**Lab 1 - CORSICA Product Description**

**Team Red**

**Bitaseme Mboe**

**CS 411W**

**Professor Janet Brunelle**

**October 8, 2014**

**Version 2.0**

Table of Contents

[**1** **INTRODUCTION** 3](#_Toc401219929)

[**2** **CORSICA PRODUCT DESCRIPTION** 4](#_Toc401219930)

[2.1 Key Product Features and Capabilities 4](#_Toc401219931)

[2.2 Major Components (Hardware/Software) 6](#_Toc401219932)

[**3**  **IDENTIFICATION OF CASE STUDY** 7](#_Toc401219933)

[**4**  **CORSICA PRODUCT PROTOTYPE DESCRIPTION** 7](#_Toc401219934)

[4.1 Prototype Architecture (Hardware/Software) 9](#_Toc401219935)

[4.2 Prototype Features and Capabilities 12](#_Toc401219936)

[4.3 Prototype Development Challenges 12](#_Toc401219937)

[**Glossary** 14](#_Toc401219938)

[**REFERENCES** 17](#_Toc401219939)

*List of Figures*

*Figure 1.* Hardware Components………………………………………………………………. 6

*Figure 2*. Prototype Major Functional Components Diagram…………………………………. 9

*List of Tables*

*Table 1.* CORSICA Access Control Table……………………………………………………... 5

*Table 2*. Real World Product and Prototype Comparison ………………………………………8

*Table 3*. Description of Algorithms ………………………………………………………..10-12

*Table 4.* Prototype Challenges………………………………………………………………...13

(This space intentionally left blank.)

# **1 INTRODUCTION**

Registering for courses each semester in universities can be an anxious time of year. There are a limited amount of seats in courses available 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 also causes confusion, is time consuming, and is 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.

The system that is being implemented to remedy this issue is called COmputeR ScIenCe wAit-list (CORSICA). 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.

# **2 CORSICA PRODUCT DESCRIPTION**

CORSICA is an effective wait-list system that provides a method for students to wait for an available seat in a course. A notification will be sent to students on a wait-list when there is an open seat in a course. These notifications will be in the form of e-mail.

## 2.1 Key Product Features and Capabilities

A key feature of CORSICA is to notify wait-listed students of available seats in courses. When a student receives a notification, they will have 24 hours to register for the specific course. If this same student should not register within 24 hours, this student is removed from the wait-list and the next student will have the opportunity to get into the course. The notification system will also alert students when a wait-list has been closed by the Scheduler.

Other key features of CORSICA include its Graphical User Interface (GUI), browser compatibility, the ability for students to view their wait-list position, and the privilege to add or remove themselves from a wait-list.

CORSICA will also handle courses that have lectures along with labs and recitations. Courses such as these require a student to register for three separate wait-lists (one wait-list for each component of the course). The reason for this requirement is because course components like these are linked in Banner.

Another capability of CORSICA is the use of user access levels, which is determined at the time of log-in. The different types of users are categorized into Administrators, Schedulers, Advisors, Students, and Visitors. Table 1 details all five user access levels.

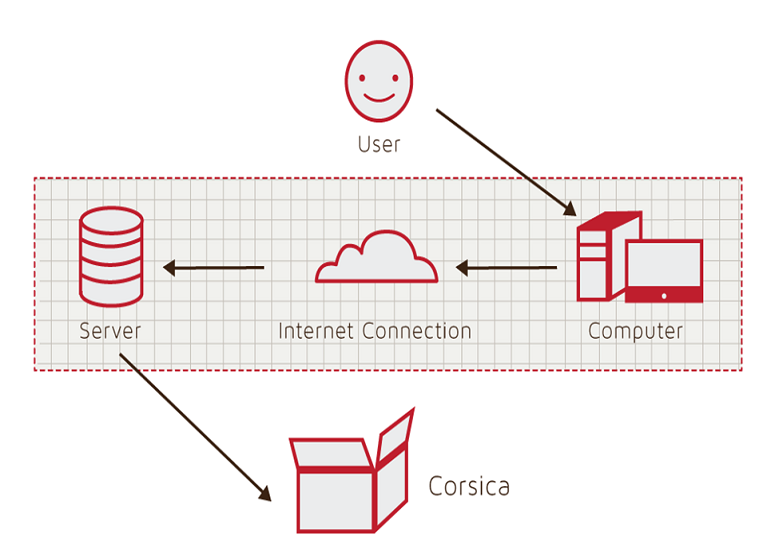


*Table 1*. CORSICA Access Control Table

## 2.2 Major Components (Hardware/Software)

Hardware and software components are required for CORSICA’s implementation. Figure 1 shows three main components: a computer, an Internet connection, and a server. The computer and Internet connection are for connecting to and using the Internet. The server consists of a web and database server. The web server will display the user interface and deliver web pages and the database server will hold the database.

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. Certain algorithms for CORSICA would include: Load Enrollment File Data, Close Course, Notification, Add and Drop Student to Wait-list, and Check for Open Seats.



*Figure 1.* Hardware Components

# **3 IDENTIFICATION OF CASE STUDY**

CORSICA will be developed for ODU’s CS Department with help from the CORSICA Team’s faculty mentor, Dr. Irwin Levinstein. Dr. Levinstein is an Associate Professor at ODU and is the Scheduler for the CS department. As a Scheduler, Dr. Levinstein can determine if course capacity should be increased or decreased. CORSICA will optimize course registration. This will eliminate the emails that advisors and faculty receive from students requesting to register for courses that are full. For the students, this means not having to check LeoOnline periodically to see if there has been any change in enrollment for a particular course.

# **4 CORSICA PRODUCT PROTOTYPE DESCRIPTION**

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 mitigate risks. The prototype will have operations for Administrators, Schedulers, Advisors, Students, and Visitors.

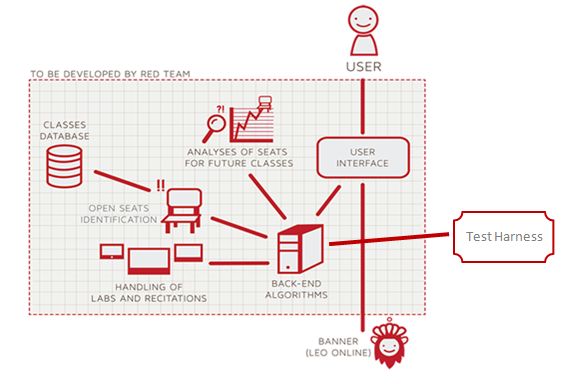
The prototype will have the following features: operation as a mainly automated system, current notification system efficiency, fair enrollment process, notifications, and wait-list seat reservations. CORSICA’s notification system will be automated and send emails to wait-listed students about their wait-list position status. Seats will be reserved for Students on a wait-list to demonstrate fairness. Table 2 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 2*. Real World Product and Prototype Comparison

## 4.1 Prototype Architecture (Hardware/Software)

Figure 2 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 2*. Prototype Major Functional Components Diagram

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 3 contains the algorithm descriptions.

|  |  |
| --- | --- |
| **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 3*. Description of Algorithms

## 4.2 Prototype Features and Capabilities

The CORSICA prototype will be an example of the product’s functionality. The user roles are Administrator, Advisor, Scheduler, Student, and Visitor. The prototype will only demonstrate Administrator, Scheduler, and Student use cases.

## 4.3 Prototype Development Challenges

The prototype objectives referenced in Table 2 do not come without challenges. Table 4 shows prototype objectives and the respective challenges.

(This space is intentionally left blank.)

|  |  |  |
| --- | --- | --- |
| **Objectives** | **Prototype** | **Challenges** |
| **Environments for all Users:** | **No**  · **Will demonstrate student, admin, and scheduler users** | **Working out all the bugs in CORSICA to allow all users to use CORSICA as intended** |
| **Notification System** | **No**  · **Will be simulated with text box** | **Allowing CORSICA to sync EXACTLY with the University’s Clock** |
| **Mostly automated** | **No**  · **Will rely heavily on user interaction** | **Users need to be knowledgeable of CORSICA** |
| **Link to Banner** | **No**  · **Will be loaded with data.txt files instead** | **Using Black Box Testing to certify the text file compatibility** |
| **Link to LEO Online** | **No**  · **Will be simulated with command box menu** | **Maintaining LEO Online’s layout while appending the CORSICA option on the course registration screen** |
| **GUI** | **Very Basic (Text System)** | **Coding a GUI that looks professional and is simple to navigate** |

*Table 4*. Prototype Challenges

# **Glossary**

**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.

(This space is intentionally left blank.)

# **REFERENCES**

Author Unknown. (2013). “Digest of Education Statistics: 2012.” Retrieved September 18, 2014.