**Description of the problem**

I plan to make a typing program that will measure your words per minute or typing speed. The program will come up with random words from a predetermined list. You will be able to customise various aspects of the program, such as the difficulty and customise words. The program will prompt you to type a specified word and move onto the next word once you hit space. This will continue until the test is over and will calculate your accuracy and words per minute. You will have the option to input your name to submit your score into a leader board, otherwise it will save your score without a name. This will save your name for the next round that you do. You will not be able to log in, and the only player indicator that will be saved is the name. The program and data will be displayed within a website, which is where you will do each game.

This project meets the requirements for AH as:

Software Design and Development

* All Inputs will be validated.
* It covers 2 AH Software concepts
* OOP with an array of objects to store the name, WPM, and accuracy of players.
* Bubble sort to re-sort the scores each time a player enters a score.

Website Design and Development

* Integrate within a website and provide a suitable user interface.

**Scope, Boundaries and Constraints**

Scope

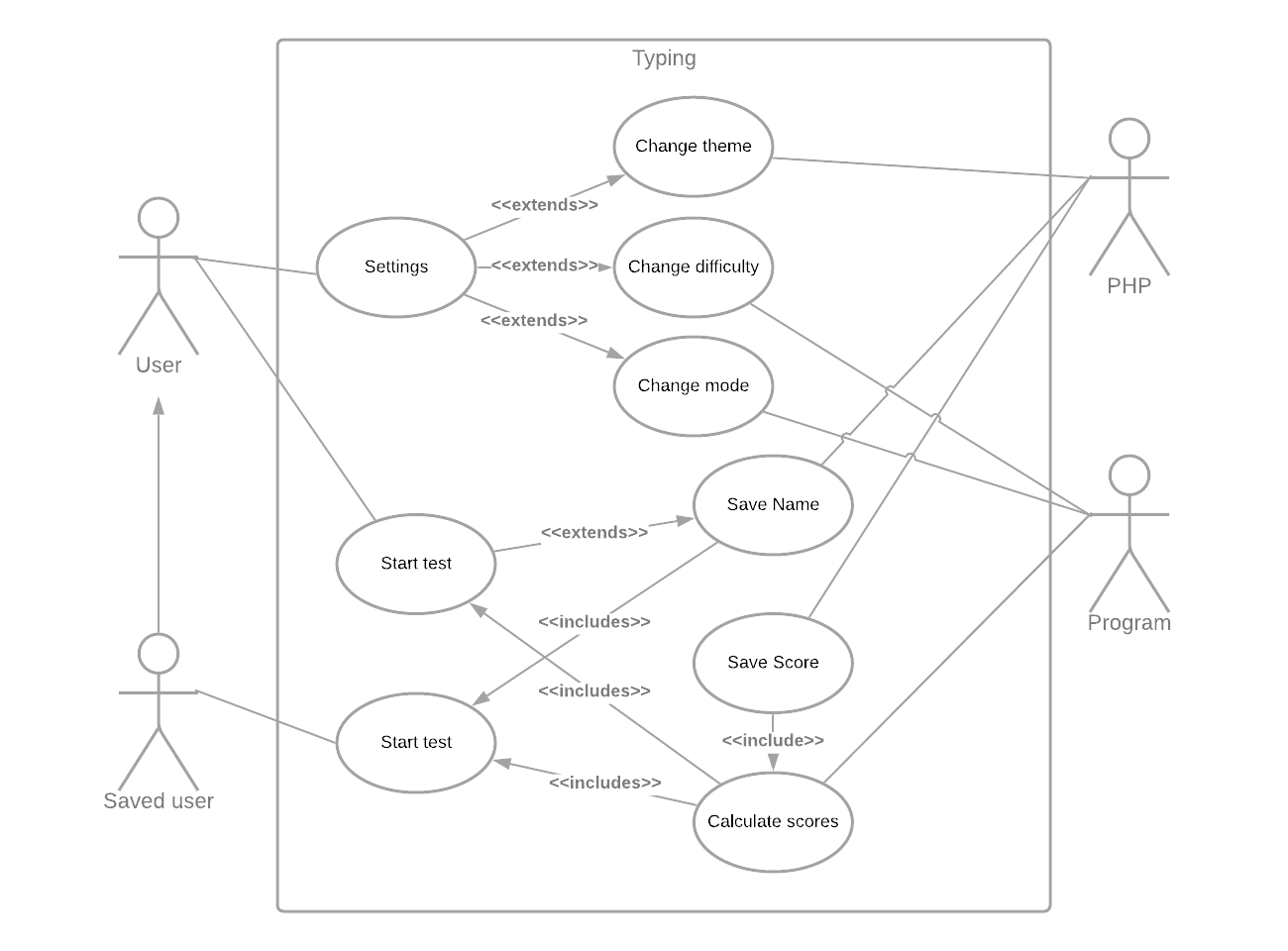
* A finished design of the project with pseudocode, UML Class diagram and wireframes to show the interface.
* A fully working implementation of the design that lets you do tests, save them on a scoreboard and retry.
* A completed test design with personas, test cases and expected outputs
* The results of final testing
* An evaluation report that covers fitness for purpose, maintainability, and robustness

Boundaries

* The top 200 words will be the default size of the array.
* 2 test types will be available: number of words typed within a specific time, or time to type a specific number of words.
* You will be able to customise the words that will be in the program and sort the words in the array by length or alphabetical.
* You will be able to save your username at the end of a test, and session variables to save the data.
* Your score will be saved into a leader board file corresponding to your difficulty.
* A leader board showing the top 10 scores in order will be shown on the website.
* Will be able to switch from light and dark theme.

Constraints

* Technicality
* The languages I will be coding in will be HTML, PHP, CSS, and JavaScript due to it being a website, and the program being integrated inside of it.
* The website will use XAMPP to create a web server to host the website which will be run on Windows 10.
* Time
* The project will be completed before Friday 19th March, so I can send the project to the SQA on time.
* Economics
* There will be no costs when developing my project as I am using software that has been licenced by the school and available to use.

**UML Use Case diagram**

**Requirements Specification**

Functional Requirements

Program (JavaScript)

1. You will be able to customise the settings of the game.
   1. The program will let you change the difficulty and the mode of the program. Default is 1 minute.
   2. The number of words in the array will be the top 200 words.
   3. The mode can be selected from specified time (2 min, 1 min, 30 secs) or specified words (25 words, 50 words, 100 words)
2. The program will let you display the words used in a list.
   1. Words will be displayed alphabetical, or by length with a bubble sort.
3. The program will play a typing game that measures your WPM and accuracy.
   1. The program will detect when you start typing in the input box and start the program.
   2. The program will get an array of random words from a file containing the stored words.
   3. The program will detect when a space bar is pressed, move onto the next word, and generate a new value in the array.
   4. The program will check whether the word you typed is correct and will change word’s text colour to red if it is wrong.
4. When the program finishes, the user’s data will be calculated.
   1. WPM (Words Per minute) – The number of words over the time taken in minutes rounded to nearest %
   2. Accuracy – The number of correct words divided by the total number of words typed rounded to nearest %

Scoreboard (PHP, JavaScript)

1. When the program finishes, the user will have the option to save their score to the leader board.
   1. Scores will be saved in an array of records in JavaScript.
   2. The user will also have the option to save their name to the leader board.
      1. If the user does not input a name, the name shows as blank.
      2. Checkbox to save your name for next rounds, which will be stored in a session variable.
      3. Any further rounds, the name box will be automatically inputted.
2. The top 10 scores will be shown on a scoreboard.
   1. The scores will be calculated using a bubble sort to put the value in

End User Requirements

1. Clean interface that is intuitive to work with.
2. Interactivity with text colour changes while doing a test (green for pass, red for fail)
3. Ability to change theme to fit the user’s needs (light/dark)

**Inputs, Processes and Outputs**

Inputs

* Settings button to change settings.
* Text input field to type words in.
* Text input field at end of test to type name.
* Submit button to end test and restart to default state.

Processes

* Editing setting variables.
* Initialising array with random words.
* Listen for character typed in field to start.
* Listen for spacebar typed in field and move on to next array element.
* Check if word is typed correctly.
* Append new value to word array.
* Calculate data.
* Restart test.
* Save Name using session variable.

Outputs

* Text colour change depending on if the word is typed correctly.
* Timer / Number of words left counting down to indicate test length.
* Finish popup showing user data.
* User data displayed in Scoreboard.

**Resources required.**

Analysis

* Text processor - Microsoft Word
* Gantt Chart – Microsoft Excel

Design

* Text processor - Microsoft Word
* Wireframing – Adobe XD

Implementation

* Text editor - Atom
* Research and website testing - Google Chrome
* Web server – XAMPP
* Repository managing – Atom and Github Desktop

Testing

* Text processor - Microsoft Word

**Project Plan**  
