
Software Requirements Specification

for

Extension Tracker

Beta version

**Prepared by
William McCord
Christian Riley
Mark Selcik
Hollis Pultz**

CPSC_430_Group_1

Jan 24, 2019

Table of Contents

Table of Contents	1
Introduction	2
Purpose	2
Scope	2
References	2
Overview	2
Project Description	3
System Overview	3
Client Characteristics	3
User Characteristics	3
Product functions	4
Requirements	5
Hosting	5
Domain and account functions	5
User Interface	6
Storage	8
Logging	9
Cost	9
Accessibility	9
Performance	10
Product	10
Use case diagrams	11
Non-requirements	14
Assumptions	14
Appendices	15
Glossary	15
Author contributions	16
Additional documents	16

1. Introduction

1.1. Purpose

This document is designed to provide a comprehensive description of the system requirements for Extension Tracker, the online tracker for time-bank-day spending in classes taught by Dr. Zeitz. Contents of this document will be easily understandable by the client, Dr. Zeitz, as well as students enrolled in CPSC430 Software Engineering at the University of Mary Washington.

1.2. Scope

Extension Tracker will be an online system for tracking the usage of time-bank days by students in Dr. Zeitz's classes. Dr. Zeitz will be able to view time-bank-day usage for each student to ensure that she does not miss the usage of a time bank day. Students will be able to register for her classes and spend time bank days using a webpage as opposed to having to personally contact Dr. Zeitz.

1.3. References

There are no additional references for Extension Tracker.

1.4. Overview

This document has been organized to give an overview of Extension Tracker, provide some background on the client, define requirements and non-requirements, and assumptions about the system. To prepare for development using the Agile approach, requirements will be written as user stories. An appendix with a glossary of uncommon terms and additional information is provided at the end of the document.

2. Project Description

2.1. System Overview

Each semester at UMW, Dr. Zeitz teaches multiple computer science classes. In each of these classes, she provides each student with three “time-bank days.” A time-bank day acts as a 24-hour extension on the due date of a selected assignment. The purpose of providing students with time-bank days is to allow them to extend the due dates of assignments if they are crunched for time. This project will allow for both the Dr. Zeitz and her students to keep track of how many extension days they have used. The Extension Tracker will be a website hosted on the UMW CS server that UMW students from Dr. Zeitz’s classes will register and login at. Students will have access to a simple page that will show their remaining time-bank days and allow them to use one or more. Dr. Zeitz will have a more comprehensive view that will allow her to look at the used time-bank days and remaining time-bank days of her students. The Extension Tracker will make providing time-bank days to students easier for Dr. Zeitz and prevent discrepancies between her and her students.

2.2. Client Characteristics

Dr. Zeitz is a computer science professor at the University of Mary Washington. During her past few years of teaching, Dr. Zeitz has found it difficult to keep track of time-bank days for every student. She reported that students do not always accurately remember their remaining time-bank days either. Furthermore, Dr. Zeitz would prefer to not receive any email or notification whenever a student wants to use a time-bank day, which is the current method. If a class of 30 students used all of their time-bank days for the semester one at a time, that would results in 90 emails in Dr. Zeitz’s inbox. The Extension Tracker will make it easy for her view the usage of her student’s time-bank days.

2.3. User Characteristics

The users of the Extension Tracker will be Dr. Zeitz and her students. Dr. Zeitz will be the only admin-level user of the website, while her students will be regular, non-privileged users. Students will only have access to their own accounts. They will not be able to view other student’s account

information or directly communicate with Dr. Zeitz. Also, given the nature of Extension Tracker, students may be in a rush to login and use a time-bank day, which should be taken into consideration during development. Finally, students are expected to have the required knowledge and skills to navigate the Extension Tracker website with minimal difficulty.

2.4 Product functions

Both Dr. Zeitz, the only privileged user, and her students, the non-privileged users, will be required to login to the Extension Tracker website each time they want to use it. Students will be able to do a few things once they are logged in: view their remaining time-bank days, use one or more time-bank days (if they have at least one time-bank day remaining), register for a class, and view the assignments for a class. Once students are done using the Extension Tracker, they can logout and close out of the website. For Dr. Zeitz, there will be more functionality available from the website. At the beginning of the semester, she will be able to create classes that students can register to. Assignments for those classes can be created and deleted. Upon logging in, Dr. Zeitz will have a dashboard showing all of the classes she is currently teaching. Once a class is selected, she will be able to view remaining time-bank data for that class by student and assignment. Other admin-only functionalities include giving students time-bank days and deleting student accounts. Once Dr. Zietz is done using the Extension Tracker, she will be able to logout and close the website.

3. Requirements

This section contains all the requirements for the project in the form of user stories. Use case diagrams for functional requirements are also provided at the end of this section.

3.1. Hosting

This section includes requirements relating to what platform the site will be hosted on, as well as the language(s)/service(s) to be used when creating the site

Non-Functional

3.1.1. Host on CSUMW

As the professor/site-owner, I want the site to be hosted on the CSUMW server, so that the site will be cost free to host.

3.1.2. The website will be created via html coding

As the professor/site-owner, I want the site to be written in html, so that initial implementation and future edits of/to the site will be doable (based on the UMW computer science curriculum) by UMW students.

3.2. Domain and account functions

These requirements involve the types of people using the site and the accounts needed for them. Certain account specific permissions are also detailed here, with others listed in the non-requirements section.

Functional

3.2.1. Professor(admin) account

As the professor/admin, I would like to log into the administrative account, so that I can manage class data. This account will allow me to view all class, assignment, and student data.

3.2.2. Student account

As a student, I would like to log into a student account, so that I can manage my time bank days. I would like to see my remaining time bank days for each class I am enrolled in that uses this site, as well as be able to spend my available time bank days.

3.2.3. Creation of student account (professor)

As the professor, I would like to add a student to a class, so that I can assist a student having difficulties with creating his/her own account.

3.2.4. Creation of student account (student)

As a student, I want to sign up for my own account with the site, so that I can manage my time bank days.

3.2.5. Agree to terms of service

As a student, I want to read and confirm that I have read the terms of service for the site, so that I know what I will be held accountable for as a user of the site.

3.2.6. Delete student account

As the professor, I want to delete a student account, so that a student that has dropped the class will no longer be on the site.

Non-Functional

3.2.7. Student account information

As the professor, I want each student account to contain a student's name, user name, password, and class he/she is enrolled in, so that I can have easy access to student information while viewing time bank spending data. I expect that students will enter this information during account creation, unless I created the account, in which case I will have added the information.

3.2.8. Initial time bank days

As the professor, I want each student account to be preloaded with three time bank days upon creation, so that a student can use time bank days as soon as he/she creates the account.

3.3. User Interface

This section details the various functions a user will be able to perform through this site. This includes details on the view for each account as well as each function one can perform through each view. Some details about views and functions are listed in the non requirements section.

Functional

3.3.1. Class creation

As the professor, I want to add courses to the site, so that students will be able to sign up for the courses of mine they are enrolled in. I will be the only site user capable of adding a class.

3.3.2. Dashboard (professor)

As the professor, I want to select a class to view from my dashboard/home-screen, so that I can view student information by class. This will be the first page I see after logging in.

3.3.3. Class view of students (professor)

As the professor, after selecting a class from the dashboard I want to view student name and time bank information in a table organized by student last name, so that I can easily see what students have time bank days left.

3.3.4. Class view of assignments (professor)

As the professor, after selecting a class from the dashboard I want to view assignments and their due dates, in a table organized by due date, so that I can select an assignment to view student time bank usage for.

3.3.5. Assignment view

As the professor, I want to view student names and the time bank days used by each student for an assignment, ordered by student last name, so that grading with this site open next to canvas requires little head movement. I should be able to choose to show all students or to show only students that used one or more time bank days.

3.3.6. Add time bank days

As the professor, I want to add time bank day(s) to a student time bank, so that I can use extra time bank days as extra credit rewards. This addition will elicit a notification on the student's view on next login.

3.3.7. Assignment creation

As the professor, I want to add an assignment along with its due date to the assignment list for a class from the class view, so that I can have a list of assignments for students to spend time bank days on.

3.3.8. Assignment deletion

As the professor, I want to delete an assignment along with its due date from the assignment list for a class from the class view, so that I can remove assignments that are cancelled.

3.3.9. Dashboard view (student)

As a student, I want to see a list of the classes I am enrolled in that use this site, so that I can select one to view my time bank information for. This will be the first page I see after logging in.

3.3.10. Join class

As a student, I want to join a class using a class registration code provided by the professor, so that I can manage my time band days for that class

3.3.11. Class view (student)

As a student, after selecting a class from the dashboard, I want to see a

list of assignments for that class along with their due dates ordered by their due dates, so that I can see what assignments will be due soon in relation to the day I am viewing the page.

3.3.12. Time bank display

As a student, I want to see in the class view a display of my number of remaining time bank days in a bright font larger than any other font on the page, so that it is immediately noticeable how many time bank days I have left.

3.3.13. Time bank usage record

As a student, I want to see how many time bank days I have used on each assignment from the class view, so that I can see where I have previously used time bank days.

3.3.14. Spend time bank day(s)

As a student, I want to spend time bank days by clicking an assignment from the assignment list and selecting from a dropdown menu how many days I'd like to spend, so that I do not have to type any information to spend time bank days.

3.3.15. Due date update

As a student, I want to see an updated due date in the class view for each assignment that I have spent one or more time bank days on, so that I know when each assignment is actually due for me.

3.4. Storage

This section details the format data will be stored in as well as specific functions related to the database.

Functional

3.4.1. Semester reset

As the professor, I want to select an option with password verification from the dashboard to clear all class data, so that I can reset the site at the conclusion of the semester.

Non-Functional

3.4.2. Database

As the professor/admin, I want all class, assignment, and student information to be stored on the CSUMW server using the already installed MySQL database software, so that the database can be manually edited by me.

3.5. Logging

This section details the actions that will be specifically logged for the admin to view.

Non-Functional

3.5.1. Logging logins

As the professor/admin, I want the website to keep a plaintext log file of all login attempts by students and myself, both successful and unsuccessful, so that I can monitor for any potential malicious login attempts.

3.5.2. Logging time bank usage

As the professor/admin, I want the website to keep a plaintext log file of all time bank activity by students and myself, so that I can have an additional record of student time bank usage incase of a discrepancy about time bank days.

3.6. Cost

This section details the total costs incurred by this project.

Non-Functional

3.6.1. Expenses

As the professor, I want the total cost of the project to be \$0.00, so that providing this interface for the students will be free of charge.

3.7. Accessibility

This section details the various design choices to make the site accessible to as many people as possible. This includes compatibility with various devices and software as well as readability requirements.

Non-Functional

3.7.1. Font Size

As the professor, I need the font size to be large, so that I and all my students can clearly read it.

3.7.2. Site Colors

As the professor, I need the colors of the website to contrast, so that myself and all my students can clearly see the text when compared to the background color.

3.7.3. Browser Compatibility

As the professor, I would like the site to be compatible with Chrome, Firefox, and Safari, so that myself as well as all my students can use the website from a variety of internet browsers.

3.7.4. Mobile format

As the professor, I would not mind the site being mobile friendly, so that my students could use it from their phones, despite it not being a feature I would take advantage of.

3.8. Performance

This section lists the specifications related to how well the website should perform on certain tasks.

Non-Functional

3.8.1. Active Connections

As the professor/admin, I want the system to support multiple users being logged in and performing actions on the site with little to no noticeable lag for the users, so that time bank operations can be done quickly.

3.9. Product

This section includes important details related to rights to the project itself.

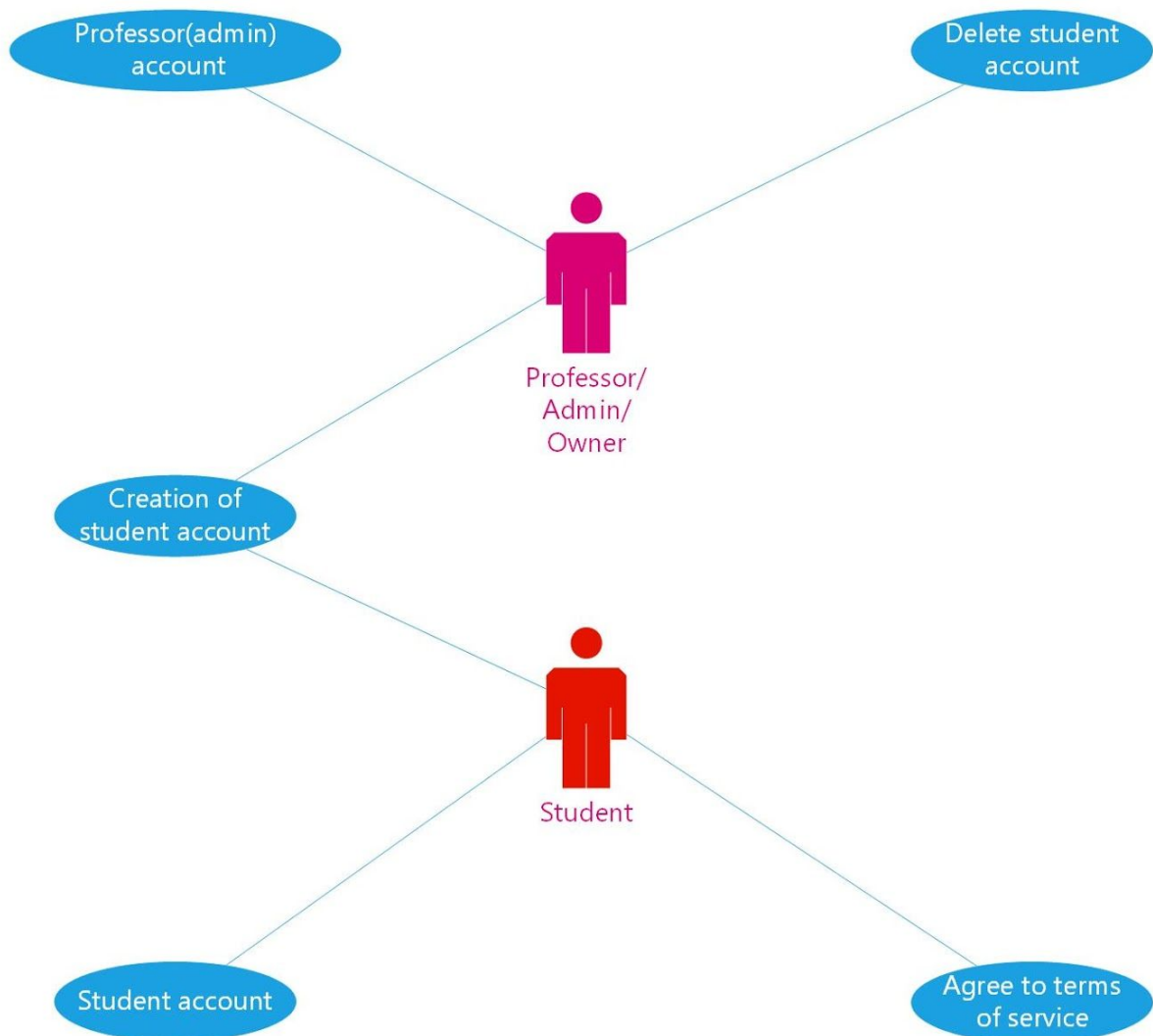
Non-Functional

3.9.1. Product Rights

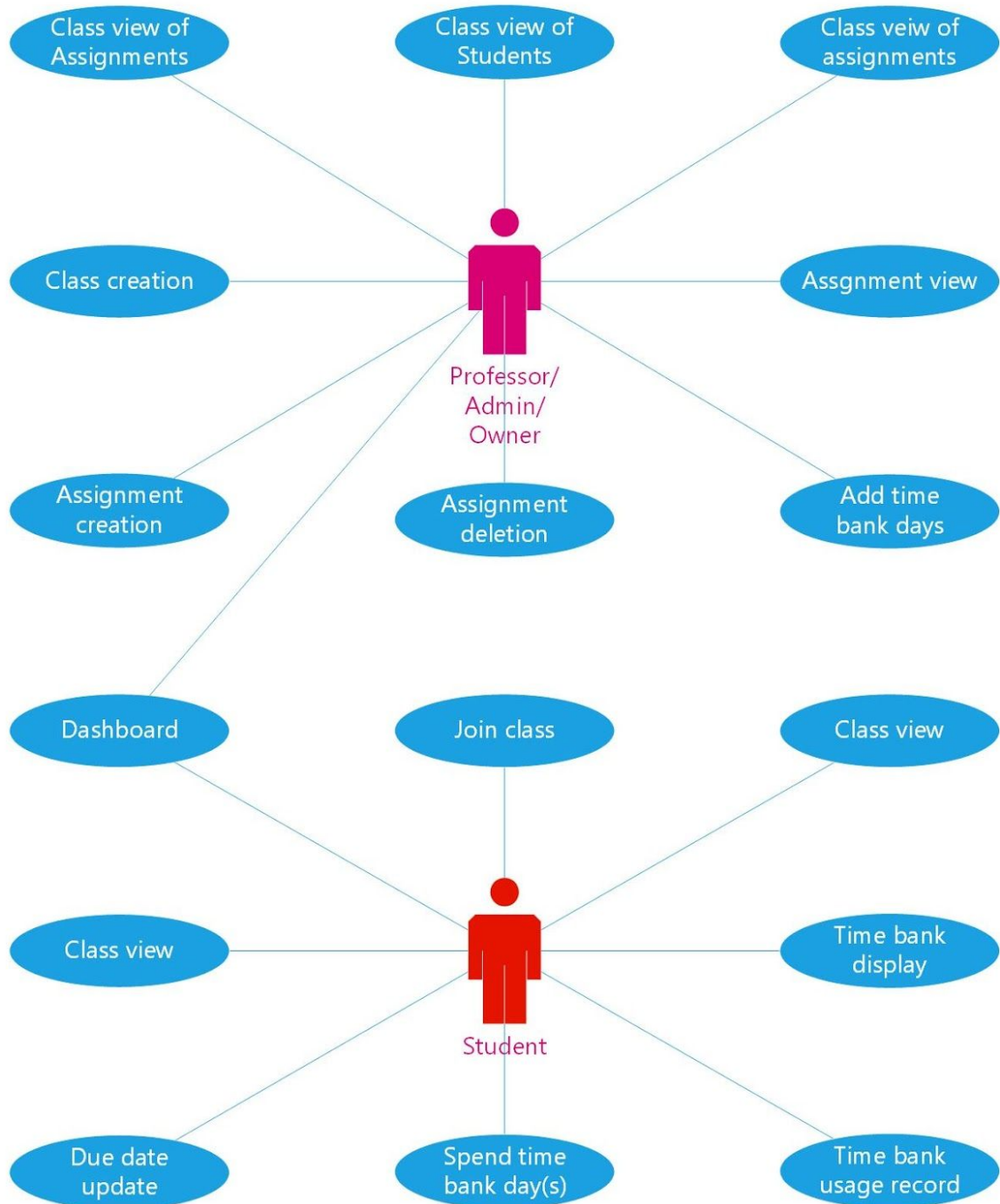
As the professor/owner, I want the full rights to the product given to me upon project completion, so that I can do whatever I want with/to the site in the future.

3.10. Use case diagrams

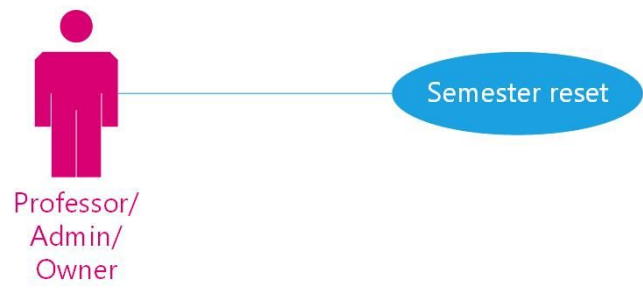
Domain and account functions



User interface



Storage



4. Non-requirements

- 4.1.** This section entails the requirements the client has on what cannot be included in the final product made by the implementation team. These requirements are negotiable if new ideas can be conceptualized but for the large part remain immutable. These items are by explicit request from the client and should any of them be infringed upon, the client should be notified for approval of the changes to be made accordingly.
 - 4.1.1.** Students should only be able to view/use their own time bank days.
 - 4.1.2.** Students cannot make their own accounts, they are created for the classes by the client with classes added to accounts accordingly.
 - 4.1.3.** Students cannot use time bank days on non-applicable assignments. (Group Assignments and Assignments due the last day of class)
 - 4.1.4.** Students can only view class pages that they are attending.
 - 4.1.5.** No calendar to display assignments and etc.
 - 4.1.6.** Data for classes does not carry over between semesters, deletion of classes should be handled manually by Administrator.
 - 4.1.7.** Extension Tracker cannot send e-mails to the professor.
 - 4.1.8.** Aggregate running cost of the project should not go above \$0.00
 - 4.1.9.** Extension Tracker does not maintain a history of time bank usage beyond the current semester, information is always deleted upon semester end.

5. Assumptions

This section is a list of assumptions we have made about how the implementation may be planned as well as some goals the end product should achieve if time is applicable to the design process.

- 5.1.** Finished product should be viewable through url while connected to umw wi-fi for ease of access while being hosted on the CS server.
- 5.2.** May be easiest to perform back-end operations using javascript or PHP due to prior class experiences in 350 (Applications of Databases).
- 5.3.** If it can be integrated well, an aggregated e-mail to be sent upon reaching an assignment due date with a list of students who chose to use time bank days could be added as a function of the extension tracker.
- 5.4.** The user interface of the system should be easily navigated and simple to operate for both the student and the administrator while also having adequate functionality to incorporate all tasks required. Keep it simple.
- 5.5.** If the project can be finished to the client's demands at an earlier date, constructing a mobile-friendly version of the web page can be an additional task.

6. Appendices

6.1. Glossary

- 6.1.1. Time-bank day:** In case any stakeholder has not attended a class taught by Dr. Zeitz, she has 3 days given at the start of a semester to be used on any assignment (other than group projects) to add a day onto the due-date. These can be used until the last day of the semester and sometimes there are opportunities to get additional days.
- 6.1.2. MySQL:** This is an open source relational database management system. Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.
- 6.1.3. PHP:** Hypertext Preprocessor is a server-side scripting language designed for web development. Primarily implemented using C with some parts of C++ for the language.
- 6.1.4. Javascript:** often abbreviated as JS, is a high-level, interpreted programming language that conforms to the ECMAScript specification. It is a language that is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm.

6.2. Author contributions

- 6.2.1.** William - Non-requirements, Assumptions, Appendices
- 6.2.2.** Christian - Requirements
- 6.2.3.** Mark - Project Description
- 6.2.4.** Hollis - Title page, ToC, Introduction, Use Case Diagrams

6.3. Additional documents

No documents from client to provide, decided instead to include some tutorials so that in case any implementation team member is new to web development they don't have to google search this stuff on their own, also provides some examples of how the project could be implemented.

- 6.3.1.** Learn an intro to HTML here in case you are new to the design process.
<https://www.codecademy.com/learn/learn-html>
- 6.3.2.** An example of how you could integrate your requests to the database on your html webpage as well as an example of good design by handling backwards compatibility with older browsers using PHP. This method

uses table-based design so if you are using stylesheets, it may get funky unless you add a css div id tag for separate formatting.

https://www.w3schools.com/php/php_ajax_database.asp

- 6.3.3.** See the above, shows how Id's work on the HTML and CSS side.

https://www.w3schools.com/html/html_id.asp

- 6.3.4.** Take this tutorial with a grain of salt as it's not an official one so its reliability is dubious; a tutorial on how to make a desktop site more mobile friendly.

<https://www.instructables.com/id/Make-your-HTML-Website-suitable-for-Mobile-Devices/>

- 6.3.5.** DISCLAIMER: Use the above documents if you wish but feel free to use your own method to implement the project if it is easier to perform based on how integration works with the CS server.