SOFTWARE REQUIREMENTS SPECIFICATION (SRS)
FOR

Team Name: White Volley Girls

Class: COMP 490/L Senior Design Project

Instructor: Edmund Dantes

Security Marking

Revision History

Revision Letter	By	Change Description	Date
0/-	Christian Jarmon	Initial Draft	12/Oct/22
A	Christian Jarmon	Added non-functional requirements to Requirements Verification chart (Table IV)	16/Oct/22

1. INTRODUCTION

1.1 Scope

This Software Requirements Specification defines the requirements for the CSUN Dashboard for COMP 490 developed by The White Volley Girls.

1.2 Product Value

The defragmentation of information for the consumers is the ultimate end goal of this project will allow them to effectively plan and coordinate their financial and short-term/long-term projections and movement.

1.3 Intended Audience

Meant for CSUN students and advisors assisting students alike.

1.4 Intended Use

This product allows for the adequate schedule and financial planning of students with or without the help of an external party.

2. FUNCTIONAL REQUIREMENTS

- ***FUNC_SRS_(1.0)*** – The program shall have 3 distinct pages: Class Planner, Professor Viewer, Major Explorer
- ***FUNC_SRS_(2.0)*** - The Class Planner shall allow the user to explore classes and create possible schedules for a given semester
    - ***FUNC_SRS_(2.1)*** - The Class Planner shall allow the user to pick classes and react as the user selects a class.
    - ***FUNC_SRS_(2.2)*** - The tool shall have dropdowns that allow the user to pick a subject and have it display all the sections available for that subject in a pre-chosen semester.
    - ***FUNC_SRS_(2.3)*** - If the user attempts to pick a class that conflicts with an already chosen class, it shall warn the user that the schedule would be invalid.
    - ***FUNC_SRS_(2.4)*** - Per class chosen, the tool shall show the user the tuition cost of their current selections and should update on every addition or removal of a class
- ***FUNC_SRS_(3.0)*** - The professor viewer shall allow the user to look at professors within a chosen department.
    - ***FUNC_SRS_(3.1)*** - The product shall allow users to rate a professor for a given class in a given subject.
        - The parameters of the rating shall be as such:
            - A 1-5 Star rating
            - A description by the user of their experience
            - The Grade the user got,
            - The Class that the rating is for
            - Whether attendance was mandatory or not
            - Type of class (Online - Async, Online - Sync, In-Person…)
    - ***FUNC_SRS_(3.2)*** - The viewer shall display contact information as available of professors within a department.
        - The viewer will provide information available about a Professor as such.
            - First and Last Name
            - Telephone Number
            - Website
            - Mail-drop location
            - Office Location
- ***FUNC_SRS_(4.0)*** - The major explorer shall allow the user to look at the class requirements for each major provided by CSUN.

3. EXTERNAL INTERFACE REQUIREMENTS

3.1 User Interface Requirements

EXTINTF_SRS_(1.0) - The software utilizes a point-and-click system for interaction with it.

3.2 Hardware Interface Requirements

EXTINTF_SRS_(2.0) - As the software is a webapp, it can run on any device that is HTML5 browser capable

3.3 Software Interface Requirements

EXTINTF_SRS_(3.0) - With the software built with ReactJS, data will be transferred back and forth by a series of GET and POST request to a Flask/Python built backend using a series of JSON files and MySQL database to store data.

3.4 Communication Interface Requirements

EXTINTF_SRS_(4.0) - For the Professor Rating System, the user input form to rate a professor will be embedded and once the user submits, it will be recorded using a POST request as stated in EXTINTF_SRS_(3.0)

4. NON FUNCTIONAL REQUIREMENTS

4.1 Security

N/A

4.2 Capacity

NONFUNC_SRS_(1.0) - As a very lightweight application, if a user were to strip mine this software of all the data needed to run it, it would only consume 19 MB's of space.

4.3 Compatibility

N/A

4.4 Reliability

N/A

4.5 Scalability

N/A

4.6 Usability

NONFUNC_SRS_(2.0) - The software provides visual cues for every action the user can take. The user needs only a mouse to operate the software.

4.7 Other

N/A

5. QUALIFICATION PROVISIONS

Qualification in this specification is interpreted as requirement verification. The following are the base definitions for the verification methods.

A – Analysis: Use of analytical data or simulations under defined conditions to show theoretical compliance. Used where testing to realistic conditions cannot be achieved or is not cost-effective. Analysis (including simulation) may be used when such means establish that the appropriate requirement, specification, or derived requirement is met by the proposed solution. Examples include the reduction, interpretation or extrapolation of test data.

D – Demonstration: A qualitative exhibition of functional performance, usually accomplished with no or minimal instrumentation. Demonstration (a set of test activities with stimuli selected by the developer) may be used to show that the CSCI, or a part of the CSCI, response to stimuli is suitable (e.g. observation of fin deployment, etc.). Demonstration may be appropriate when requirements or specifications are given in statistical terms (e.g. mean time to repair, etc.).

I – Inspection: The examination of the CSCI code against applicable documentation to confirm compliance with requirements. Inspection is used to verify properties best determined by examination and observation.

T – Test: An action by which the operability, supportability, or performance capability of the CSCI, or a part of the CSCI, is verified when subjected to controlled conditions that are real or simulated. These verifications often use special test equipment or instrumentation to obtain very accurate quantitative data for analysis.

Table IV. Requirements Verification

SRS Req. ID	Paragraph Title	Verification Method
FUNC_SRS_(1.0)	The program shall have 3 distinct pages: Class Planner, Professor Viewer, Major Explorer	Demonstration
FUNC_SRS_(2.0)	The Class Planner shall allow the user to explore classes and create possible schedules for a given semester	Demonstration
FUNC_SRS_(2.1)	The Class Planner shall allow the user to pick classes and react as the user selects a class.	Demonstration
FUNC_SRS_(2.2)	The tool shall have dropdowns that allow the user to pick a subject and have it display all the sections available for that subject in a pre-chosen semester.	Demonstration

SRS Req. ID	Paragraph Title	Verification Method
FUNC_SRS_(2.3)	If the user attempts to pick a class that conflicts with an already chosen class, it shall warn the user that the schedule would be invalid.	Demonstration
FUNC_SRS_(2.4)	Per class chosen, the tool shall show the user the tuition cost of their current selections and should update on every addition or removal of a class	Demonstration
FUNC_SRS_(3.0)	The professor viewer shall allow the user to look at professors within a chosen department.	Demonstration
FUNC_SRS_(3.1)	The product shall allow users to rate a professor for a given class in a given subject.	Demonstration
FUNC_SRS_(3.2)	The viewer shall display contact information as available of professors within a department.	Demonstration
FUNC_SRS_(4.0)	The major explorer shall allow the user to look at the class requirements for each major provided by CSUN.	Demonstration
EXTINTF_SRS_(1.0)	The software utilizes a point-and-click system for interaction with it.	Demonstration
EXTINTF_SRS_(2.0)	As the software is a webapp, it can run on any device that is HTML5 browser capable.	Demonstration
EXTINTF_SRS_(3.0)	With the software built with ReactJS, data will be transferred back and forth by a series of GET and POST request to a Flask/Python built backend using a series of JSON files and MySQL database to store data.	Demonstration

SRS Req. ID	Paragraph Title	Verification Method
EXTINTF_ SRS_(4.0)	For the Professor Rating System, the user input form to rate a professor will be embedded and once the user submits, it will be recorded using a POST request as stated in EXTINTF_SRS_(3.0)	Demonstration
NONFUNC _SRS_(1.0)	As a very lightweight application, if a user were to strip mine this software of all the data needed to run it, it would only consume 19 MB's of space.	Demonstration
NONFUNC _SRS_(2.0)	The software provides visual cues for every action the user can take. The user needs only a mouse to operate the software.	Demonstration

6. NOTES

6.1 Acronyms and Abbreviations

Table VII. Acronyms and Abbreviations

Abbreviation	Full name
CSUN	California State University, Northridge
ReactJS	Javascript library for web development
GET Request	A server to client transfer of data.
POST Request	A client to server transfer of data.
JSON	Javascript Object Notation
MySQL	Relation Database Library