Lab II – CORSICA Prototype Product Specification

Team Red

Nicholas LoGioco

CS411W

Janet Brunelle

October 15th, 2014

Version I

Table of Contents

[1      Introduction.](http://ple.odu.edu/courses/201410/cs411w/modules/3/4/4#_Toc157958548) 3

[1.1       Purpose.](http://ple.odu.edu/courses/201410/cs411w/modules/3/4/4#_Toc157958549) 3

[1.2       Scope.](http://ple.odu.edu/courses/201410/cs411w/modules/3/4/4#_Toc157958550) 6

[1.3       Definitions, Acronyms, and Abbreviations.](http://ple.odu.edu/courses/201410/cs411w/modules/3/4/4#_Toc157958551) 8

[1.4       References.](http://ple.odu.edu/courses/201410/cs411w/modules/3/4/4#_Toc157958552) 9

[1.5       Overview.](http://ple.odu.edu/courses/201410/cs411w/modules/3/4/4#_Toc157958553) 10

[2      General Description.](http://ple.odu.edu/courses/201410/cs411w/modules/3/4/4#_Toc157958554) 10

[2.1       Prototype Architecture Description.](http://ple.odu.edu/courses/201410/cs411w/modules/3/4/4#_Toc157958555) 10

[2.2       Prototype Functional Description.](http://ple.odu.edu/courses/201410/cs411w/modules/3/4/4#_Toc157958556) 12

[2.3       External Interfaces. 1](http://ple.odu.edu/courses/201410/cs411w/modules/3/4/4#_Toc157958557)5

**List of Figures**

Figure 1 Process Flow. 6

[Figure 2 Major Functional Components Diagram.](http://ple.odu.edu/courses/201410/cs411w/modules/3/4/4#_Toc157958557) 11

[Figure 3 Database Schematics.](http://ple.odu.edu/courses/201410/cs411w/modules/3/4/4#_Toc157958557) 15

**List of Tables**

[Table 1 Real World Product and Prototype Comparison.](http://ple.odu.edu/courses/201410/cs411w/modules/3/4/4#_Toc157958557) 7

[Table 2 Description of Algorithms.](http://ple.odu.edu/courses/201410/cs411w/modules/3/4/4#_Toc157958557) 12

**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 for seats. 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, students may be forced to sign up for courses they do not need, only to maintain status as a full-time student. Courses reaching capacity may also lead to advisors receiving a lot of emails from students to force enroll them into that course.

**1.1 Purpose**

Wait-lists at post-secondary schools are becoming increasingly necessary. ODU has Computer Science students working on a wait-list solution called CORSICA, or COmputeR ScIenCe wAit-list. CORSICA will send notifications to students on a wait-list when a seat becomes available in a closed course. The notification will be sent via email and/or text message to students on the wait-list.

CORSICA's main goal is to efficiently notify students when a seat for a closed course has become available. As students approach graduation, the importance of getting 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 its wait-list. CORSICA has the following 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.

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 after it has reached capacity. 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 or not they want to receive notifications via email and/or text message.

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, 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 it has been closed. Another feature of CORSICA is that the Graphical User Interface (GUI) is handled over the Internet. Other features include the ability for students to sign up using CRNs for a wait-list, go onto the CORSICA website and view their position on a wait-list, and 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 of these sections open; students also have the option to register for a specific Course Registration Number (CRN) for each section. 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 these three parts. Therefore, CORSICA will make the student register either for three wait-lists or none of them. *Figure 1* shows the process flow of CORSICAs product.

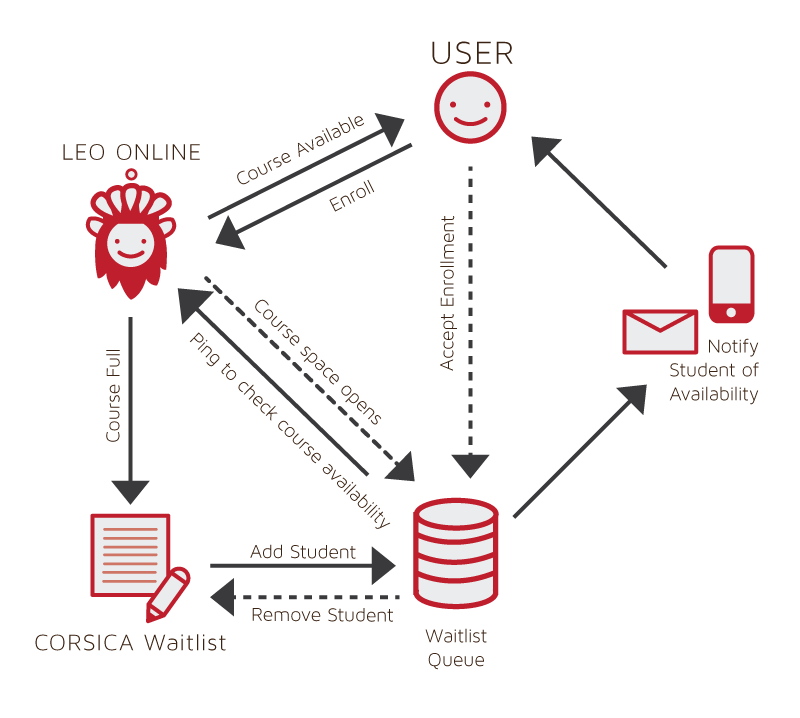


Figure 1

Process Flow

**1.2 Scope**

CORSICA team will develop a prototype with reduced capabilities to show mitigation of risks and proof of concept. During the prototype demonstration, CORSICA will use fake data to handle different scenarios. The fake data used during the prototype demonstration 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 when registration begins, when a course becomes full, when a student puts themselves on a wait-list for a course, when a spot opens up for the course triggering a notification, when student registers for the course, when student does not register for the course, and when registration ends. CORSICA will be an automated system, ensuring accuracy and fairness. The notification system will be implemented 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

**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.

**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.

**1.4 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

Nicholas LoGioco. (August 29th, 2014)*. Lab I – CORSICA Product Description*. Old Dominion University

**1.5 Overview**

The software and hardware components, capabilities and features are described in this product specification. The remaining sections include the hardware, software, and external interface architecture. The external interface used in the CORSICA prototype will be user interaction.

**2      General Description**

The real world product of CORSICA would take more resources to implement than a prototype; CORSICA will implement a prototype with reduced capabilities to provide proof of concept. This also allows the team to show mitigation of risk, which is very important to reduce adverse effects. Finally, prototyping will allow CORSICA to do experiments and get feedback from the audience.

**2.1 Prototype Architecture Description**

The CORSICA prototype is a javascript web application which can be run on all operating systems and requires a connection to the Internet. Figure 1 shows the following major functional components: database, notification, user interface, back-end algorithms, test harness, 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.

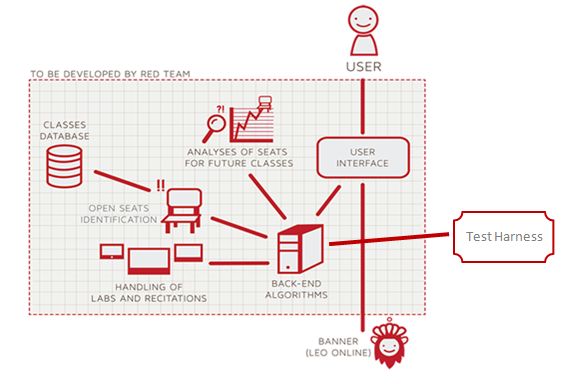


 Figure 1

Prototype Major Functional Components Diagram

The database component will contain the fake data used during the prototype. This data will be used by the GUI to display information and by the algorithms to calculate results. The data will be pre-populated to show all the capabilities of the prototype.

The notification component will be used to communicate to students when the capacity of a course is no longer full. The message sent to students will be dependent on how many seats are available and the position of that student on the wait-list.

The user interface component will display web pages to the user. These web pages will be dependent on the type of user that is logged in. There will be different access levels for the following users: Administrators, Advisors, Schedulers, Students, and Visitors.

The algorithms component will consist of seven algorithms: Load Enrollment Data Files, Open Course, Check for Open Seats, Add Student to Wait-list, Notification, Drop Student from Wait-list, and Close Course. These algorithms will be initiated by either the loading of the enrollment file, or by user interaction.

The analysis component will be used by the scheduler once the semester is over. The scheduler will use the information to predict future course capacity based on previous course capacity and wait-list size.

|  |
| --- |
|  |
|  |  |

**2.2 Prototype Functional Description**

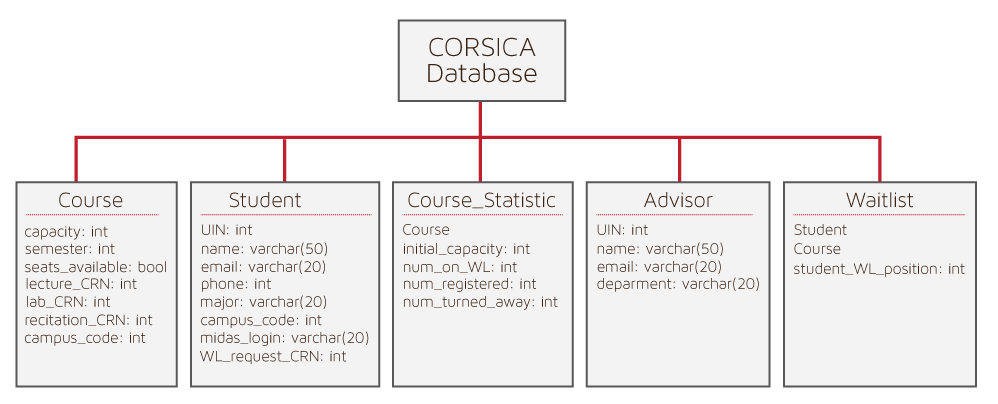
CORSICA will implement seven algorithms which are explained in *Table 2.*

|  |  |
| --- | --- |
| **Algorithm** | **Functionality** |
| **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 2

Description of Algorithms

CORSICA offers 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 multiple functionalitiy 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. All of these users will require some sort of data from the database. *Figure 2* shows the database schematics.



|  |
| --- |
|  |
|  |  |

 Figure 2

Database Schematics

**2.3 External Interfaces**

The CORSICA prototype will only need User Interfaces.

**2.3.1    Hardware Interfaces**

Not Applicable.

**2.3.2    Software Interfaces**

Not Applicable.

**2.3.3    User Interfaces**

The user will interact with CORSICA through a monitor. This monitor will be needed to display the CORSICA prototype GUI. A keyboard will be needed for the user to input data. A mouse will be needed in order for the user to navigate between web pages.

**2.3.4    Communications Protocols and Interfaces**

Not Applicable.