CORSICA Product Description

Anthony Baron
Janet Brunelle

Version 1.0

Old Dominion University

May 8, 2014

CS 410 Lab 1

Table of Contents

List of Tables

List of Figures

CS 411W LAB I – PRODUCT DOCUMENT

1. 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 and sign up through that interface. Often, enough spaces in a course are not allocated to meet the high demand of student enrollment. 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 additional sign ups. Due to this inefficiency, problems arise for many individuals at the university: the student, faculty, advisors, and schedulers. When a course reaches max capacity, students communicate with faculty and their advisors in attempts to enroll in that course section. In addition, students will spend time monitoring LEO and wait for the opportunity to enroll in a course that has reached capacity.

A wait-list system must be implemented in order to streamline the enrollment process. The COmputeR ScIenCe wAit-list (CORSICA) system will be prototyped for the Computer Science (CS) department to demonstrate its benefit to ODU. Once a course, lab, or recitation

section becomes full, the wait-list system will become available to students as a service available over the Internet via LEO.

2 CORSICA PRODUCT DESCRIPTION

CORSICA is the solution to ODU's problem of not having a simple, fair, and efficient wait-list process. The program will cater to a set of users: Advisors, Schedulers, and Students. However, it will be managed by Administrative users and allow for Visitor users to examine the Graphical User Interface (GUI) without sign in credentials. Users will access the program's features through LEO which is the current enrollment system provided by ODU via the Internet. CORSICA will enable students to secure their position if/when a course seat becomes available. The system will provide notifications to students informing them of this change as well as act as a monitoring system should the student wish to view the wait-list status.

Once a course reaches capacity, it will no longer be available to enroll via LEO. CORSICA will handle the enrollment process until the current semesters add / drop period has ended. When CORSICA is engaged, a student will locate the course they wish to be enrolled in and opt to be added to the course's wait-list. Upon initial sign up, the student will have the opportunity to provide their cell phone number and opt in for text alerts in addition to email notifications. After being placed on the wait-list the amount of attention the student needs to place on monitoring LEO for an enrollment opportunity in a full course is lessened. With a built in notification system, CORSICA will send students alerts via email, and if desired, via a text message.

2.1 Key Product Features and Capabilities

CORSICA was designed to provide ODU with a fair, simple, and efficient wait-list process. Once a course reaches capacity, three procedures will be followed. One, the system blocks LEO online from allowing students to enroll for a course that is now maintained by the wait-list. This stops students who are not on the wait-list from enrolling for a course. Two, once students begin to enroll on the wait-list the system will maintain the order in which sign ups have occurred. This helps keep order and ensure a first come first serve method of operation. Three, the system will ensure that a student may sign up for no more than six wait-lists (including different sections of the same course). This helps inhibit student abuse of the system. These procedures ensure a fair process.

CORSICA is a simple to use wait-list process for students and its other users. Once a student user wishes to sign up for a course wait-list all that the user needs to do is click the "Enroll in Wait-list" button. For advisors, the system will allow simple navigation through their available commands. These users will be able to quickly via: view waitlists a student is on, view a list of courses that do / do not have waitlists, search for a course, search for a student, and move / delete a student on a wait-list via "one click buttons". For schedulers, the system will allow simple navigation through their available commands. These users will be able to quickly: expand / decrease capacity of a course and create / close a course to be a wait-list option via "one click buttons". These capabilities ensure that CORSICA is a simple and user friendly application.

CORSCIA is an efficient application in that it: operates as a mostly automated system, notifies students of an opening for a course in a timely manner, and can handle courses with labs / recitations. The ability of having CORSICA as a mostly automated system assists in making it

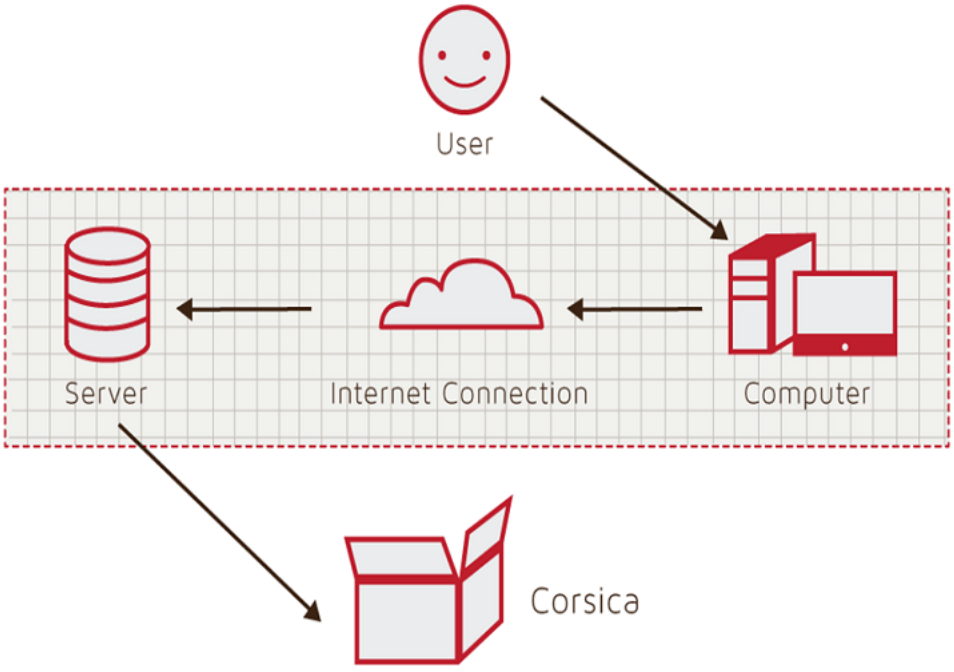
a simple and efficient wait-list process. All student users need to do is enroll on a wait-list and enroll in a course when that opportunity becomes available.

The notification system will promptly alert students when they have the opportunity to enroll in a course. Students will have a 24-hour window of time to enroll in a course after receiving their alert. If that student fails to enroll within the time frame, they are dropped from the wait-list and the next student in line has that opportunity.

Courses that have lab or recitation sections are managed by the wait-list system as well. If a lecture section becomes full, all relative labs and/or recitations are treated as having reached max capacity. The same is true if either the lab or recitation became full first. This helps to ensure that students are able to enroll in all needed sections.

2.2 Major Components (Hardware/Software)

Figure 1 shows CORSICA will be comprised of three major hardware components: a computer, an Internet connection, and be connected to a web / database server.



The computer and the Internet connection work together in allowing the user to interact with the program. The web server allows the computer to display the web pages which displays the program's GUI. The database server allows the CORSICA database (CDB) to be connected to the rest of the program.

The wait-list system will be comprised of various software components which include: the CDB, GUIs, and back-end algorithms. The CDB will be comprised of: user, student, course, wait-list, and course statistic data tables. All tables will hold important data the enable CORSICA to be functional.

User tables hold all information pertaining to the user categories: administrator, advisor, faculty, student, and visitor. Each category holds different privileges and a log in must be required in order to distinguish users. Student tables will contain all relevant information about a student for the wait-list. The program will need to know the student's name, University Identification Number (UIN), email address, and if provided, their cell phone number. Course tables contain all relevant course information: course prefix, name, course registration number (CRN), meeting times, capacity, amount of enrolled students, and if it has a lab / recitation section. The wait-list tables will only store data about a course's wait-list and the students on them. Course statistic data tables keep track of how many courses had wait-lists, the number of students on them, and which courses did not. This data is valuable to schedulers as it will assist in decisions regarding course structure (size, number of sections, etc.).

Figure 1 - Hardware Components

CORSICA will have multiple GUIs. They will be created using various scripting languages: HTML, CSS, and JavaScript. Each user group will have their own unique version.

The purpose of this is to separate user capabilities. Every user has different commands they are able to execute. For example, administrators will be able to have master control over the system where as student users will simply be able to add / drop themselves from a wait-list. For detailed visuals of the differ user GUIs, see http://www.cs.odu.edu/~411red/deliverables.html, under the user interface section.

Back-end algorithms will be a major software component in CORSICA. The algorithms will include: load enrollment data files, add student to wait-list, drop student from wait-list, check for open seats, notification, increase course capacity, open course, close course, and linked CRNs. They will be coded with C++ and allow the system to be a functional product.

3 IDENTIFICATION OF CASE STUDY

A development team is implementing CORSICA for ODU's Computer Science Department. The team is under Dr. Irwin Levinstein who is an Associate Professor at ODU in Computer Science. He is also the course scheduler for the CS department. The wait-list program will be trialed by the CS department and statistical data of its operation will be gathered. This data will assist course schedulers in preparing upcoming semester course rosters. Pending a successful prototype, CORSICA could be expanded to ODU's other departments and other universities.

4 CORSICA PRODUCT PROTOTYPE DESCRIPTION

Capabilities will be reduced in the CORSICA prototype in order to provide proof of concept. 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. The Seat Analysis System will be omitted in the prototype as it does not contribute essential functionality. **Simulation** will show

how CORSICA operates for various users: Administrator, Advisor, Scheduler, Student, and Visitor.

The CORSICA prototype will demonstrate the following functional goals and objectives: operate as a mainly automated system, make efficient use of current notification system (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 user requests / actions. Allowing the system to be mainly automated ensures the greatest ease of use for all users. ODU's current alert system will be tied into CORSICA; specifically the text and email notification features. The current alert system sends text messages and emails to students in the event of school closures. 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 of Scheduler changes course information to ensure all changes are intentional. 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 and Prototype Comparison

4.1 Prototype Architecture (Hardware/Software)

Figure 2 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 I/O. The course database will be simulated with a text file. The file will contain all the information the prototype will require: Course name, Course sections, Course capacity, Number of students enrolled in the course, and the amount of available seats. The data will be arbitrarily generated as CORSICA will not be granted access to real data during the prototype demonstration.

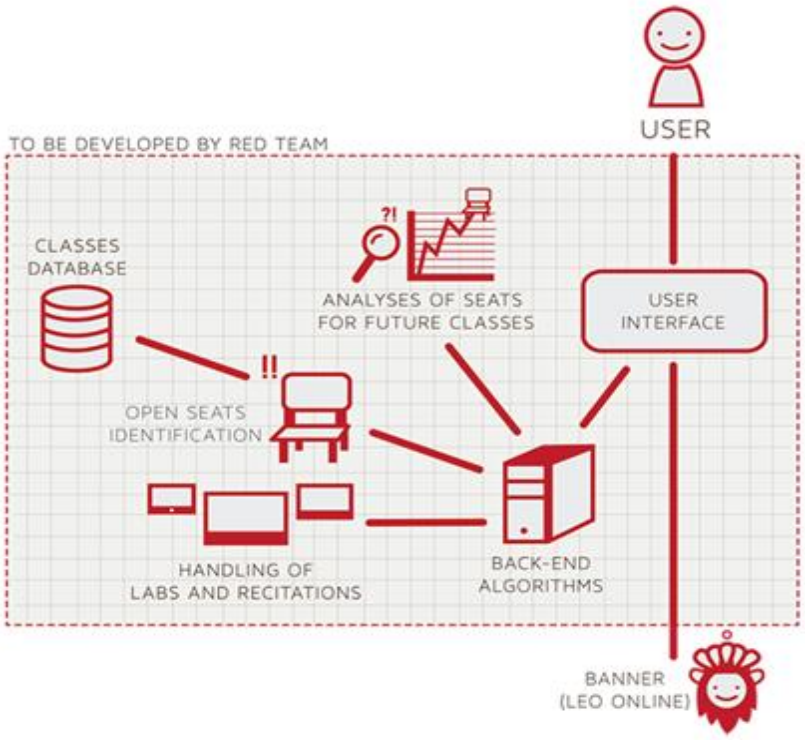


Figure 2 - Prototype Major Functional Components Diagram

Back-end algorithms will be developed for the prototype. They include the following: Load Enrollment Data File, Open Course, Add Student to Wait-list, Check for Open Seats, Notification, Increase Course Capacity, Drop Student from Wait-list, and Close Course. 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**
	Student X wishes to be dropped from Course Y's wait-list or the time

Drop Student from Wait-list	**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

CORSICA will integrate with ODU's current notification system. More information on the current system may be found at https://www.odu.edu/content/odu/life/health-

safety/safety/alerts.html. The prototype will simulate the notification process with a prompt alerting the user that students have been notified.

4.2 Prototype Features and Capabilities

The purpose of the CORSICA prototype is to demonstrate functionality and proof of concept. The prototype will show how three of the five user roles may interact with the system. It will be functional for the Administrator, the Scheduler, and the Student users. A visitor user won't demonstrate much functionality of the system and an Advisor user has many of the same abilities as a Scheduler so they are being left out. Each of the three included user roles will be described.

When an administrator user logs into CORSICA their home page will be a list of courses and their wait-lists. They will see: a course name, CRN, instructor, and wait-list status and an "Add Course" button on the bottom of their screen. The user may search for a student and view what course wait-lists they are enrolled in. A course may also be searched. Detailed information about that course will be displayed. The admin will see: the course name, CRN, instructor, wait-list status, and what students are currently enrolled on the wait-list.

When a scheduler user logs into CORSICA their home page will be a list of courses and their wait-lists. They will see: a course name, CRN, instructor, and wait-list status and an "Add Course" button on the bottom of their screen. The scheduler may, like the admin, search for a student and view what wait-lists they are enrolled in. A course may be searched which details what students are enrolled on its wait-list.

When a student user logs into CORSICA their home page will be a list of wait-lists they are enrolled in. The student may search for a particular instructor and view what courses they are teaching are currently wait-list options. From that list, the student may choose a section and enroll in the wait-list. Upon doing so, the student will be alerted of what wait-list they are enrolling on as well as their position on it.

4.3 Prototype Development Challenges

In software development, various developmental challenges occur. A prototype will often not obtain the full functionality of a real-world product. The goal of a prototype is to demonstrate proof of concept. This is why the CORSICA prototype focuses on "needs" of the program rather than the "wants". It was developed to be a functional program but has had some features sacrificed. Table 3 details what challenges the prototype has faced and how they are handled.

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**
Alert System	**No** **· Will be simulated with**	**Making the Alert System actually recognize each change to help**

	text box	**ensure intentional changes**
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 3 - 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.

***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

References

1. (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