**CORSICA Collaborative Outline**

**Red Team**

**Old Dominion University**

**August 29th, 2014**

**CS 411 Lab 1**

# 

# **Table of Contents**

**1 INTRODUCTION (PATRICK)...................................................................................... 3**

**2 CORSICA PRODUCT DESCRIPTION (NICK).......................................................... 4**

**2.1 Key Product Features and Capabilities (NICK)................................................ 5**

**2.2 Major Components (Hardware/Software) (LATIMER).................................. 6**

**3 IDENTIFICATION OF CASE STUDY (LATIMER).................................................. 8**

**4 CORSICA PRODUCT PROTOTYPE DESCRIPTION (TONY).............................. 9**

**4.1 Prototype Architecture (Hardware/Software) (TONY).................................. 10**

**4.2 Prototype Features and Capabilities (BITASEME)........................................ 14**

**4.3 Prototype Development Challenges (LOOKMAI).......................................... 17**

**Glossary……………………………………………………………………………….……….. 19**

**References……………………………………………………………………………….…….. 21**

**Figures**

**Figure 1 - Hardware Components…………………………………………...…………...…. 7**

**Figure 2 - Real World Product and Prototype Comparison…………………...…………. 10**

**Figure 3 - Major Functional Components Diagram……………………………...……..… 11**

**Figure 4 - Description of Algorithms…………………………..………………………..…. 12**

**Figure 5 - Risk Table and Matrix……….……………………………...………….……..... 16**

**Figure 6 - Prototype Challenges………………………………………...………….……..... 17**

# **CS 411W LAB I - PRODUCT DESCRIPTION DOCUMENT**

# **1** **INTRODUCTION**

# **2** **CORSICA PRODUCT DESCRIPTION**

## 2.1 Key Product Features and Capabilities

## 2.2 Major Components (Hardware/Software)

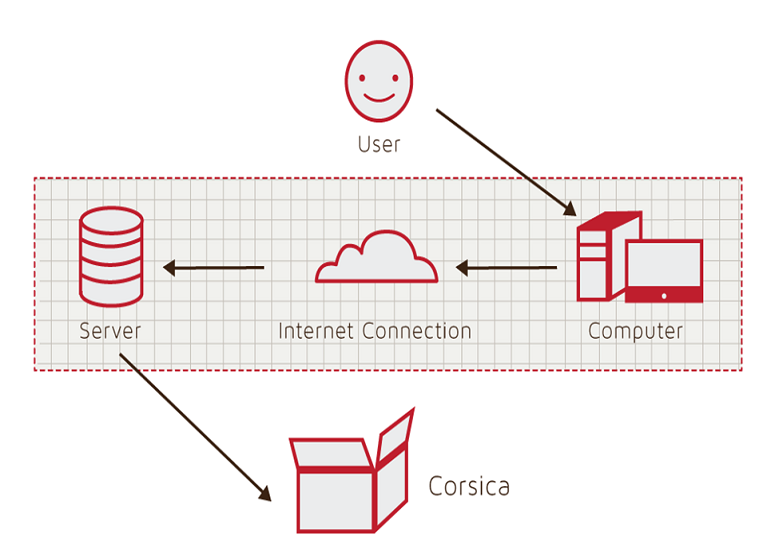


Figure 1

Hardware Components

# **3** **IDENTIFICATION OF CASE STUDY**

# **4** **CORSICA PRODUCT PROTOTYPE DESCRIPTION**

|  |  |  |
| --- | --- | --- |
|  | **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** |

Figure 2

Real World Product and Prototype Comparison

## 4.1 Prototype Architecture (Hardware/Software)

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

Figure 3

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 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** |
| **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** |

Figure 4

Description of Algorithms

## 4.2 Prototype Features and Capabilities

|  |  |
| --- | --- |
| **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 |

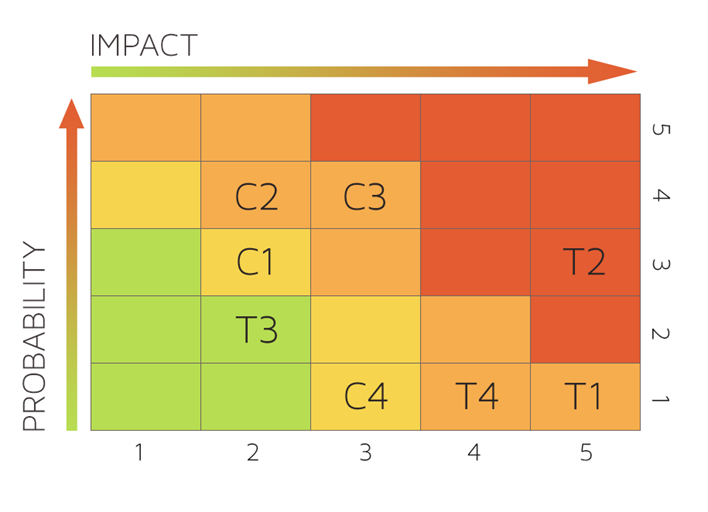


Figure 5  
Risk Table and Matrix

## 4.3 Prototype Development 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** |

Figure 6  
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

**\*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.

**\***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>

2. 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