Lab 1 - CORSICA Product Description

Team Red

Bitaseme Mboe

CS 411W

Professor Janet Brunelle

September 16, 2014

Version 1.0

Table of Contents

List of Figures

List of Tables

CS 411W LAB I – PRODUCT DESCRIPTION DOCUMENT

1 INTRODUCTION

Registering for courses each semester in universities can be an anxious time of year. There are usually a limited amount of spots in courses available and the problem is that there are usually more students available than there are course spots. 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 for handling course overflows.

Why is ODU's class enrollment system insufficient? Well besides ODU having more students enrolled than there are course spots, the current system also causes confusion, is time consuming, and is unfair. Who is affected? Faculty, advisors, schedulers, administrators, and of course, students. When a course has reached capacity, students will contact advisors for an override. Advisors have to get in touch with the respective faculty to see if they will allow an additional student. Sometimes a student gets lucky and can get an override into the course before the course begins. Other times students need to wait for an undetermined amount of time to get into the course. This latter process takes time out of the advisor's and faculty's schedule and the student will have missed course material that the student now needs to make-up.

The current work-in-progress solution to 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. CORSICA effectively will increase student enrollment rates in courses with wait-lists.

2 CORSICA PRODUCT DESCRIPTION

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

2.1 Key Product Features and Capabilities

A key feature of CORSICA is to notify wait-listed students of available spots 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, then 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's (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.

Yet, another capability of CORSICA is the use of user access levels, determined at the time of log-in. The different types of users are categorized into Administrators, Schedulers, Advisors, Students, and Visitors. The Administrators will be responsible for and oversee CORSICA. Schedulers can open or close a course's wait-list and view wait-list statistics to help determine if a specific's course capacity should increase or decrease. Advisors can view student wait-list positions, add or remove students from a wait-list, and change a student's position on a wait-list. Students can view wait-lists that they have registered for, view other wait-lists, and see the number of students on a wait-list. Visitors don't have log-in credentials and therefore will only be able to view certain pages including the CORSICA home page, 'Overview', 'Deliverables', 'Presentations', 'Risks', 'Bibliography', and 'About Us'.

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 and deliver web pages and the database server will manage 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. These latter four languages are what the GUIs would be coded in. Certain algorithms for

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

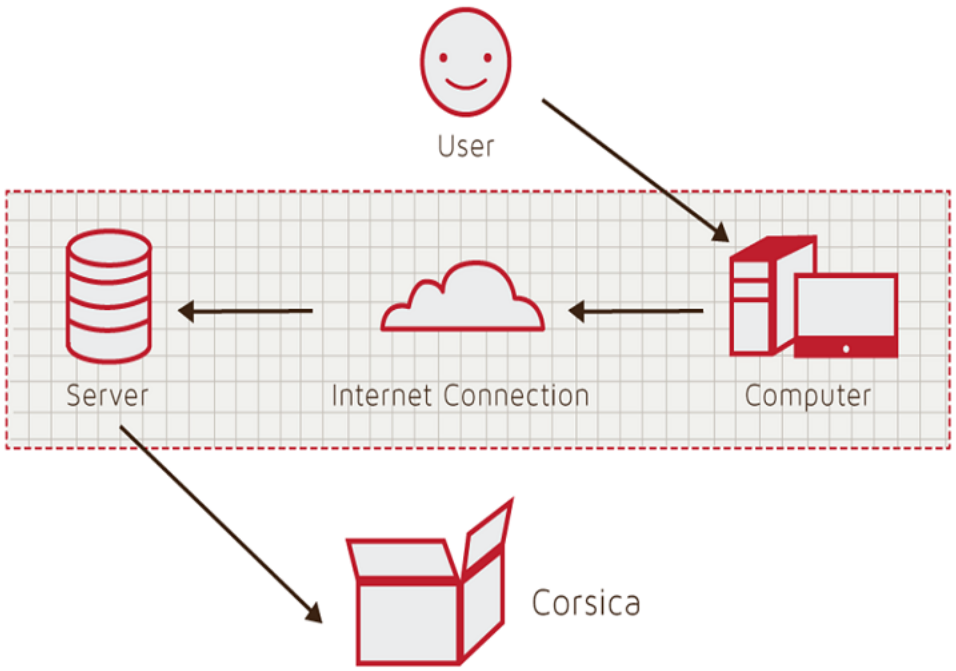

Figure 1. Hardware Components

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 aims to 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.

If successful, CORSICA can be used in other departments at ODU and eventually at other universities.

4 CORSICA PRODUCT PROTOTYPE DESCRIPTION

The CORSICA prototype will have limited capabilities for the purpose of providing core functionality. Enrollment test data files of students and courses will be used. The prototype will display operations for Administrators, Schedulers, Advisors, Students, and Visitors.

The prototype will demonstrate the following: operation as a mainly automated system, current notification system efficiency, fair enrollment process, trigger alerts, and wait-list spot reservations. CORSICA's notification system will be tied into ODU's current alert system, which sends text messages and emails to students about pertinent ODU information. Spots will be reserved for Students on a wait-list to demonstrate fairness. Alerts will be displayed in the form of pop-up textboxes when an Administrator or Scheduler changes course information to confirm intentions. 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 the prototype’s four major components: course database, back-end algorithms, notification system, and user interface. Prototype interaction will be done through a command prompt. The course database will utilize a pre-populated text file that contains course names, course sections, course capacity, number of students enrolled in the course, and number of available spots.

The back-end algorithms in the prototype include Load Enrollment Data File, Open Course, Check for Open Spots, Add Student to Wait-list, Notification, Drop Student from Wait-list, Increase Course Capacity, and Close Course. Table 2 contains the algorithm descriptions.

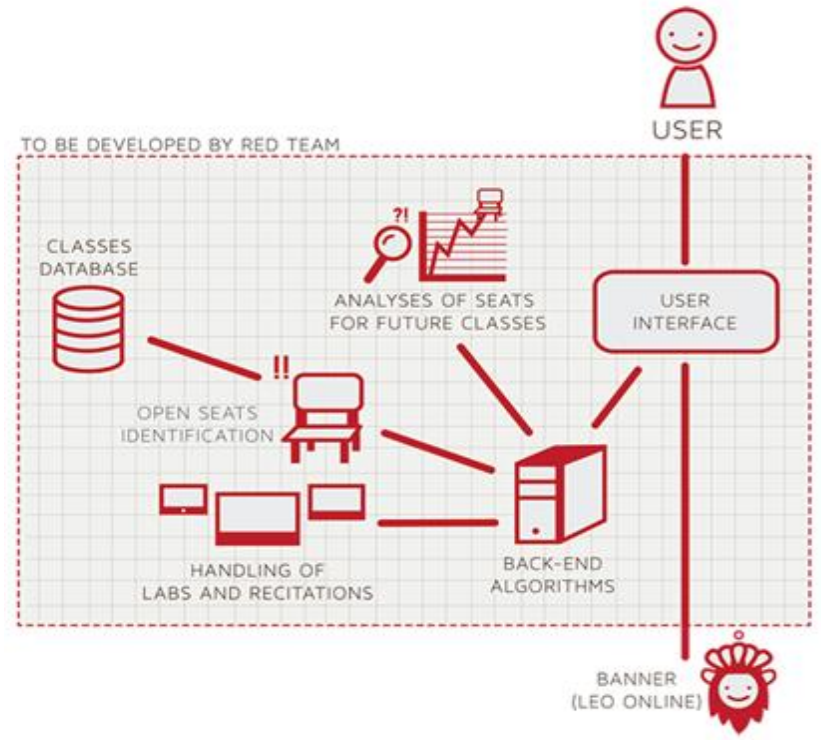
USER
TO BE DEVELOPED BY RED TEAM
CLASSES DATABASE
?!
ANALYSES OF SEATS FOR FUTURE CLASSES
USER INTERFACE
!!
OPEN SEATS IDENTIFICATION
HANDLING OF LABS AND RECITATIONS
BACK-END ALGORITHMS
BANNER (LEO ONLINE)

Figure 2. Prototype Major Functional Components Diagram

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 Spots	**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 spot 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 spots 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**

Table 2. Description of Algorithms

4.2 Prototype Features and Capabilities

The CORSICA prototype will aim to show consumers an example of the product's functionality. The audience of user roles are Administrator, Advisor, Scheduler, Student, and Visitor. The prototype will only demonstrate Administrator, Scheduler, and Student use cases.

An Administrator use case would start with logging into CORSICA. The Administrator will be greeted with a list of courses for the current semester and the corresponding wait-lists. A search for a student by name is necessary in order to see the student's wait-list history. The Administrator can also search for a specific course and see its description and wait-list history. Wait-list contents can be manipulated by changing Student positions as well as adding/removing Students from a wait-list or even a wait-list itself.

Once logged into CORSICA, a Scheduler will see a list of courses for the current semester. Wait-list details can be viewed such as what and how many Students are on the wait-list, and what course the wait-list is linked to. Wait-lists from previous semesters can also be viewed by searching for a list of wait-lists from previous semesters in the search bar. Using the same method, wait-list statistics and analytics can also be viewed. To open and/or close a wait-list for a course, the Scheduler simply needs to search for the particular course and then perform the necessary function(s).

When a Student logs into CORSICA, they will see the wait-lists that they are currently active on. The Student can search the last name of an instructor to return results of all the wait-lists that belong to the course(s) that the instructor is currently teaching. A full course can be chosen and then register a Student on its wait-list. An alert box will be displayed containing the course, semester, and spot on the wait-list that the Student is attempting to register for. After confirmation, the Student is added to the wait-list.

CORSICA is not a product that is guaranteed to be in high demand meaning that there are risks that are associated with its development. Table 3 names all the associated risks and Figure 3 shows their probability and impact.

Customer Risks	**Technical Risks**
C1: Department Use Rejection	T1: Ability to Integrate with Banner
C2: Transition to New GUI	T2: Software Upgrades
C3: Cost of Product	T3: Availability of Server Storage
C4: Product Interest	T4: Security Vulnerability

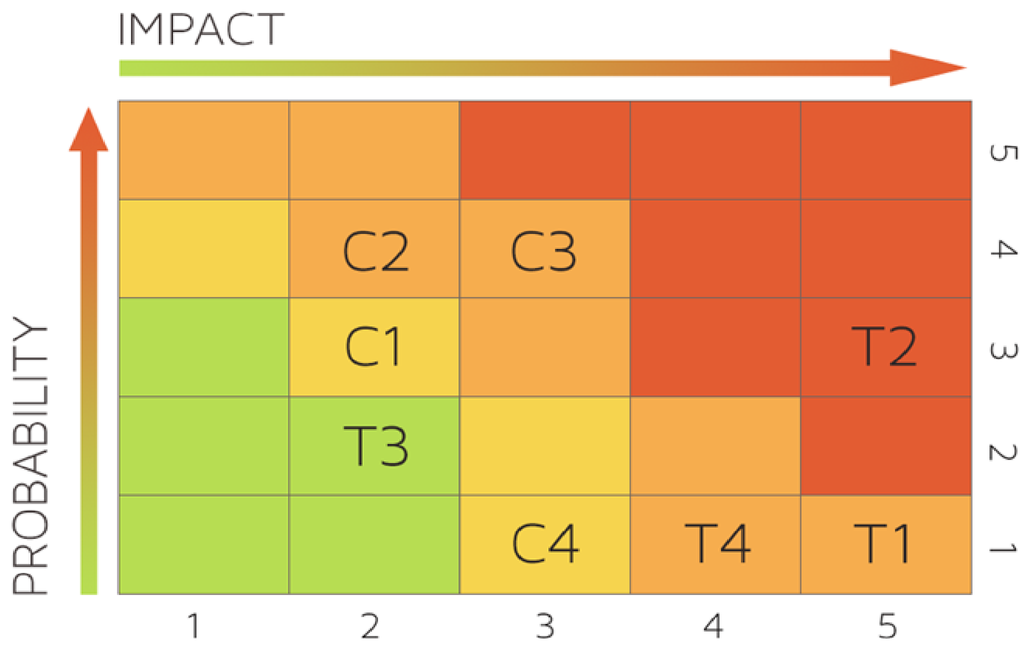

Table 3. Risk Table

Figure 3. Risk Matrix

Prototype Risk Mitigation Addressing: It is possible that the customer and technical risks are not risks at all. It is also possible that those risks have a higher impact and/or probability than estimated. In either case, the objective is to mitigate the risks. Customer risks can be mitigated by displaying CORSICA's ease-of-use and product effectiveness. Transitioning to a new GUI can, overtime, be trivial due to CORSICA updates. Technical risks are dynamic with respect to the particular university. For T1, CORSICA is dependent on Banner so this risk should not be a major issue (if an issue at all). An update feature in the prototype was absent however it was not necessary to show the functionality. T3 and T4 are variable risks, dependent on the university using CORSICA.

4.3 Prototype Development Challenges

The prototype objectives in Table 1 do not come without challenges. Table 4 shows prototype objectives and the respective challenges.

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 text box**	**Making the Alert System actually recognize each change to help 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 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.

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.

REFERENCES

National Center for Education Statistics, U.S. Department of Education. (2013). *Digest of Education Statistics: 2012*. Retrieved September 18, 2014.