**Software Requirements Specification**

**for**

**WUMBO**

**Version 1.0 approved**

**Prepared by Roi Chico, Kristen Marenco, Luis Gonzalez, Francisco Gudino**

**Richard Cross**

**10/2/2017**

# Table of Contents

Table of Contents................................................................................................................... 2

Revision History.....................................................................................................................3

1. Introduction................................................................................................................ 4
   1. Purpose........................................................................................................... 4
   2. Intended Audience and Reading Suggestions................................................ 4
   3. Product Scope................................................................................................ 4
   4. Definitions, Acronyms, and Abbreviations .................................................. 4
   5. References......................................................................................................5
2. Overall Description.................................................................................................... 6
   1. Product Perspective........................................................................................ 6
   2. Product Functions...........................................................................................6
   3. User Classes and Characteristics....................................................................6
   4. Operating Environment.................................................................................. 6
   5. Design and Implementation Constraints........................................................6
   6. User Documentation...................................................................................... 7
   7. Assumptions and Dependencies.................................................................... 7
   8. Apportioning of Requirements...................................................................... 7
3. External Interface Requirements............................................................................... 8
   1. User Interfaces............................................................................................... 8
   2. Hardware Interfaces....................................................................................... 8
   3. Software Interfaces........................................................................................ 8
   4. Communications Interfaces........................................................................... 8
4. Requirements Specification....................................................................................... 9
   1. Functional Requirements............................................................................... 9
   2. External Interface Requirements................................................................... 9
   3. Logical Database Requirements.................................................................... 9
   4. Design Constraints......................................................................................... 10
5. Other Nonfunctional Requirements........................................................................... 11
   1. Performance Requirements............................................................................11
   2. Safety Requirements...................................................................................... 11
   3. Security Requirements................................................................................... 11
   4. Software Quality Attributes........................................................................... 11
   5. Business Rules............................................................................................... 11
6. Other Requirements................................................................................................... 12

Appendix A: Glossary............................................................................................................ 13

Appendix B: Analysis Models............................................................................................... 14

Appendix C: To Be Determined List......................................................................................15

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 1. Introduction

This document will start by giving the users an overview of what the software is intended to do and give a detailed list for developers of each function that needs to be incorporated.

## 1.1 Purpose

The Web User Major Bound Organizer (WUMBO) will generate which classes the user must take each term to graduate. The organizer will update automatically after the user has met prerequisites and when the Admin profile opens classes for each term.

## 1.2 Intended Audience and Reading Suggestions

This documentation is intended for developers and users. The users should focus on sections 1 through 3 to have a description of what the project is and what it does. The developers should read the rest which contains the detailed information on how to create the project.

## 1.3 Product Scope

The product that will be produced is Web User Major Bound Organizer (WUMBO). It will generate which classes the user must take each term in order for that user to graduate. The organizer will update automatically after the user has met prerequisites as well as when the Admin profile opens or closes classes for each term.

The product will not enroll the user in the classes, nor will it guarantee students a spot in any of the classes needed.

Upon (WUMBO) release, students will be able to have a true schedule planner that will help improve students graduating on time success. It will also allow department head quick access to students schedule, as well as an improved, and more efficient way of editing class availability.

## 1.4 Definitions, Acronyms, and Abbreviations

* WUMBO - Web User Major Bound Organizer
* AIAP - Auto Individualized Assessment Plan
* IAP - Individualized Assessment Plan
* CSNS - Computer Science Network Services
* CS - Computer Science
* CS3 - CSNS server
* CSS - Styling for a web page
* HTML - Hyper Text Markup Language
* CSNS - Computer Science Network Service
* CSULA - California State University, Los Angeles

## 1.5 References

* <http://getbootstrap.com/docs/4.0/getting-started/introduction/>
* <https://bootsnipp.com/>

# 2. Overall Description

Our group's web program (Wumbo) will allow all CS major students to plan out their classes for the whole year or even their entire stay at CSULA.

## 2.1 Product Perspective

This product is completely independent of any other software products.

## 2.2 Product Functions

* Create user: this will create a new student account
* Login: this will create a session for the student that logins and will go to the home page
* Update tree: this will happen automatically. The tree will make sure all classes are available and prerequisites are met and organize accordingly
* Create tree: this occurs when a new user logins in, classes will be organized for them
* Update course: this will allow a student to look at other options just in case they cannot take

## 2.3 User Classes and Characteristics

Admin Class

* Has ability to set a course from available to not available
* Can view a list of all students registered on the website

Student Class

* Has an individualized assessment plan (IAP) based on courses the student has taken

## 2.4 Operating Environment

This web program that we are creating will be up in the CS3 server, including all the information that we will be using such as student's information and profile picture.

## 2.5 Design and Implementation Constraints

There should not be any design or implementation constraints.

## 2.6 User Documentation

There will not be any third party documentation to refer to.

## 2.7 Assumptions and Dependencies

This software relies on the Tomcat server and the MySQL database that the CS3 server will be providing. It also will heavily rely on the CSS provided by Bootstrap and Bootsnipp.

## 2.8 Apportioning of Requirements

There are no known requirements that will be delayed for future versions.

# 3. External Interface Requirements

This web program will be interacting with a database software called MySQL, which will hold all the students information. Allowing the web program to pull information as when the student’s signs in.

## 3.1 User Interfaces

* Error message that shows when user submits the wrong username or password.
* Menu bar that shows in all pages.

## 3.2 Hardware Interfaces

This software does not have a hardware interface.

## 3.3 Software Interfaces

* CS3 server which includes MySQL and the Tomcat server.
* API: Bootstrap

## 3.4 Communications Interfaces

* This software will be using an FTP to manage the application files.
* HTTP will be used since this is a web program
* It will be compatible with all web browsers

# 4. Requirements Specification

1.1 The system shall have a registration page for users.

1.2 The system shall have a login page for users.

1.3 The system shall have a home page for users.

1.4 The system shall allow the users to see which courses they have taken.

1.5 The system shall allow the users to add classes.

1.6 The system shall allow the users to manage their classes.

1.7 The system shall have a webpage with a visual of the user’s IAP.

1.8 The system might allow the admin to see what classes are being taken by students.

## 4.1 Functional Requirements

1. The system shall validate the user’s password.
2. The system shall validate the user’s username.
3. The system shall verify a student has passed a course.
4. The system shall adjust the IAP when a student passes a course.
5. This system shall adjust the IAP when a student fails a course.
6. The system shall adjust the IAP when the admin has toggled a course available/unavailable.

## 4.2 External Interface Requirements

1. Student’s schedule
   1. The algorithm that creates the IAP, that follows the student’s progression throughout their bachelor's, will interact with the MySQL database on the CS3 server. When the student verifies which prerequisites they have met the student IAP table will be updated on the database. When the Admin toggles courses available/not available the database will be updated.
   2. The IAP will be visible to the user on a webpage.

## 4.3 Logical Database Requirements

* Using MySQL, the following tables will be present
  + student
    - id - creates a unique id for each user
    - username - a user’s unique username for the system
    - email - this will be for future reference and emailing users
    - cin - connects to the cins table, which already contains students from the school system
    - password - a user’s password to login into the system
  + cins
    - cin - students cin which is already is system
    - gpa - GPA of student
    - is\_used - determines whether an account has already be created using the particular cin
  + student\_iap
    - student\_id - links to the student
    - course1
    - course2
    - course3
    - course4
    - course5
    - course6
      * each course a student takes stores the id of said course. If all courses are not currently being taken, then the remaining course cells will be nil.
  + courses
    - is\_available - determines if the class is being offered
    - code - the class code
    - name - name of the course on CSNS
    - units - number of units for the class
    - coordinator - the coordinator of the course on CSNS
    - semester - which semester the course is being offered (season year)

## 4.4 Design Constraints

There should not be any design or implementation constraints.

# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

* There should be enough terminals as there are administrators.
* This system should account for the total population of students in the CS major.
* This web program should allow students to see the different CS courses they can take

## 5.2 Safety Requirements

* Data in MySQL shall be kept safe
* Password for the admin should only be known to admin

## 5.3 Security Requirements

* The only security concern for this software is the student profile.

## 5.4 Software Quality Attributes

* Availability: this product shall be available to all CS major students
* This product shall be available to the faculty of the CS department
* Maintainability: this product shall be prepared to be maintained for future updates
* Portability: this product shall be designed to be accessed on many forms of hardware, e.g. PCs and mobile web browsers
* Reliability: this product shall be able to perform its intended purposes
* Testability: this product shall be rigorously tested
* Usability: this product shall require a minimal learning curve to access its features

## 5.5 Business Rules

Only administrators will be in control of the administrator account to manage and oversee student profiles.

# 6. Other Requirements

There are no other requirements that need to be documented.

# Appendix A: Glossary

N/A

# Appendix B: Analysis Models

N/A

# Appendix C: To Be Determined List

N/A