

# Software Requirements Specification

for

## Project\_Chronic

Version 1.0

Prepared by

Group Name: Team\_BanZai

James Toan Do  
Jaiteg Mundi  
Thomas Dill  
Riley McGuigan

11596503  
11422441  
11446780  
11538824

toan.do@wsu.edu  
jaiteg.mundi@wsu.edu  
thomas.dill@wsu.edu  
mcguiganriley@gmail.com

Date: October 7<sup>th</sup>, 2018

<b>REVISIONS</b>	<b>III</b>
<b>1 INTRODUCTION</b>	<b>4</b>
1.1 DOCUMENT PURPOSE	4
1.2 PRODUCT SCOPE	4
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW	5
1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS	5
1.5 DOCUMENT CONVENTIONS	5
1.6 REFERENCES AND ACKNOWLEDGMENTS	5
<b>2 OVERALL DESCRIPTION</b>	<b>6</b>
2.1 PRODUCT PERSPECTIVE	6
2.2 PRODUCT FUNCTIONALITY	6
2.3 USERS AND CHARACTERISTICS	6
2.4 OPERATING ENVIRONMENT	6
2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS	6
2.6 USER DOCUMENTATION	7
2.7 ASSUMPTIONS AND DEPENDENCIES	7
<b>3 SPECIFIC REQUIREMENTS</b>	<b>7</b>
3.1 EXTERNAL INTERFACE REQUIREMENTS	7
3.2 FUNCTIONAL REQUIREMENTS	8
3.3 BEHAVIOUR REQUIREMENTS	8
<b>4 OTHER NON-FUNCTIONAL REQUIREMENTS</b>	<b>9</b>
4.1 PERFORMANCE REQUIREMENTS	9
4.2 SAFETY AND SECURITY REQUIREMENTS	9
4.3 SOFTWARE QUALITY ATTRIBUTES	9
<b>5 OTHER REQUIREMENTS</b>	<b>10</b>
<b>APPENDIX A – DATA DICTIONARY</b>	<b>11</b>
<b>APPENDIX B - GROUP LOG</b>	<b>12</b>

## Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Initial Draft version 1.0	Thomas Dill James Toan Do Jaiteg Mundi Riley McGuigan	Initial draft of the SRS, uploaded to GitHub	10/7/18

# 1 Introduction

Our project will help a Washington State University (WSU) Vancouver Student enrolled or interested in majoring in Computer Science a comprehensive “road map” of the classes they need to take in order to graduate with a Bachelor’s of Science degree in Computer Science.

## 1.1 Document Purpose

Version 1.0

This document will provide the Software Requirement Specification for Team BanZai’s project. Included in this documentation will be a short description of the project, the intended scope, audience, and any relevant technical documentation such as definitions or software terms.

The project will contain three main pages: The “splash page” (landing page) containing the GUI, a second page made up of all the courses as presented in the 2018 Computer Science “List of Classes”, and a third page containing the history of classes taken.

The prospective student, hereafter referred to as ‘user’, will be presented with a graphical user interface containing a small graphic of all applicable classes, the class name, and recommended year/semester they should take a class. The user will be able to click on each class and a new window will open containing expanded information on the class. This will be the full title of the class and a brief description of the class as taken from the WSU Course Catalog. A space will be provided for a user to input the name of the instructor, date/time, and course location. Finally, the user will be able to input their final grade in the class if desired.

A second reference page will contain a full list of all the Computer Science classes offered at WSU Vancouver and a description of each taken from the WSU Course Catalog.

The third and final page will be the User’s History of classes, and a planner for future classes. In addition, a user will be able to see the year/semester they took a particular class and their final grade along with their GPA.

## 1.2 Product Scope

The scope of this project is to make it easier for a prospective or current Computer Science (CS) student to view the CS requirements and help plan out their student career at WSU Vancouver. We intend to use web crawling and web scraping to provide the most up to date information to keep the document (project) current and future proof.

As this will be a web application, we intend to use HTML/CSS in the development of the web pages. HTML/CSS offer extensive support and documentation in the creation of web pages. Javascript interfaces very well with HTML/CSS, and does not require a compiler. Therefore, once we upload the project onto a server, it should be self contained.

At this time, no software has been identified for the Graphical User Interface (GUI) portions of the project. This is currently being researched. Said software must be extensible for the web environment we are developing for, and self contained requiring no extra “packages” or software for the user interface beyond the typical web browsers. We will be optimizing the project for the Google Chrome web browser.

### **1.3 Intended Audience and Document Overview**

The audience for this document includes: Development staff, intended ‘client’, Dr. Zhao (CS320 instructor) and Computer Science major students.

Document Overview: This SRS is divided into five sections and two Appendixes. See the table of contents for each section.

### **1.4 Definitions, Acronyms and Abbreviations**

CSS: **Cascading Style Sheets**  
FERPA: **Family Education Rights and Privacy Act**  
GUI: **Graphical User Interface**  
GPA: **Grade Point Average**  
HTML: **Hypertext Markup Language**  
HTTP: **Hyper Text Transfer Protocol**  
SRS: **Software Requirement Specification**  
WSU: **Washington State University**

### **1.5 Document Conventions**

#### **Formatting Conventions**

IEEE Citation Style Guide was followed in the development of this document. Specifically, Arial font size 11, 1” margins, and bold face to denote subcategories within the document.

Abbreviations encountered in the document for the first time will list the entire subject followed by the abbreviation in parenthesis. I.e.; “...Software Requirements Specification (SRS)...”

### **1.6 References and Acknowledgments**

Github Repository, “Banzai”, <https://github.com/jsmundi/Banzai>, updated 10/7/2018

## 2 Overall Description

### 2.1 Product Perspective

Project\_Chronic is a brand new product. It is designed to solve specific issues that many college students at WSU Vancouver have regarding to class schedules and prerequisites. While the overall concept has been attempted before, Team\_BanZai's approach to address those aforementioned issues will be very effective and elegant for the end-user's experience.

### 2.2 Product Functionality

1. An overview of four year course plan.
2. A calculation of users GPA
2. An overview of relevant classes.

### 2.3 Users and Characteristics

1. Users of this project would be Computer Science Students enrolled in Washington State University. They will use this program to better ascertain their schedule for each semester, what the classes are and their descriptions and there overall performance in the classes they have taken.
3. The most important users to satisfy are CS Majors enrolled at WSU.

### 2.4 Operating Environment

This is a web application developed using HTML/CSS and Javascript. As such, it was tested and optimized for the Google Chrome Web Browser. Any machine able to use Google Chrome should be able to use this software.

### 2.5 Design and Implementation Constraints

In the current state of development planning there are a few constraints that will be rectify when the project is near its completion, including:

- Lack of dedicated server hosting
- Specific knowledge of HTML/CSS
- Future application maintenance
- Security of any data (grades and GPA) of the User entered into the database used by the Application. See section 4.2.

## 2.6 User Documentation

There will be no user manuals developed for this project. However potential user manuals required for this project are operational manuals that show how each of the various functions of the project are utilized. In addition documentation could also be save data for the user to refer to as a physical copy.

## 2.7 Assumptions and Dependencies

We will assume that are users will be students enrolled at WSU and are intent on getting a CS Major. We will also be assuming that the users will have existing knowledge of operating online resources and have access to their current grades. We will use existing software when necessary but have not yet identified any suitable components.

# 3 Specific Requirements

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

There will be three windows available to the user. The landing page will contain the name of the project, the names of the developers and a brief description on the intentions of the project and how it is to be used and will then contain a link the project itself.

The main page of the project will contain an image that lists each of the classes that will be taken each year by freshmen, sophomores, juniors and seniors. In this page users can click on each of the classes listed which will pull up another page. The Main Page will also contain a calculated GPA and a list of the class grades when available. Grades will be inputted by the user.

The third page is the class page and this is accessed by clicking a link on the main page. This page will contain the class description name and professor and will allow the user to set the grade they received in the class or to test how a certain grade would affect their gpa. These inputs will then be saved onto the main page.

### 3.1.2 Hardware Interfaces

Not-applicable for this project.

### 3.1.3 Software Interfaces

This application requires:

Windows Operating System (Windows 8.1 or later) or MacOS X or later.

An internet connection.

### 3.1.4 Communications Interfaces

The application requires a web browser to function properly. Google Chrome will be the default web browser used. Communication standards will utilize HTTP for web scraping/crawling of data to populate the courses and descriptions of the classes.

## 3.2 Functional Requirements

From Section 2.2:

1. An overview of four year course plan.
  - a. Create the GUI interface using the latest information from the WSU University Catalog. This will be "hard coded" into the interface.
  - b. The expansion of the window by the user.
  - c. The expansion will draw on the actual course title, description, and prerequisites by web scraping the WSU website.
2. A calculation of users GPA
  - a. This will be contained in a data file entered in by the user and calculated in the back end of the code.
  - b. Create a section for the user to input the GPA of the class.
  - c. Create a function to display the overall GPA.
3. An overview of relevant classes.
  - a. This will be data provided by web crawling the WSU webpage.
4. A simple HTML menu for navigation of the web pages

## 3.3 Behaviour Requirements

### 3.1 Use Case View

*<A use case defines a goal-oriented set of interactions between external actors and the system under consideration.*

*TO DO: Provide a use case diagram which shows the entire system and all possible actors. Do not include detailed use case descriptions (these will be needed when you will be working on the Test Plan), but make sure to include a short description of what every use-case is, who are the actors in your diagram.>*



# 1 Other Non-functional Requirements

## 4.1 Performance Requirements

1. The user should be able to use the app on a personal computer using a windows based OS or MacOS (X or above) and Google Chrome.
2. Future development will extend the functionality to other devices, such as cell phones or portable data devices (iPad).

## 4.2 Safety and Security Requirements

- The application is hosted locally on the User's Computer. If the User enters in grades and GPA, the authors of this application collectively known as 'Team Banzai' will not, and can not be held responsible if the user's system is compromised and the information is retrieved by unauthorized third parties.
- It is the user's responsibility to ensure any data entered into this application is safeguarded by them, including not sharing usernames or passwords with other users.
- This software does not meet the requirements of FERPA as established by WSU. Please visit <https://www.ronet.wsu.edu/Main/Apps/FerpaInfo.ASP> for more information about FERPA.

## 4.3 Software Quality Attributes

### 4.3.1 Adaptability

This application will not be adaptable for any other degree program at WSU Vancouver.

### 4.3.2 Availability

The application will require an internet connection to work properly. The user will have to input course grades in order to use the GPA functionality of the application.

### 4.3.3 Correctness

The information displayed by this application will only be as current or correct as the WSU Vancouver Course Catalog.

### 4.3.4 Flexibility

The application software is written and tested for a Windows Operating System 8.1 or above, MacOS X or above, and Google Chrome. The GUI is for a display of 1280x720 resolution. Outside of these constraints, the application may/may not work.

**4.3.5 Interoperability**

While the application was developed and used in Google Chrome, it should function without modification in FireFox, Safari, Internet Explorer, and Edge. However, the software will be optimized for Google Chrome and may not display or work correctly in any other web browser.

**4.3.6 Maintainability**

The base code of the application should require no modifications or updates unless the WSU Vancouver website is updated substantially outside of its intended use. The code will be commented to enable future programmers to adjust the web crawling aspects of the application to crawl a different web page. If the WSU Vancouver CS Degree Plan is changed, then this application will no longer be applicable or useful to the user.

**4.3.7 Portability**

The software for the application will be stored locally on the user's computer. A local file on that computer will be created to store the GPA information for that user. If the user installs and uses the software on another computer, they will have to repopulate the database of the application and enter in the GPA again.

## 2 Other Requirements

At this time, there are no other requirements.

## **Appendix A – Data Dictionary**

Not applicable for SRS version 1.0 at this time. As development proceeds, more will be added to this section.

## Appendix B - Group Log

*<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist the Teaching Assistant to determine the effort put forth to produce this document>*

*9/25/18 Group Formed; Discord Channel created, e-mails confirmed, and project ideas tabled. Thomas proposed meeting on 10/2 for final selection of ideas.*

*10/2/18 Live group meeting; final project idea decided, initial designs sketched out by James; JT floated some code for the GUI, Thomas proposed management of the project and Riley floated initial algorithms for the back end. Next meeting scheduled for 10/7 on Discord.*

*10/7/18 Group meeting on Discord and Google Docs to fill out the SRS Document. Github created and initial SRS committed/pushed to Github.*