user groups to interact with CORSICA.

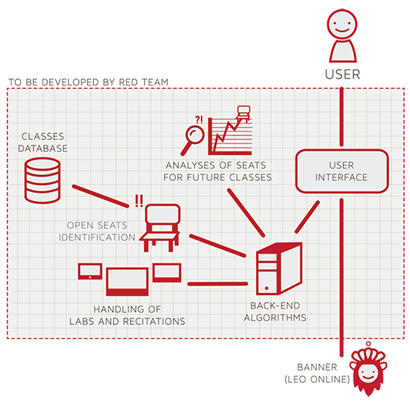
## **Prototype Architecture Description**

Capabilities will be reduced in the CORSICA prototype. It will not have the full functionality of the real world product. 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 |
| Alert System | Yes | No  · Will be simulated with text box |
| 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 vs. Prototype

Figure 3 shows that the CORSICA prototype will be comprised of five major components: a course database, back-end algorithms, a notification system, a user interface, and a test harness. A user will interact with the prototype through the command prompt using basic Input/Output. The course database will be simulated with a text file. The file will contain all the information the prototype will require, which is as follows: Course name, Course sections, Course capacity, Number of students enrolled in the course, and the amount of available seats. The data will be acquired via Dr. Levinstein’s course scheduling website.



Test Harness

Figure 3 – Major Functional Components Diagram

Back-end algorithms will be developed for the prototype. They include the following: Load Enrollment Data File, Add Student to Wait-list, Check for Open Seats, Notification, and Drop Student from Wait-list. Table 2 provides explanations as to how the algorithms will function.

|  |  |
| --- | --- |
| 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 |
| Increase Course Capacity | Administrator logs into Banner and increases course capacity for Course Y    Banner database is updated    CORSICA database is notified of change and 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 |
| Linked CRN | This algorithm will check to see if a course has any linked CRNs. This will inform CORSICA if a lecture section has a lab or recitation to go along with it.  If it does, then if one or the other becomes full, both will be treated as a wait-list course. |

Table 2  
Description of Algorithms

The prototype will simulate the notification process by using a prompt that alerts the user that students have been notified.

## **Prototype Functional Description**

The Banner database of students and courses will be simulated with false data files in the prototype as access will have not yet been granted to the development team. The Seat Analysis System will be omitted in the prototype as it is not essential functionality. The prototype will show how CORSICA operates for the following users: Administrator, Advisor, Scheduler, Student, and Visitor.

The CORSICA prototype will demonstrate the following functional goals and objectives: operate as a mainly automated system, provide an efficient notification systems, ensure a fair enrollment process, and trigger alerts. CORSICA will operate as an automated system while still accepting user requests / actions. Having a mainly automated system ensures ease of use for all users. In an effort to ensure fairness during course enrollment periods, CORSICA will reserve seats for student users on a course wait-list and prohibit sign-up attempts from students who are not. Finally, alerts will be triggered, by means of a pop up textbox, when an Administrator user of Scheduler changes course information to ensure all changes are intentional.

## **External Interfaces**

No extended interfaces will be needed to complete development of the CORSICA prototype. The prototype will have all the resources necessary to demonstrate proof of concept. Therefore, no additional resources will be required for the development and demonstration of the CORSICA prototype.