Running head: LAB 1 -- CORSICA DESCRIPTION

Lab 1 -- CORSICA Product Description

Nicholas LoGioco

CS411W

Janet Brunelle

August 29th, 2014

Table of Contents

List of Figures

List of Tables

Lab 1 -- CORSICA Product Description

1 INTRODUCTION

According to the National Center for Education Statistics, in the Fall of 2014, 21.6 million students were expected to attend American colleges and universities. This is an increase of about 6.2 million since the Fall of 2000 (National Center for Education Statistics, n.d.). With the rise of students going to colleges and universities, these post-secondary schools will need to increase the number of courses available to students. Another option would be to increase the capacity of each course. However, these options are not always available for post-secondary schools to implement, which can create a problem for students. For a student to graduate on time, that student would need to have a passing grade in each of the courses needed for their major. If a course reaches full capacity, that student would not be able to enroll into that course. Students are then forced to sign up for courses they do not need, only to maintain status as a full-time student. Courses reaching capacity can also lead to students emailing advisors asking to be put into that course.

Dr. Levinstein, an associate professor at Old Dominion University (ODU), has attempted to implement a wait-list system for students. Wait-lists at post-secondary schools are becoming increasingly popular. ODU has a group of senior Computer Science students working on a wait-list solution called CORSICA, or COmputeR ScIenCe wAit-list. CORSICA will allow students to enroll onto a wait-list once a course reaches capacity. Once that courses capacity is no longer full, CORSICA will send a notification via email and/or text message to students on the wait-list. The notification will include the following details: the student spot on the wait-list, and a message saying the student can register for the course given that the student has the appropriate spot on the wait-list.

2 CORSICA PRODUCT DESCRIPTION

2.1 Key Product Features and Capabilities

CORSICA is a wait-list management system that will provide students a simple, fair, and effective way to register for an available seat in a course. It will be available to Administrators, Advisors, Schedulers, Students, and Visitors as a service provided over the Internet. When the capacity for a course is reached, CORSICA will become active for that course. Once this happens, students can place their name on the wait-list and wait to be notified once the capacity is no longer full. This will reduce the negative attitude students would otherwise face when waiting for an available seat. Once a seat becomes available, by another student dropping the course or having a course's capacity raised, the system will send a notification to the first student on the wait-list. To receive notifications, students must choose whether they want to receive notifications via email and/or text message.

CORSICA's main goal is to efficiently notify students when a seat for a course has become available. As students approach graduation, the importance to get into certain courses is very high. It is usually these exact courses that reach maximum capacity. Many students may have to delay graduation simply because they were unable to get into a course or promptly get on the wait-list. CORSICA has a couple of objectives: to eliminate the unnecessary communication between faculty and students once a course reaches capacity, and to do away with the need for students to continuously check to see if the course is still at capacity.

One of the key features of CORSICA is the efficient manner of the notification that students receive when a seat for a course becomes available. After receiving this notification, students will have a 24 hour window to register. However, if they do not register within 24 hours then they will be removed from the wait-list, allowing the next student the opportunity to register. The automated removal of students from the wait-list allows for the next student to be notified. If the Scheduler decides to close a wait-list, CORSICA will notify all the students on the wait-list that the wait-list has been closed. Another feature of CORSICA is that the Graphical User Interface (GUI) is handled over the Internet. This allows users that have different web browsers the ability to use CORSICA. Other features include: the ability for students to sign up using no more than six CRNs for a wait-list, students can go onto the CORSICA website and view their position on a wait-list, and students can add or remove themselves from a wait-list. All of these features aid students in knowing when a seat for a course becomes available.

A major capability of CORSICA is to be able to handle courses that have lectures, labs and recitations. CORSICA will allow students to register for notifications when a seat for any lecture, lab and recitation opens; or students can register for a specific Course Registration Number (CRN) for each lecture, lab and recitation. The student will be given an error if he or she tries to register for one part of a course's wait-list that offers three parts, which includes: lecture, lab and recitation. Therefore, CORSICA will make the student register for three wait-lists or none of them. Another capability that CORSICA offers is allowing different levels of access depending on the credentials entered at log-in. The different types of users offered are Administrators, Advisors, Schedulers, Students, and Visitors. Administrators have full control over the system. Schedulers are important to CORSICA because they view course wait-list statistics for analytical purposes. Schedulers can view these statistics to assist them with knowing whether they should increase or decrease course capacity. They can also manually open or close a wait-list for a course. Advisors have many different tasks including: the ability to view a student's position on a wait-list, the ability to add or remove a student on a wait-list, as well as the ability to move a student up or down on a wait-list. Students can view courses that have wait-lists, the number of students on the wait-list, as well as the wait-lists they are registered for. Visitors have limited functionality in CORSICA, they have the ability to log-in as visitors and view static pages, including: home page, about CORSICA, contact CORSICA, and CORSICA FAQS.

2.2 Major Components (Hardware/Software)

In order for CORSICA to be fully implemented, there are hardware and software components required. These components include: a computer, an Internet connection, and a server. *Figure 1* shows the relationship between these components.

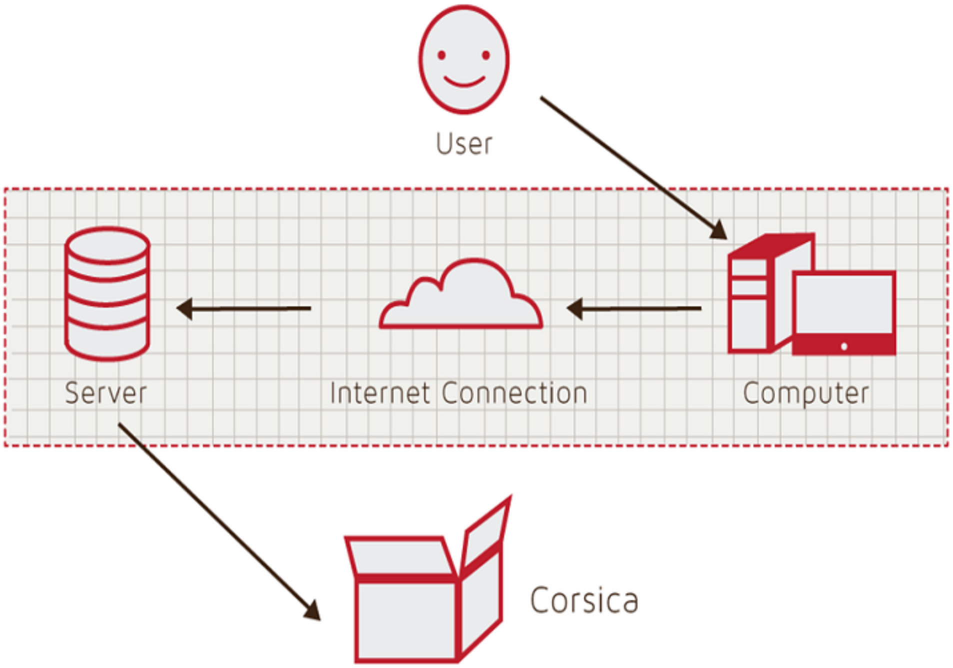

Figure 1

Hardware Components

For someone to use CORSICA, they will need a computer with an Internet connection to connect to the web application. The server will host the application, and provide database services to the application. The software components will include a data created with MySQL. The database will be manipulated using PHP. To navigate through the web application, there will be a Graphical User Interface (GUI) written in HTML, CSS, and javascript. For CORSICA to fully work, it will require algorithms. The algorithms will consist of the following: Load Enrollment Data File, Open Course, Check for Open Seats, Add Student to Wait-list, Notification, Drop Student from Wait-list, Increase Course Capacity, and Close Course. These algorithms will be written in C++. Lastly, a notification system will need to be implemented for students to get notified.

3 IDENTIFICATION OF CASE STUDY

CORSICA will initially be developed for the Computer Science department at Old Dominion University. The future of this wait-list management will rely on Janet Brunelle and Dr. Levinstein. Mrs. Brunelle is a Senior Lecturer at ODU in Computer Science who presented the wait-list project to the CORSICA team. Dr. Levinstein, an Associate Professor at ODU in Computer Science, will be assisting the team with insight and enrollment data he has acquired. Being a scheduler for the department, he can use the analysis component of CORSICA to predict future course capacity sizes.

4 CORSICA PRODUCT PROTOTYPE DESCRIPTION

CORSICA will develop a prototype with reduced capabilities, this is due to not having access to the Banner database, and the small amount of time allotted to develop the wait-list management system. Since the Banner database will not be accessible, the data will be created by the team. The prototype will use this data to simulate how CORSICA operates. The prototype will demonstrate many key functionalities including: registration begins, a course becomes full, a student puts themselves on a wait-list for a course, a spot opens up for the course triggering a notification, student registers for the course, student does not register for the course, and registration ends. CORSICA will be an automated system, ensuring accuracy and fairness. The notification system will be implemented enough to show that once a seat opens up, a notification will be sent via email and/or text message. *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 following major functional components: database, notification, user interface, back-end algorithms, and analysis. The demonstration of the prototype will begin with a user going to the CORSICA GUI. The user will navigate through the GUI, and the back-end algorithms will be called, followed by the database receiving requests or posting requests.

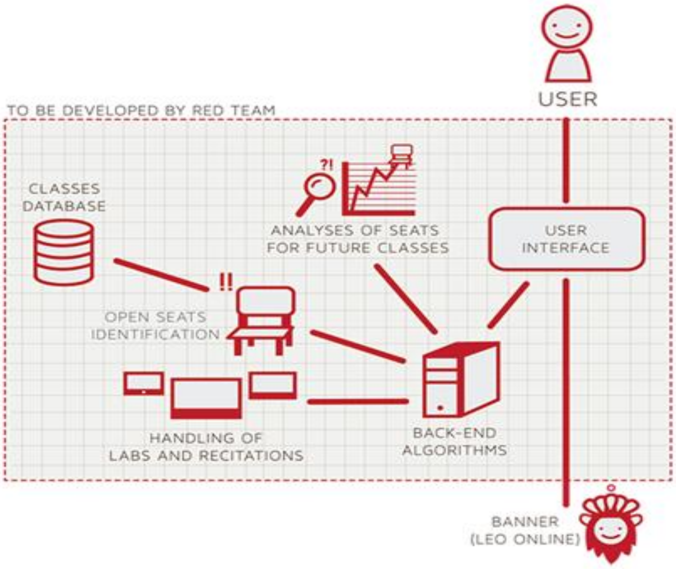
USER
TO BE DEVELOPED BY RED TEAM
CLASSES DATABASE
?!
ANALYSES OF SEATS FOR FUTURE CLASSES
USER INTERFACE
!!
OPEN SEATS IDENTIFICATION
HANDLING OF LABS AND RECITATIONS
BACK-END ALGORITHMS
BANNER (LEO ONLINE)

Figure 2

Prototype Major Functional Components Diagram

CORSICA will use eight algorithms: Load Enrollment Data Files, Open Course, Check for Open Seats, Add Student to Wait-list, Notification, Drop Student from Wait-list, Increase Course Capacity, and Close Course. *Table 2* explains how each algorithm 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**
	Student X wishes to enroll in Course

Add Student to Wait-list	**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**

Table 2

Description of Algorithms

4.2 Prototype Features and Capabilities

During the prototype demonstration, the presenters will show the capabilities of five different users, including: an Administrator, an Advisor, a Scheduler, a Student, and a Visitor. Each user has a different user interface that will be presented when they log in. The Administrator user will monitor all advisor and scheduler activity. Though the Administrator has the highest privileges, they will not be able to delete this activity. This user also has the capability of looking at wait-lists, students, and courses.

The Advisor user can search for students and courses. When searching for a student, this user will be able to view two lists: courses that student has enrolled into, and wait-lists that student has registered for. Clicking on a wait-list will display every student on that wait-list. The Advisor can make a couple of changes here: they can change the order in which students are represented on the wait-list, or they can remove a student from the wait-list. When searching for a course, the Advisor can view how many students are enrolled into that course, the course capacity, and if there is a wait-list for the course. Clicking on the course will display the course description, and the name or University Identification Number (UIN) of each student enrolled into the course. On that same page, the Advisor can view the wait-list and the students on that list. They also have the capability to put a student on a wait-list.

The Scheduler user can search for a course, displaying the number of students enrolled into that course and the capacity. This user can click on a course to view the wait-list for a course. If the course does not have a wait-list, but it is at maximum capacity, the Scheduler can open a wait-list. They can also do the opposite and remove the wait-list. Schedulers will have the capability to increase or decrease the wait-list capacity. After registration has ended, the Scheduler can view the analytics for a single course or a group of courses. This data will be used for future predictions on whether to increase or decrease a courses capacity.

The Student user will have to create a profile providing the following information: first name, last name, UIN, email address, and phone number. This information will be used for either display reasons, or for notification reasons. The user will also choose whether they want to receive notifications via email and/or text message. Once this has been done, this user can search for courses. Clicking on a course will display the description along with pre-requisites for the course. If the course has a wait-list that is enabled, they can add themselves to the wait-list, which will display their position.

The Visitor user can view static pages designed to give valuable information pertaining to CORSICA and courses. These pages will include: an about page, a FAQ page, a log in page, and a page to search for courses and their description.

4.3 Prototype Development Challenges

The challenging part of the prototype will be completing every scenario possible without it failing. *Table 3* displays the objectives and their corresponding 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**	**Maintaining LEO Online's layout while appending the CORSICA**

	command box menu	**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.

CORSICA Database (CDB) CORSICA's prototype database that simulates Banner.

Course Reference Number (CRN): ODU given number to register for a seat in a course.

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.

Old Dominion University (ODU): 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

National Center for Education Statistics. (n.d.). *Fast Facts*. Retrieved September 15, 2014, from National Center for Education Statistics from http://nces.ed.gov/fastfacts/display.asp?id=372