Comp 1004 - Representation of Final Project Plan, Final UML Diagrams & Summary

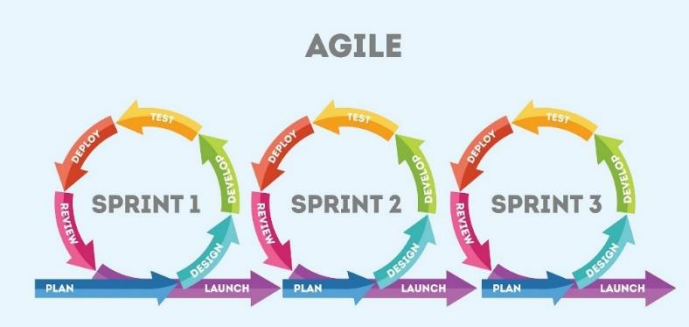
Introduction

Academics have intensive workloads requiring hundreds of hours to meet deadlines often at the cost of damaging their work-life balance affecting the mental health of the individuals. This is a study which investigated work-life balance being a predictor of anxiety depression and concluded that Educational Institutions should do more to help manage pupils' work-life balance (Sprung and Rogers, 2020). This Hue Lights Project is designed to be a useful tool to assist students and business employees manage their personal time management as a suggestion to developers and academics to implement similar systems in their products. Developing this project required an in-depth analysis of the Software Development Life Cycle methodologies and executing the Agile model successfully and consistently.

Software Development Life Cycle (SDLC)

- **Requirements Analysis**
- **Design**
- **Implementation**
- **Testing**
- **Maintenance**

A Breakdown of the SDLC comprised of 5 stages (Davis, Bersoff and Comer, 1988)



(Slawek-Polczynska, 2020)

How Has Agile Been Applied to my Project

Agile has successfully been applied to this project consistently with areas for improvement. I have managed to implement 9 sprint plans and sprint reviews every 2 weeks. This has been a personal achievement to keep consistent and certain sprints have been extremely productive. However, I have had various points of the project of significant lack of productivity due to bugs which have strongly benefited from agile allowing me to redesign the project as I go along and have been afflicted extremely badly with COVID-19 affecting my asthma holding the project back 2/3 weeks despite rapidly making progress after recovery due to very well planned architecture.

Major deviations within my project have occurred due to fixing a bug related to the SPA and form logic. Due to SPA applications with JQUERY reloading when constantly changed my original wireframes made the user go back to the form to add details due to not considering use cases had caused forms

to lose all data. Solving this logic error took weeks which is unacceptable and perhaps if I spent more time considering the architecture at the beginning of the sprints this may have been avoided. COVID-19 was outside my control and the university is aware of my extenuating circumstances however the main error of my project had been fixed helped make my final sprints extremely productive despite being so far behind.

The Sprint Plans could have been more ambitious at times, due to other subjects I have had to balance time accordingly focusing primarily on deadlines which may have put this project aside during the first half of this project as I spread this project's workload evenly which is good but too thin at times. Perhaps, I could have implemented more of my advanced features if I increased the sprint workload more although certain sprints have been very productive after overcoming certain obstacles indicating more time should have been applied to the individual sprints design phase.

Project Description

Mental Health is a major concern for education and the workplace due to intense pressure to output assignments on time. Various research demonstrates the strong relationship between anxiety and poor time management such the International Journal of Physical and Social Sciences (Beri, 2015). This Project takes inspiration from various marketplace products such as calendar applications and takes an interesting twist. This will grab user activity data from various forms and be able to calculate their time management is either good, ok or bad. Using this data, I could take things further such as visually representing the user feedback statistically using graphs or just output to lightbulbs.

JSON capabilities have been included in this project's functional requirements to be able to Save, Load or start again into the JSON format specified. Matching government GDPR guidelines has been attempted to be incorporated in various ways in regards to user personal data (GDPR), 2022) such as the 'Right to be forgotten' will be easily applied when the user clicks 'new'. Other GDPR laws will be followed by only asking the minimum amount of data required for the calculations and respecting the 'Right of Access' by showing all the data we have collected of the user visually.

Planned Usage vs Actual Usage of GDPR

This product will follow the current government guidelines of GDPR to respect individuals' rights. The regulations state that individuals must have the right to be informed that their data is being used and the right to erasure (GDPR), 2022). This project will attempt the best it can to follow government GDPR laws by adding to the features these as non-functional requirements of high priority. Due to time constraints, it may be difficult to implement all features but worth striving for such as the right to erasure could be implemented with relative ease.

In practice, only certain GDPR laws could have been implemented due to limiting time constraints this project prioritised functional requirements over non-functional requirements such as GDPR. Despite these certain laws have been implemented such as the 'right to be forgotten' may be easily applied when the person clicks new and the right of access which the user may see all the JSON data we have collected, and we have used the 'Right to be informed' in a separate page in clear sight informing the user personal data will be collected.

Functional Requirements

1. The Program must be Single Paged Application showing some form of interactivity
2. The Program must perform a JSON output to local storage
3. The Program must perform a JSON input from local storage
4. The Program must output to a lightbulb if ranked good, medium, bad to a corresponding traffic light colour RGB
5. Must have an information page to show basic advice and basic stats
6. Read a JSON file using the format specified in Project scenarios to identify how much time is spent on leisure activities
7. Output JSON to a new file to save information on how the user wishes to manage their time and deadlines
8. Have a navigation bar
9. Have a login and support different users
10. Have advanced CSS animations
11. Have a nice CSS layout that is scalable on different sized screens for mobile
12. Use a JavaScript Library such as React or Node JS
13. Have an accessibility mode that makes the text larger
14. Be able to add user schedule
15. Be able to change the schedule
16. Make the website easily detectable by search engines
17. Be able to play around with lightbulbs freely
18. Stats page to show advanced stats
19. Make a paragraph of how we are following GDPR Guidelines in the footer
20. Being able to enter their current schedule and desired schedule
21. Change colour for accessibility for colourblind people
22. Output schedule and desired on the information page
23. Output result if good or bad on information page (basic) and stats page (adv.)
24. Delete stored user data after 2 weeks (basic) or desired time (adv.) after being end duration and date
25. Disclose data usage and meeting GDPR in the footer
26. Add Lightbulb image green, amber and red in Information depending on lightbulb output
27. When the name is submitted, or the user loads last save hide the name form. Until new is pressed show it. Can be achieved with CSS or JS.

Must: 1 – 8, 14, 19, 20, 22 , 23, 26, 27

Should: 9, 13, 15, 18, 21, 24, 25

Could : 10, 11, 16, 17

Non-functional Requirements

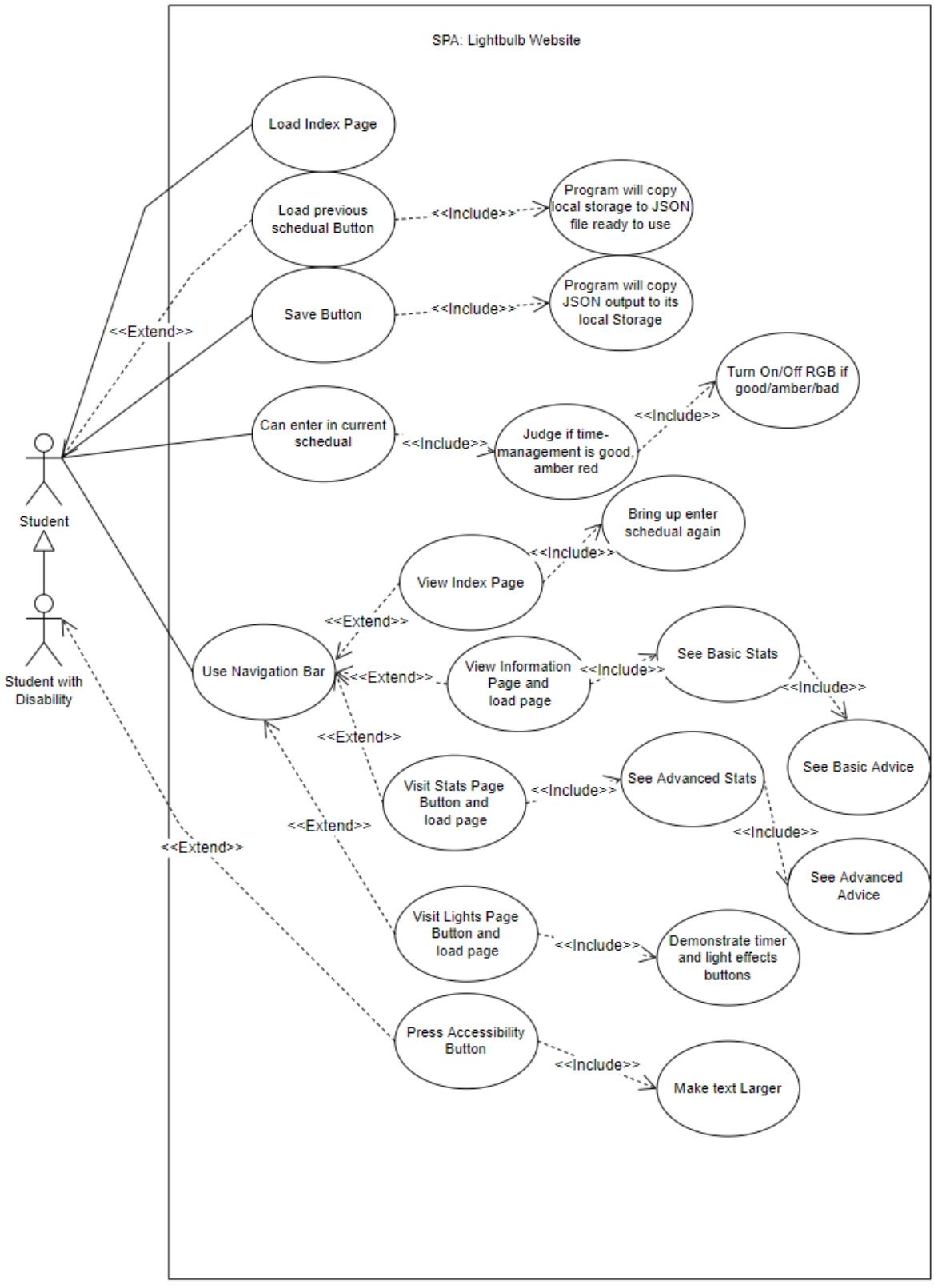
1. Be scalable on different devices by using Chromes Dev tools

[redacted]

Complete set of UML Diagrams

User Stories

Student	Student with disability
As a user of this application, I want to be able to enter my current schedule	I want to be able to make my text readable with an accessibility feature that enlarges text throughout the website making it easier to read
As a user of this application, I want to be able to change my current schedule As a user of this application, I want to save how I am going to manage my time	I want to be able to change the colour to suit my colourblind disability with a press of a button
As a user of this application, I want to see if my time management is good	
As a user of this application, I would like the website to be able to load in my previous schedule if I desire from my local storage	
As a user of this application, I want to be able to change from my index page to other pages using a navigation bar	
As a user of this application, I want to use a navigation bar to select the page and section of the website I wish to use.	
As a user of this application, I would like my personal user data to be erased after 2 weeks	
As a user of this application, I want the lights to show me visually if my schedule vs ideal schedule is good.	

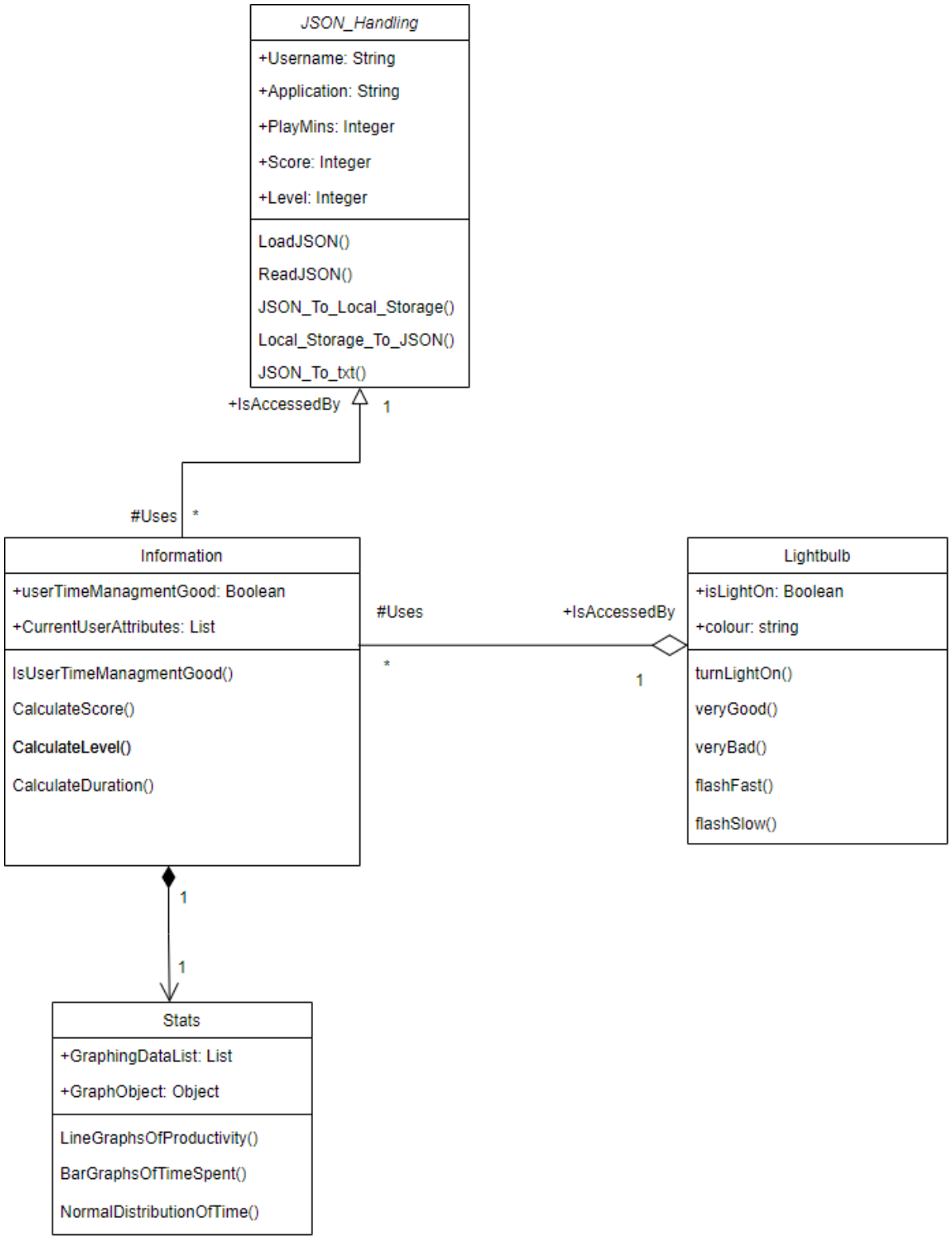

SPA: Lightbulb Website
Load Index Page
Load previous schedual Button
Program will copy local storage to JSON file ready to use
<<Include>>
Save Button
Program will copy JSON output to its local Storage
<<Include>>
<<Extend>>
Can enter in current schedual
<<Include>>
Judge if time-management is good, amber red
<<Include>>
Turn On/Off RGB if good/amber/bad
Student
Bring up enter schedual again
<<Include>>
View Index Page
<<Extend>>
Student with Disability
Use Navigation Bar
<<Extend>>
View Information Page and load page
<<Include>>
See Basic Stats
<<Include>>
<<Extend>>
See Basic Advice
Visit Stats Page Button and load page
<<Include>>
See Advanced Stats
<<Extend>>
<<Include>>
<<Extend>>
See Advanced Advice
Visit Lights Page Button and load page
<<Include>>
Demonstrate timer and light effects buttons
<<Extend>>
Press Accessibility Button
<<Include>>
Make text Larger

Use Case Descriptions

Name	Save
Description	The user will save their progress by pressing a button which will be stored in local storage and text file
Pre-Condition	N/a
Post-Condition	Updated Stats, schedule and user details are saved in txt and local storage
Error Situation	There is no txt file
System State in Case Of Error	Create a new txt file
Actors	Student and Student with Disability
Triggers	User presses the save button on any page whenever they want
Standard Process	1. Save the current JSON file to local storage by stringifying the JSON file
Alternative Process	1. Save the current JSON file to local storage by stringifying the JSON file

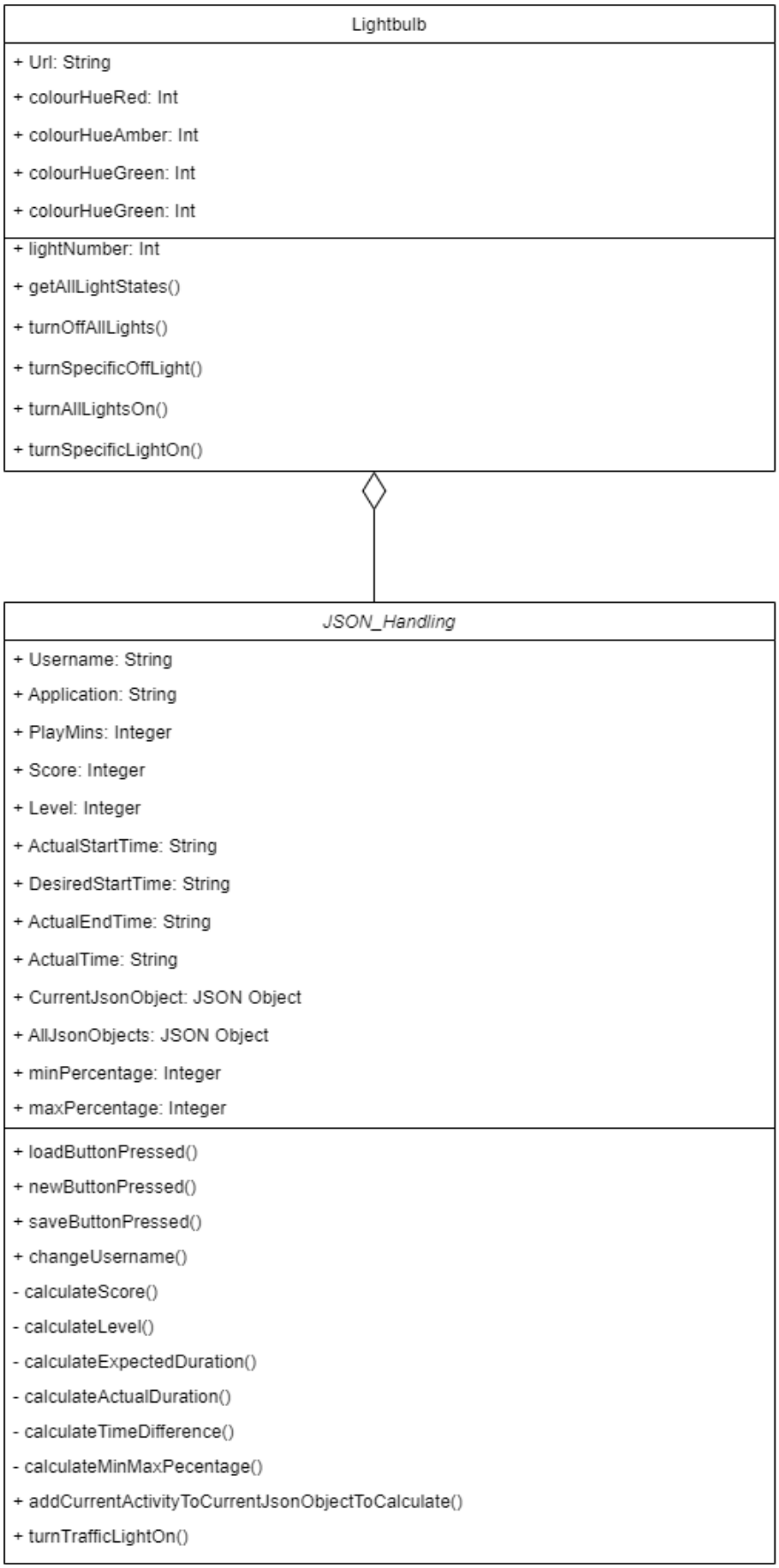
Name	Load
Description	The user will load from local storage the JSON file and store it into a JSON file which will be loaded into a List Abstract Data Type in the JavaScript. The List will be operated on.
Pre-Condition	The User must have previously saved
Post-Condition	The program loads all the data into a List to be operated on for scheduling
Error Situation	There is no txt file and no local storage or is corrupt
System State in Case Of Error	Do nothing and return an alert/error to the user saying that there is no previous save file
Actors	Student and Student with Disability
Triggers	User presses the load button on any page whenever they want
Standard Process	1. Load the local storage JSON file that is stringifide 2. Parse JSON items 3. Store as a List (able to be operated on for calculations)
Alternative Process	1. Try Load the local storage or txt file into a JSON file 2. If corrupt or no save exists 3. Do nothing and send an alert/warning to the user that there is no save

Planned – Class Diagram

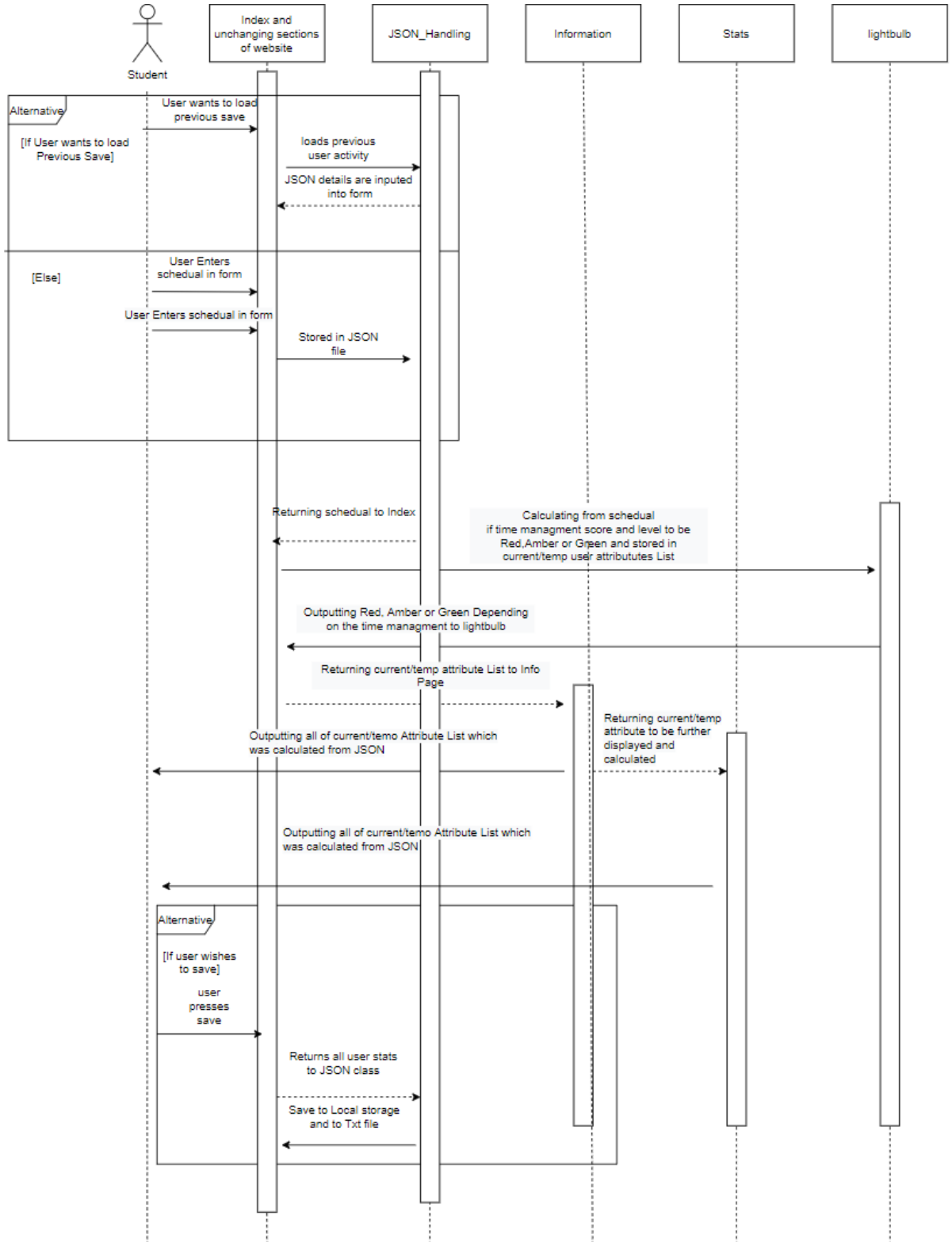


[redacted]

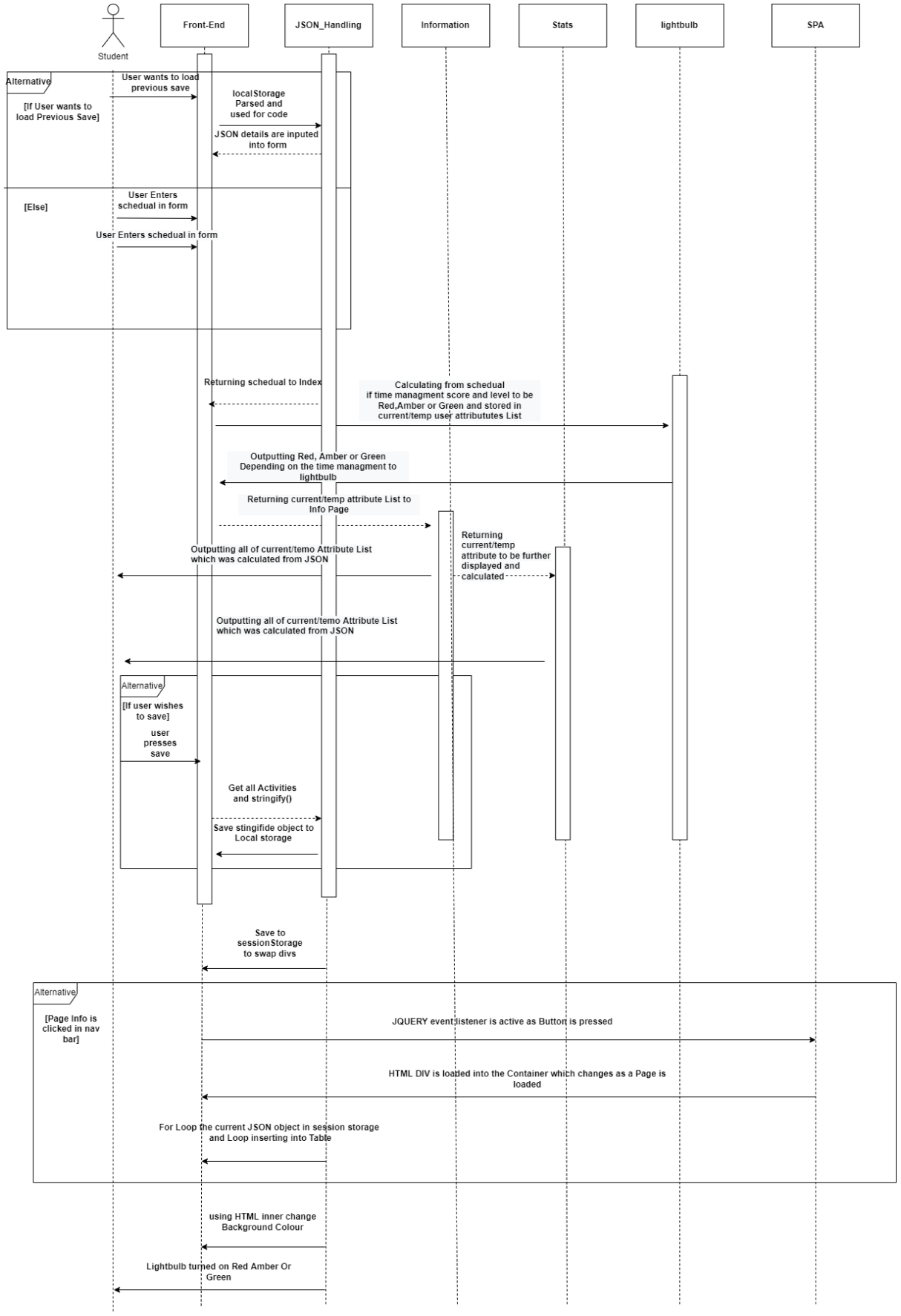
Current - Class Diagrams (simplified)



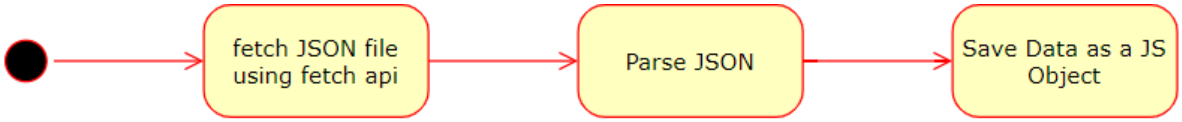
Planned – Sequence Diagram



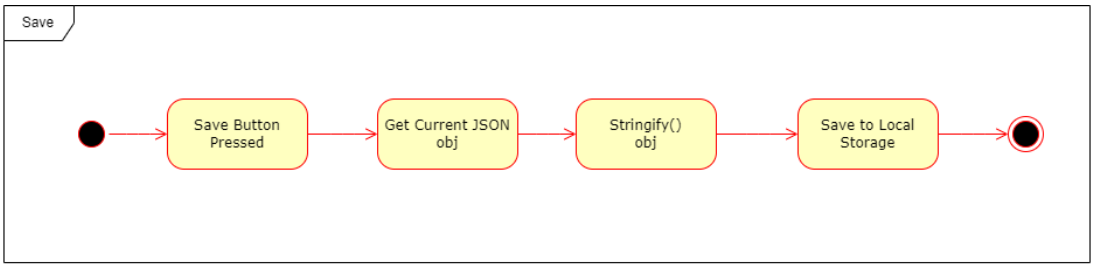
Current - Sequence Diagram (Improved)

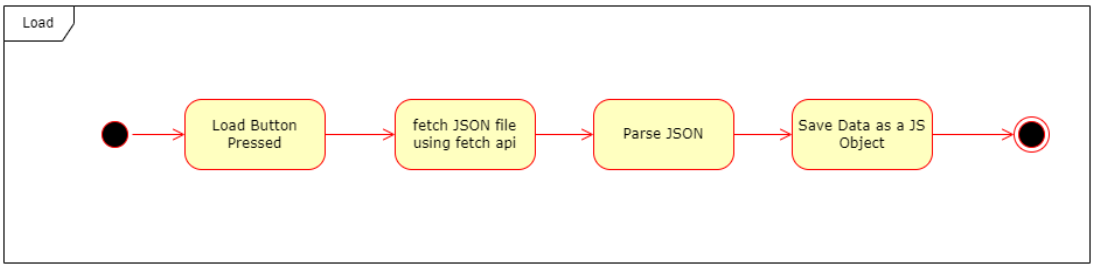


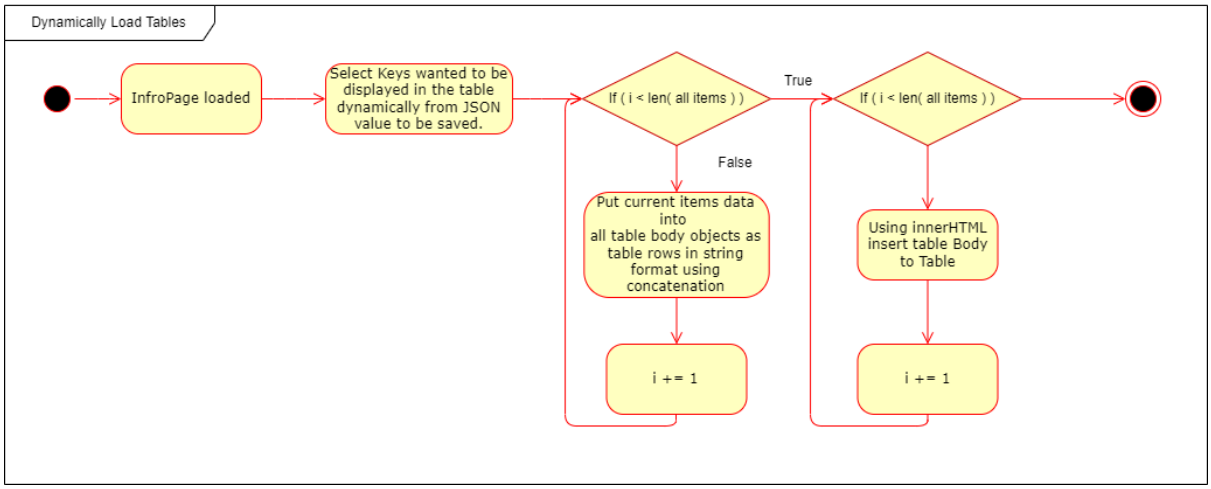
Planned – UML State Diagrams



Current - UML State Diagrams







Overview of UML Planned and Current

Planned UML diagrams are much more complex when we look at the architecture such as class diagrams indicating good planning. However, there are a lot more current diagrams such as the UML state diagrams which are more thought out and clearly represented. These diagrams have been undergoing various iterations of revisions during different sprints where my original diagrams were not thought out enough and simplified to match time constraints in certain areas. Overall, the current UML diagrams are much simpler to understand and match all functional requirements despite removing advanced features such as the statistics class to meet time constraints.

Testing

Test Plan - Test Types (Functional / Usability / Security)

Test Type	Description	Test Steps	Expected Results	Status (Pass/Fail)	Actual Result
Functional	Website should be SPA	Use the navigation bar and check if the URL has '#' and doesn't change	The URL ends with a #	Pass	As Expected (12/01/21) Pages loaded via AJAX correctly when nav bar is pressed
Functional	Navigation bar should work	Go to the website's navigation bar. Click on each link.	Ensure whenever the link is pressed the page is loaded	Pass	As Expected (12/01/21) Pages change as expected by loading in the div correctly
Functional	JSON output to local storage	Must check local storage is empty or different. Check if the JSON is stored into the local storage by seeing if there is a change.	The JSON and local storage is the same	Pass	As Expected (13/04/22)
Functional	The Program must be able to get JSON input from local storage	Must check JSON file is loading the program correctly by parsing the data and displaying it into the console	Website will display JSON format as expected into console when test is executed	Pass	As Expected (13/04/22)
Functional	Read a JSON file using the format specified in Project scenarios to identify how much time is spent on leisure activities	The JSON must output in the format as shown in the test scenario and see if time spent is saved here.	Compare program JSON with Project Scenario	Pass	As Expected (13/04/22) Loaded in correctly with extra fields with core requirement Included
Usability	The website must scale well to phones	Use Chrome Web Dev tools and change screen type to mobile	Website Scales well	Pass	As Expected (12/01/21)

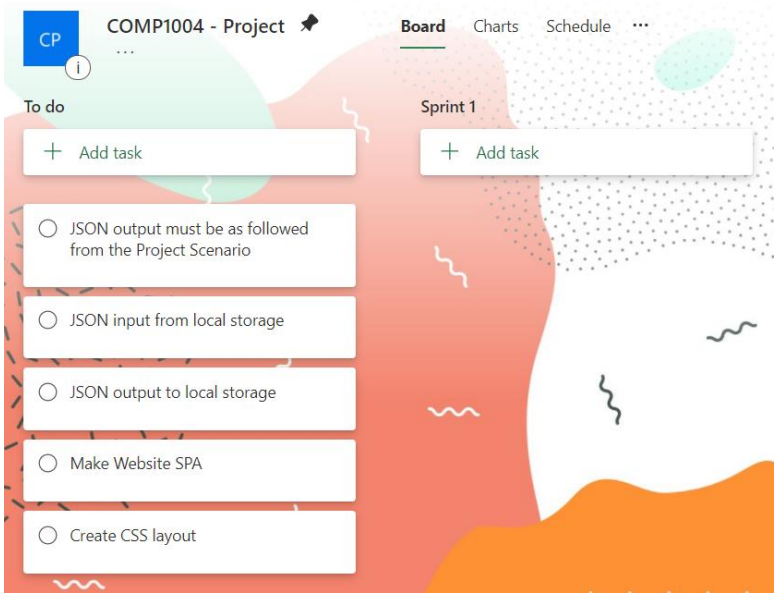
Functional	Lightbulb if ranked good, medium, bad to a corresponding traffic light colour RGB	Enter in actual schedule and expected schedule. Check if the Result is Good/bad/ok then pick the corresponding RGB traffic light	If good/bad/ok light must show corresponding traffic colour	Pass	As Expected (13/04/22)
Functional	Website can save	Press the save button and it will save to JSON then local storage	It will save	Pass	As Expected (13/04/22)
Functional	Website can change colour when Score is calculated correctly	Enter a good, ok and bad time-managed activity into the website	The website should change colour to green if good, amber if ok or red if bad	Pass	As Expected (13/04/22)
Usability	The website should look nice	The user should be able to look at the website and tell the front end looks good	The CSS should look pretty	Pass	As Expected (13/04/22) The CSS gradients make the front-end look good
Functional	The form should be dynamically loaded into the table correctly	The user must load info and see the table dynamically loaded	The table should be present and filled in with activity	Pass	26/04/22
Functional	The Hue lightbulbs should output R/A/G	Person does the form. Looks at the lights in Rm 109 and compares	The Hue lightbulbs should output R/A/G	Pass	26/04/22

Sprint Plans

Sprint 1

20th December 2021

I will make my project backlog containing all my project's functional requirements, I will initially program the front-end of my project using following the box model using a typical div structure for my layout that is scalable to match one of my usability requirements (CSS Box model - GeeksforGeeks, 2022).



Sprint 2

3rd January 2021

I plan to add HTML forms to my website to enable the user to enter their actual schedule and desired schedule and save it as JSON. Then attempt to store the JSON object into local storage to be later operated on and manipulated into calculating the score and level. On another note, since the interim presentation is close that I plan on developing my diagrams and developing my project vision further. I wish to push my project to GitHub during this sprint as I have been struggling to adapt to this workflow.

Sprint 3 – 17th January 2022

In this sprint, I aim to study how the HTTP PUT and POST commands work using Reqbin as recommended on DLE (Online API Testing Tool | Test Your API Online, 2022). I have been having issues adjusting the data. However, I believe that the JSON file that I receive from the Hue lights 'state' must be if the light is on or off and that its 'hue' is its colour and that if I use '/1' in the API's URL I get the lights number. Knowing these key facts, I wish to try to develop the Light Class and get the lights working.

Sprint 4 – 31st January 2022

During this sprint, I wish to refocus my attention on the JSON class and information class as planned. I will manage form input and whenever a button is pressed and create functions in that class to save the JSON into local storage and load it back into our website.

Sprint 5 – 14th February 2022

I plan to resolve this bug with my SPA website by looking at trying to look at different ways to load in the div using JQUERY and looking at the lectures if I made a mistake by any chance. This bug has the potential to completely ruin my project as different divisions need to interact with each other to load correctly such as the Information page.

Sprint 6 – 28th February 2022

During this sprint I plan on looking at various workarounds against my project breaking bug, I think if I add all JSON objects to the Session Storage each time you add it I could transfer the data to other pages as the divs are loaded correctly assuming my SPA is correct.

Sprint 7 – 14th March 2022

In this sprint, I plan on investigating the origins of my bug most likely linked to the SPA implementation of my website. I intend to investigate the structure of my website and brainstorm possible solutions to either work around this bug or resolve the bug directly.

Sprint 8 – 28th March 2022

Despite losing significant time due to COVID-19 affecting my asthma, I wish to follow my plan of investigating the actual structure of my website looking at the Class Structure and major code refactoring trying to use functional and procedural abstraction of the problem.

Sprint 9 – 11th March 2022

Proceeding from the major success of sprint 8, this sprint will look to fix all the bugs present in my program and add a few advanced features as I have now officially matched all minimum requirements making it a minimum viable product. The primary focus of this sprint is to make my product refined as well as show the potential and direction it could have taken if more time was added. I will have to distribute my time management towards other modules whilst focusing on this module due to lost time.

Sprint Reviews

Sprint 1 Review

To a large extent, the initial sprint has been extremely successful in achieving my first core requirement using DLE's resources teaching me SPA using JQUERY; ensuring this website is SPA and making the pages. However, due to time pressure for the interim presentation and submission deadline, I realise that I need to scale back my project and focus on the minimum requirements due to starting late. Hopefully, I may be able to make the lights before the presentation although most likely not.

Sprint 2 Review

This sprint has been somewhat successful but not as planned, initially, I planned on focusing on JSON functionality and ran into troubles. Drawing my attention elsewhere, I focused on the Lights studying the Hue Lights API and how it works. I achieved using an asynchronous HTTP GET Command and received JSON input and turned the data into an object which has been parsed using JS inbuilt JSON functionality. I realise that I will need to change this object in some manner and either POST the data

back or PUT the data back to the lightbulb to turn it on and change the colour. To a large extent, despite not much being achieved I have conceptually progressed massively making this a very significant sprint and close to a breakthrough.

Sprint 3 Review

Sprint 3 has been one of my most successful sprints thus far, I have created the Lights Class that's constructor takes in a string URL and using logic modify which light you would want to use as '/1' will give light 1 and if I use a for loop all 6 lights I can output all 6. Using a HTTP Put command I have successfully output to a Hue Light. Various functions that I have created have been fantastic as the parameters act as an interface to select the colour wanted and which lights you may use. The class created is loosely coupled meaning this can be used independently in different code bases too. However, this did slightly change my class structure the result is an even better one.

Sprint 4 Review

This sprint has been mostly successful as I developed the overall JSON Handling class structure close to plan although with deviations. I am now able to save and load JSON data of the current JSON object. I still need to work on calculations although during testing I noticed that as a change div the data is lost completely. This bug is holding back my entire project despite the major strides taken this sprint.

Sprint 5 Review

This sprint has been extremely unsuccessful as I have attempted many different approaches to solving this issue. I am currently thinking that my issues have arisen due to using JQUERY which is an older JavaScript library causing issues which I don't believe is correct. Most likely the issue is with my implementation of the SPA as further testing indicates that the div reloads everything as you go back to the form the data is lost even if it holds calculating data.

Sprint 6 Review

This sprint has been somewhat successful however not much progress has been made. I believe the Session Storage might be a valid way of transferring data between divs as they seem to act independently from each other as it seems to load a JS module only once when messing with the other divs although using just a normal 'src' to load the JS file as the div is loaded is run every time. The issue arises with the form as it reloads when I load the index Panel again the entire website refreshes losing all data.

Sprint 7 Review

This sprint didn't make much or any progress at all due to being affected by COVID-19 and has affected my well-being very badly with my long-term health condition asthma. The University is aware of my other modules, and I am yet to decide if I wish to claim extenuating circumstances if needed but am considering it. I believe there is much progress in my planned approach. The dates I have been self-isolating under formal NHS guidelines is the 19th – 30th of March causing me to essentially lose one sprint, 2/3 weeks, as I have been affected longer than the minimum isolation period. I have been granted Extenuating Circumstances for this matter. However, I do believe that this product could really become something great and I will personally take strides to meet that.

Sprint 8 Review

To a large extent, this has been a major success as my project plan despite being relatively vastly different the project has made massive momentum. The issue was resolved by removing the indexPanel and not reloading the form at all costs keeping it independent and since each div acts as its own website I realised that my approach to using session storage for each div to communicate live is extremely valid. Local Storage will only be used for storing JSON to meet requirements and not Session Storage.

The Class Structure has changed significantly merging both JSON Handling and Information Classes as they were so strongly coupled in the first place. The Light Class will be used with aggregation with the new improved JSON Handling class. The Form data has been error handled and ensured that the correct format will be used to calculate the level and score.

Level and Score calculations have been added by looping through all items of the array then calculating expected and actual durations then using logic to assign a score to the current iteration object and level is a floor division by 10 to get a level (meeting requirements) as well as outputting the lightbulb. The best part of this sprint is that I managed to make my website follow HCI principles so much better by changing the style of the website to Red, Amber or Green depending on if the person is good, ok or bad. The project's implementation is close to a minimum viable product and in many aspects already is very advanced in certain areas.

Sprint 9 Review (Final)

All minimum requirements have been met with a few extra features such as dynamically loading tables using JavaScript in the DOM to display the current activities and time scores. Additional features such as the screen changing to a radial gradient-based colour depending on how well the user has performed in keeping their time management using advanced CSS as well as JavaScript. As time constraints are running low I think from this point on I am going to focus on the documentation and updating the poster.

Comp 1004 Reflection

The project has successfully achieved the minimum criteria required in the specification whilst adding additional features taking the project further. It uses a simple form which is used by the program to calculate if their time-management is good, ok or bad and display it using Hue Lightbulbs with a traffic light system with further features planned and never implemented such as using statistical methods to suggest user improvements. The project pushed me further to read various academic papers in Human-Computer Interaction relating to colour (Partala and Surakka, 2004) whilst applying it to my project practically by changing the background dynamically and considering use cases.

Initially, the Hue Lightbulb Projects scope had to be scaled back despite the original Application Design Document has been very helpful. The UML diagrams such as the class diagram had too much advanced Object-Oriented Design such as my previous Class diagrams using a lot of aggregation and inheritance. I had to merge the Information and JSON handling classes resulting in increasing the strength of the coupling as it was becoming too complex for me to implement, and over various scrum reviews the layout of the website to fix the bug and consider use cases to always keep the form on-screen. However, due to the features being so well planned implementation has been relatively easy until various bugs slowed progress.

Following Agile, my work output has been exceptional until various bugs had stunted my progress significantly. During my initial sprints, I implemented many of my planned features fast until I hit an extremely significant bug. DIVs in JQUERY are treated as new web pages meaning if you implement JavaScript ES6 classes; JQUERY doesn't like JavaScript being implemented as modules as it thinks the page has not reloaded. Weeks passed trying to fix this bug, holding back the project significantly. Resolving this issue required redesigning wireframes and altering logic by always keeping the form on-screen and using sessionStorage to let DIVs communicate. COVID-19 slowed progress by 3 weeks badly due to my asthma which hospitalised me a few years ago despite managing to catch up.

The Programming implementation is using very good programming practices. The project's code is well commented throughout and is relatively easy to understand. The programming is mirroring the Object-Oriented Designs as shown in my current UML diagrams which is a scaled-back version of my original. However, the variable names used could be improved by using more concise naming rather than variable names too long despite being very consistently named.

To a large extent, the project has turned out extremely well and has been very rewarding as I have been struggling with bugs in the code and my health had been directly affected due to external reasons affecting my pre-existing health conditions. If I could do this project again I would have taken a similar more refined consistent approach and targeted the scope of the project correctly using modern Node JS to avoid issues. In terms of the final product, I am proud it provides visual feedback and even shows dynamic tables to provide feedback to the user.

Github Repo

https://github.com/Plymouth-University/comp1004---main-assignment-Damon-Martin

Microsoft Planner

https://tasks.office.com/live.plymouth.ac.uk/en-GB/Home/Planner/#/plantaskboard?groupId=0966ceef-a52f-405d-95de-0f2f9a7089c8&planId=n2i7udBj_0ymV3GbsTylTJYAGQOr

References

Sprung, J. and Rogers, A., 2020. Work-life balance as a predictor of college student anxiety and depression. *Journal of American College Health*, 69(7), pp.775-782.

Davis, A., Bersoff, E. and Comer, E., 1988. A strategy for comparing alternative software development life cycle models. *IEEE Transactions on Software Engineering*, 14(10), pp.1453-1461.

Tutorialspoint.com. 2022. SDLC - Waterfall Model. [online] Available at: [Accessed 01 May 2022]

Slawek-Polczynska, A. (2020). Is Agile always the best solution for software development projects? [online] SolDevelo Blog. Available at: https://www.soldevelo.com/blog/is-agile-always-the-best-solution-for-software-development-projects/.

Beri, N., 2015. Academic anxiety among secondary school students in relation to time management competency. *International Journal of Physical and Social Sciences*, *5*(9), pp.325-339.

Assets.publishing.service.gov.uk. 2022. Guide to the General Data Protection Regulation (GDPR). [online] Available at: <https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/711097/guide-to-the-general-data-protection-regulation-gdpr-1-0.pdf> [Accessed 01 May 2022]

GeeksforGeeks. 2022. CSS Box model - GeeksforGeeks. [online] Available at: <https://www.geeksforgeeks.org/css-box-model/> [Accessed 13 April 2022].

ReqBin. 2022. Online API Testing Tool | Test Your API Online. [online] Available at: <https://reqbin.com/> [Accessed 13 April 2022].

Partala, T. and Surakka, V., 2004. The effects of affective interventions in human–computer interaction. *Interacting with Computers*, 16(2), pp.295-309.