Progress and engagement report

William Bosley

registration: 100273515

1. Description of project.

On average 6.3% of UK university students drop out of their courses (Frobisher, 2020). There are many causes for student attrition, but a key academic reason is being underprepared; a strategy to counter this is building up students' academic skills (Cuseo, 2011). Sheffield University says that time management is the most important and challenging skill for a student to develop (TUOS, 2021).

'Time on-task' is the amount of time a student spends engaged in learning (Hai-Jew, 2020). The results of an article by Lee (2018) suggest that when longer, more frequent blocks of time are chosen for learning by students, the greater their likelihood of success. According to Cambridge University, students should spend an average of 42 - 46 hours per week studying (Cambridge, 2021). The Open University suggests 32 - 36 hours per week (OU, 2021). However in 2019[1] a study concluded that out of a study sample of 29,784 UK students, 56% spent less than 11 hours per week studying (Neves, 2019).

A paper published in 2019 found there was a correlation between a student's perceived control of time and their academic performance (Adams and Blair, 2019). Computer Science has the highest attrition rate of any bachelors degree course in the UK, with a drop out rate of 9.8% (Frobisher, 2020). With this information in mind, additional time management and organisation for these students seems appropriate.

The aim of this project is to create a web application that will generate a study schedule for university students of all abilities to help them manage their time effectively. Examples of applications where students input their workload and receive a personalised study schedule appeared to be very limited, and my research into this subject supported this. My objectives are:

- Create an algorithm where inputted is academic assignments, their due dates and their total percentage of the grade of their respective university modules, and outputted is a precise and useful schedule which a student could follow.
- Implement this algorithm within an application to allow students an easy way to use this scheduling algorithm and access its output.

[1]Results from a study from 2020 were available, however I chose to use the 2019 information as it would be more representative of normal life as it would not have been affected by COVID-19.

2. Problems addressed by this project.

With the information stated above, that a students perceived control of time has a positive correlation with their academic performance and that Computer Scientists have the highest attrition rate of any course, we can conclude that improving time management would be beneficial. Improving a students time management isn't only impactful because it improves their perceived control of time, as many studies have concluded that regardless of the students perceived control, time management is a strong factor leading to student success (Puzziferro (2008), Britton and Tesser (1991) Basila (2014)).

Wolters (2021) describes self-regulated learning as "students' active control of their cognitive, motivational, and behavioural engagement in learning". It has been concluded that self-regulated learners are academicly successful (Sun and Rueda (2012), Puzziferro (2008)), as well as that good time management is positively associated with increased use of self-regulatory activities, which are integral to being a self-regulated learner.

Students may be keen to succeed, but poor time management is holding them from their full potential. However, an application to help them manage time effectively could bypass this. An application that improves students time management would increase their likelihood for success, because time management is no longer a detriment to their learning ability.

46% of Britons aged between 16 and 24 accessed online learning material in 2020 (Statista, 2020), and Technavio (2021) states that the E-Learning market size is set to grow USD 9.94 billion in the UK from 2020 to 2025. With these, and the "paradigm shift" from traditional to online learning in the wake of the Covid-19 pandemic, (Pokhrel and Chhetri, 2021) which has led Pokhrel and Chhetri (2021) to conclude that online learning should be encouraged even after the pandemic, a digital solution is clearly applicable.

By the end of this project, this application will have solved the problem of students creating unprofessional and amateurish schedules which don't manage their time effectively: this application will create a professional schedule as if it had been created by a personal assistant.

3. Design and planning.

I have created an initial work breakdown structure, which can be found in the appendix, and a Gantt chart showing the initial plan of the project. The work breakdown structure describes the tasks to be completed as the project progresses, and these tasks are represented in the initial plan's Gantt chart, which can be seen in figure 1. Tasks are represented as white boxes, void sections due to holidays are represented using grey boxes, and slack is displayed using black boxes. The tasks on the Gantt chart have been numbered with their associated work breakdown structure number.

I have produced a MoSCoW analysis and an ideal product description, which can be seen in the appendix. I have prioritised requirements relating to the creation of a study schedule, as I believe it is important to focus on the functionality of the scheduling algorithm(s) rather than creating a good looking website with no substance.

I have produced wireframes for most of the look and feel of this app. Figure 3 shows a potential home screen design. The large rectangle in the centre is the user's calendar: an ideal product would display this front and centre upon loading to give users a clear and quick view of their upcoming events. Figure 4 shows a page a user could see when adding a personal commitment to their calendar before adding their work, e.g. a sports club session or a lecture. They can add a date and time for a commitment, and if it is recurring they have the option to choose other weekdays it will occur on, and to choose for how many weeks this event will run for. Figures 5 and 6 show two concepts for a page where multiple commitments can be added at once. I liked the idea of commitments being displayed as stacked cards, so the user can see the titles of all other commitments entered in an aesthetically pleasing way. I believe figure 5 is more promising as it is easier to read the titles of other commitments if they are positioned horizontally. Figures 7 and 8 show a potential screen where a user would add a work assignment / task. The two pictures show the changes made to the screen depending on whether the app or the user calculates the number of hours needed for task completion.

I looked at the options available and chose to use the Google Calendar API. Other options included the 30 boxes API and the Nylas Calendar API. I found that the 30 boxes API documentation was not extensive or easy to use compared to Google's API. Nylas Calendar, on the other hand, seemed too extensive and more aimed towards a corporate market rather than for users to quickly turn their ideas to reality.

I started learning the Google Calendar API by following Google's Calendar API

JavaScript tutorial. I chose to use the JavaScript tutorials because I am very confident in my ability to create websites using HTML and CSS, so picking JavaScript and creating my app as a web application will speed up my work significantly compared to something I am not as good at. I learnt to use the Google Developer Console to allow a test website to log in users with their Google accounts, and after they have given the app permission it will display their upcoming calendar events. Figure 9 shows the website, and figure 10 shows the portion of my calendar that the website was displaying.

Next I created a test website to allow users to add an event to their Google calendar using a simple GUI. Figure 11 shows the screen where users can add an event after the add task button has been pressed. The user is notified below the form that a Google Calendar event has been created to their specification. The number of work hours and minimum work chunk input boxes currently send a string containing their contents to the calendar event's summary. This will be changed in the future so that this information is used to place multiple events into the user's calendar according to these specifications. Figure 12 shows my Google Calendar after I used the website to add a test event.

I also produced an adjusted work plan to account for project slippage during the first semester, which is shown in figure 2.

4. Evaluation of progress.

My literature review gave me the insight that my project would be useful and stand out compared to alternatives. The numerous benefits of time management would be useful for the target audience of this prototype (who have the highest attrition rate of any course in the UK), computer science students. A similar systems analysis helped me conclude that there are no alternatives to this software readily available online or to download: no applications came with automated work scheduling features.

I did have some project slippage during the early stage of my project. A remedial action I took which is still in effect is to seek advice for increasing efficiency at researching and writing, as well as adding in large times allotted to project work to my weekly routine. After taking these actions, drawing my website wireframes and making a MoSCoW analysis went as intended.

The process of choosing a calendar API and becoming familiar with it was extremely valuable to me. During my week where I initially investigated the APIs, I found that the Google Calendar API was the most useful. Although the initial setup using the Google

Developer Console was time consuming, once I was familiar with the process, using their API in my test website was the simplest and best documented compared to the Nylas or the 30 Boxes API. I went on to develop a website that allows users to log in with their Google account, and then create events that will be synced with their Google accounts' calendar. This process taught me a lot of valuable information. I learnt how to link a web app to the Google developer console so that it can have Google log-in authorization and make use of any of Google's APIs. I learnt how to use Google's API documentation to aid me in constructing Google Calendar connected websites. All the programming I did to connect my website to the API was done in JavaScript. This taught me a lot about the language: I felt inexperienced before, but now I feel far more confident having produced these working test websites.

Figure 2 shows my adjusted Gantt chart with a plan that accounts for project slippage. Due to lack of time this semester, I have moved the design of linked events into my Christmas holidays.

References

Adams, R. V. and Blair, E. (2019). Impact of time management behaviors on undergraduate engineering students' performance. *SAGE Open*, 9(1):2158244018824506.

Basila, C. (2014). Good time management and motivation level predict student academic success in college on-line courses. *Int. J. Cyber Behav. Psychol. Learn.*, 4:45–52.

Britton, B. K. and Tesser, A. (1991). Effects of time-management practices on college grades. 83(3):405 – 410.

Cambridge (2021). Teaching and learning.

Cuseo, J. (2011). The "big picture": Key causes of student attrition key components of a comprehensive student retention plan. pages 1–2.

Frobisher, A. (2020). Degree dropouts.

Hai-Jew, S. (2020). *Strategic and Tactical "Focused Time Learning" Design for Online Learning*.

Lee, Y. (2018). Effect of uninterrupted time-on-task on students' success in massive open online courses (moocs). *Computers in Human Behavior*, 86:174–180.

Neves, J. (2019). Uk engagement survey 2019.

OU (2021). Finding time to study.

Pokhrel, S. and Chhetri, R. (2021). A literature review on impact of covid-19 pandemic on teaching and learning. *Higher Education for the Future*, 8(1):133–141.

Puzziferro, M. (2008). Online technologies self-efficacy and self-regulated learning as predictors of final grade and satisfaction in college-level online courses. *American Journal of Distance Education*, 22(2):72–89.

Statista (2020). Share of individuals using online learning materials and doing online courses in great britain in 2020, by age and gender.

Sun, J. C.-Y. and Rueda, R. (2012). Situational interest, computer self-efficacy and self-regulation: Their impact on student engagement in distance education. *British Journal of Educational Technology*, 43(2):191–204.

Technavio (2021). E-learning market in uk by product and end-user - forecast and analysis 2021-2025.

TUOS (2021). Time management.

Wolters, Christopher A. Brady, A. C. (2021). College students' time management: a self-regulated learning perspective. 33:1319 – 1351.

A. Requirements: MoSCoW analysis.

MoSCoW analysis.

Ideal product description:

The ideal product would be an application that allowed users to quickly add all their usual commitments to a calendar, either manually or by syncing with a calendar application or by importing an iCalendar file. Then, they can quickly upload all their assignments and work. The application would then use the weightings and deadlines of each task, alongside constraints set by the user (e.g. no work between 6pm and 9am) to populate their calendar accordingly. Urgent and time-consuming tasks would receive priority, whilst also scheduling regular instalments of other work so that progress is still made on other upcoming assignments. The work schedule generates around the users' prior commitments. If a new assignment is added to the schedule after the initial setup, the upcoming schedule should rearrange accordingly.

The users upcoming schedule would be displayed on an aesthetically pleasing, easy to understand calendar. The user could be able to select between viewing a day, a week, or a month of the schedule. The user can colour coordinate calendar items. If a user has not completed a work chunk, they should be able to select the incomplete chunk and the application will add it into their schedule in the future. The system will alert user to upcoming deadlines. The user can also add commitments by click and dragging to mark an area of the calendar to be taken up by this commitment. If they wanted to sync this calendar with another application, they could sync it with their Google or Outlook accounts, or export the calendar as an iCalendar file.

Must have:

Interface to allow user to add commitments to a schedule.

Users can edit commitments which are on the schedule

Ability to calculate the number of work hours / week that a piece of work will need.

Ability to place these work hours onto a calendar.

Ability for calendar constraints to be set. E.g. no work after 9pm.

Ability to apply items to a calendar within according to these constraints.

Should have:

Functionality for allowing the storing of a schedule.

Ability to adjust length of work chunk depending on the activity.

Ability to adjust placement of work hours based on priority.

Ability to add multiple commitments at once

Could have:

Account system / login system.

Easy to read user interface.

Ability to link assignments to modules/classes.

Mobile responsive webpage.

Outlook / google calendar connectivity.

Outlook / google calendar syncing.

Accessibility Settings e.g high contrast mode.

Screen for first use allowing a quick setup.

Ability for user to colour coordinate work chunks.

Ability to display a day, a week or a month of a schedule.

Would have:

Ability to import/export to iCalender Format.

Ability to select area on calendar to add/remove work chunks.

Ability to sync with non-Gregorian calendars for religious festivals / holidays.

B. Work Breakdown Structure

Work breakdown structure:

- 1: Project proposal + Literature search
  - 1.1: Project proposal - Produce a document where I will define the problems addressed in my project, identify risks in producing this project and ways to mitigate them, and identifying resources and areas of literature that are likely to be required.
  - 1.2: Literature search - Search through the relevant areas of literature described in the project proposal, and create a library of resources that will be useful for describing the key themes and problems addressed by this project.
- 2: Literature Review - Use the resources found in the literature search to produce a document that describes the project; it's themes; and the problems addressed by it, as well as a critical review of the relevant literature.
- 3: Wireframes + MoSCoW
  - 3.1: Wireframes - produce lo-fi mockup images that show what different screens of the application could look like.
  - 3.2: MoSCoW - Produce a MoSCoW analysis: a list of requirements, categorised into four sub lists of features the application *M*ust have, *S*hould have, *W*ould have and *C*ould have.
- 4: Choose API - Choose an API that will be used to implement the calendar in the application.
- 5: Add cal. event with API
  - 5.1: Build a test website that connects to a user account that has a calendar associated with it.
  - 5.2: Add functionality to this test website so that a custom event can be created and added to the user's Calendar from within the test website.
- 6: Add linked events I.E. tasks - Define a way of linking multiple calendar events together so that a list of calendar events can be recognised by the app as work chunks that contribute to the completion of a task.

- 7: Progress Report - Produce a progress and engagement report which displays my completed formative work, which evaluates my progress so far and which identifies project slippage.
- 8: App can add tasks and output events - Add functionality to the app so that a user can add an academic task and the app will schedule appropriate work chunks into the user's calendar.
- 9: System can schedule lists of tasks - Add functionality to the app so that a user can add a list of academic tasks all at once, making the app schedule them all in to the user's calendar.
- 10: Simple GUI + user can manage progress - Add a simple GUI and the functionality to adjust the number of work calendar events scheduled depending on the user's progress on the task since they created it in the application.
- 11: User can manage other calendar events - add functionality for users to manage non academic calendar events within the application so that they can plan their work schedule around their life.
- 12: Final report writing - produce a project portfolio which details the context, design documentation, outcome, evaluation of what has been produced, and a conclusion.
- 13: Inspection preparation - prepare for presentation of project which will be a slide-show explaining and analysing the produced application.

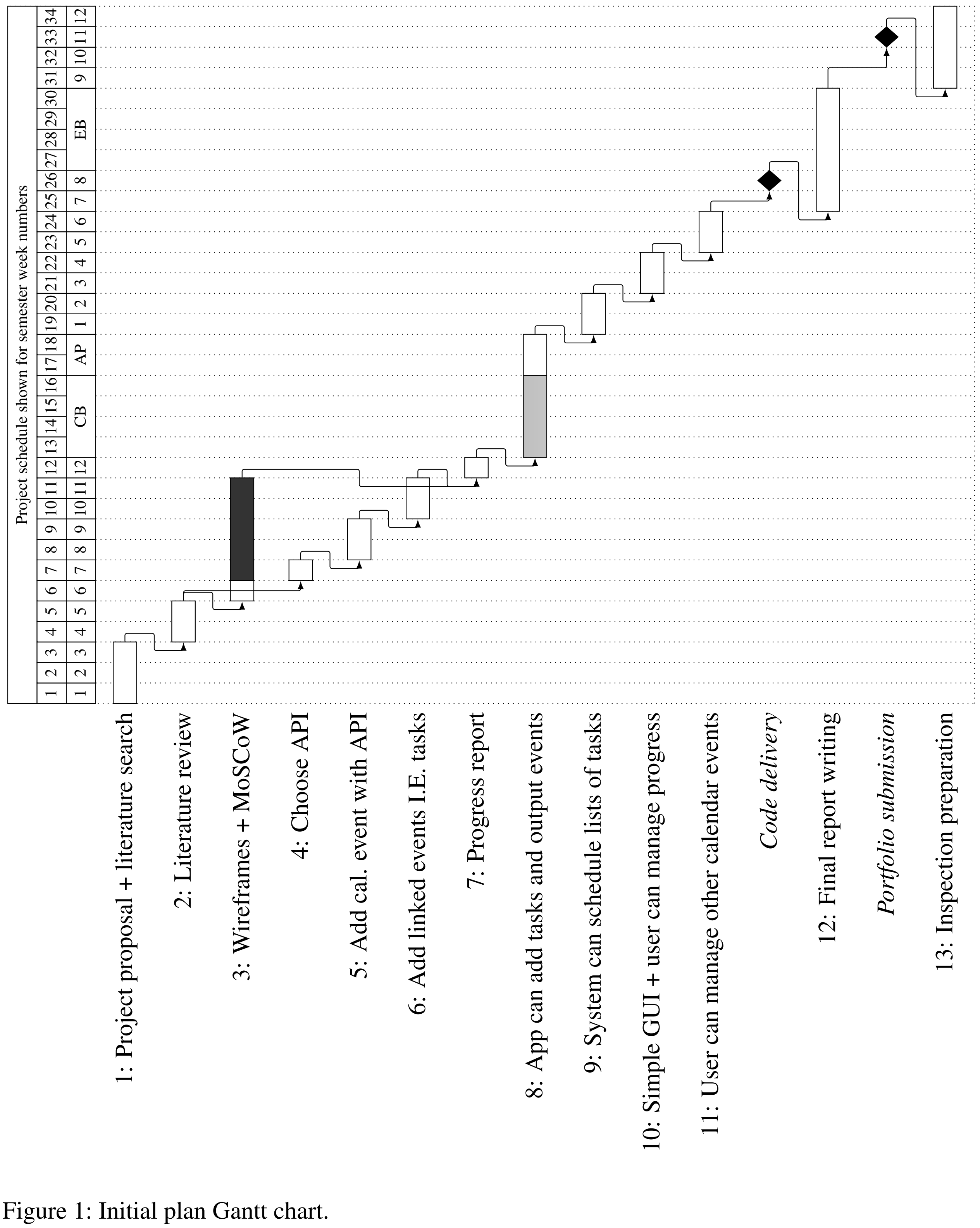


Figure 1: Initial plan Gantt chart.

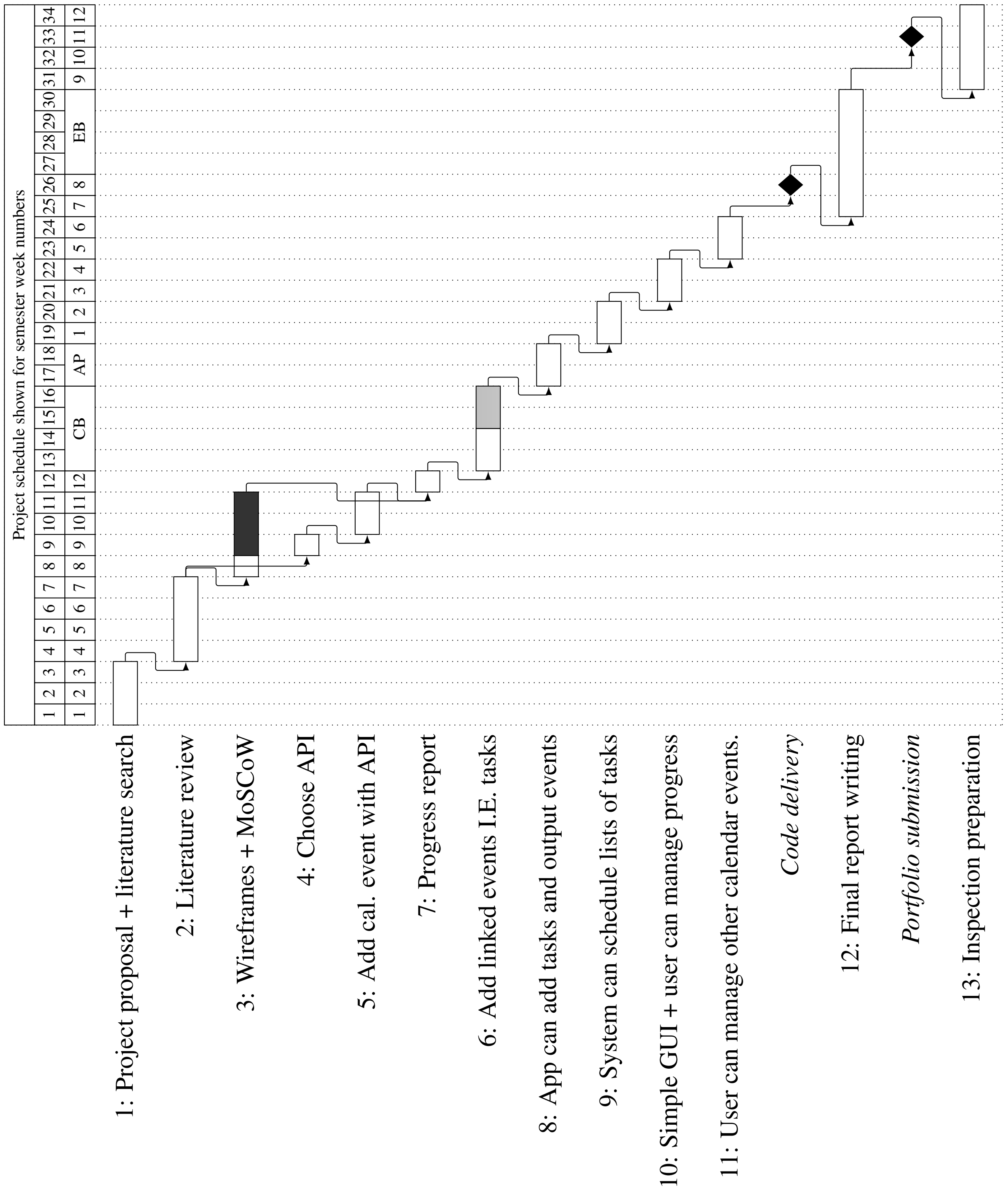


Figure 2: Adjusted Gantt chart after project slippage.

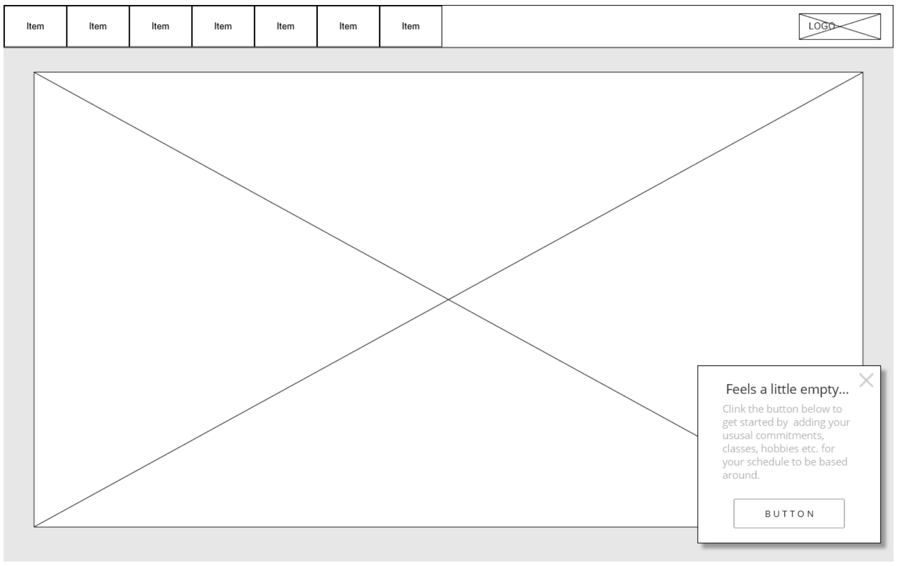

Item
Item
Item
Item
Item
Item
Item
LOGO
Feels a little empty...
Clink the button below to get started by adding your usual commitments, classes, hobbies etc. for your schedule to be based around.
BUTTON

Figure 3: Home page wireframe, with toast message which appears if calendar is empty.

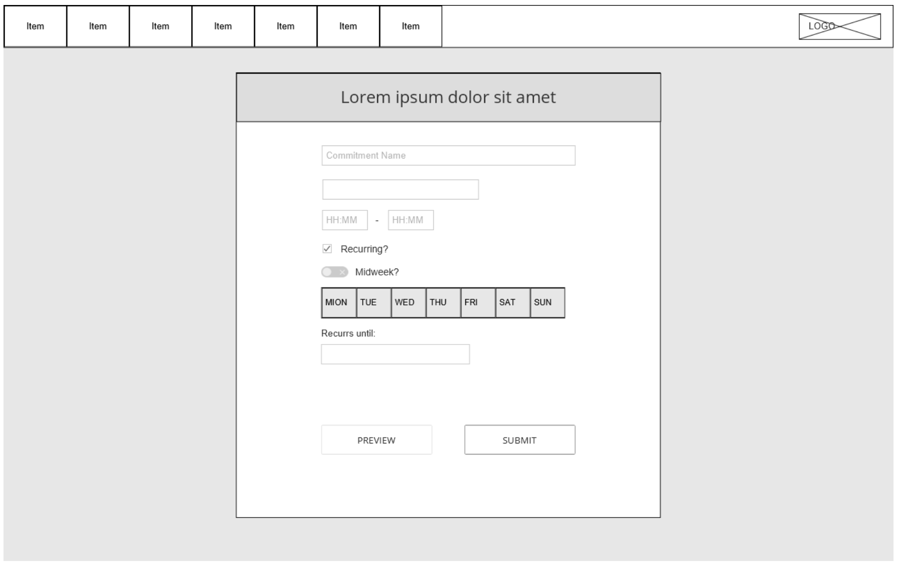


Figure 4: A wireframe for the page to add a commitment.

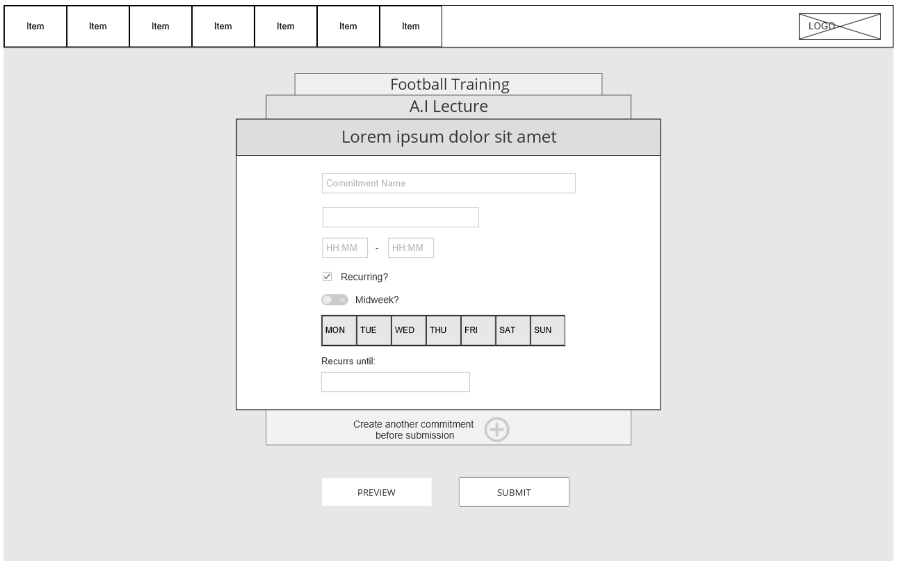


Figure 5: A wireframe for a page for adding multiple commitments.

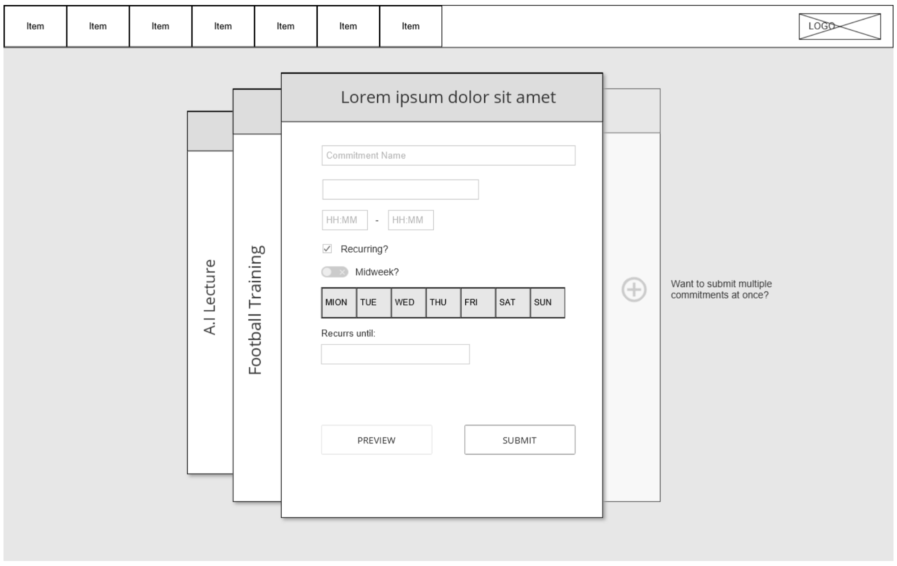


Figure 6: An altenative wireframe for the page to add a commitment.

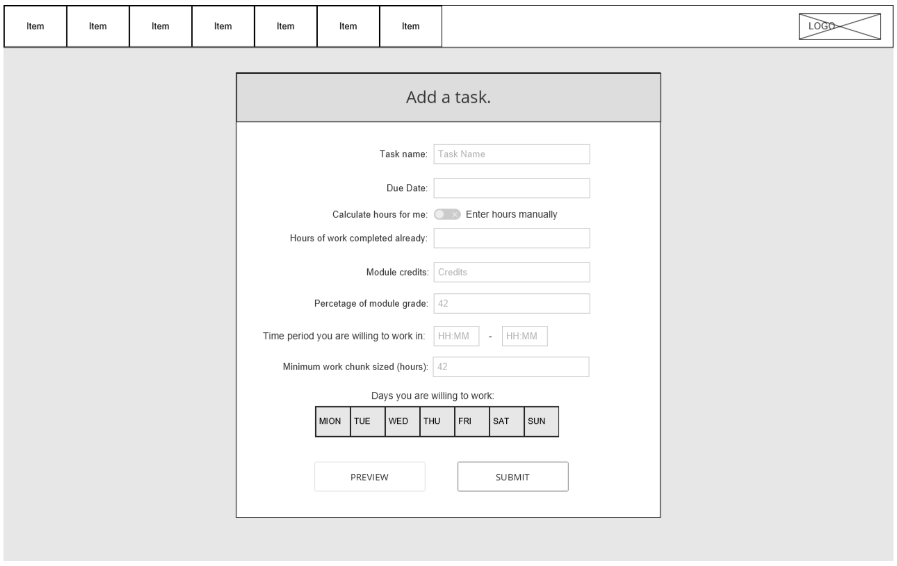


Figure 7: Wireframe for adding a task - computer calculating hours.

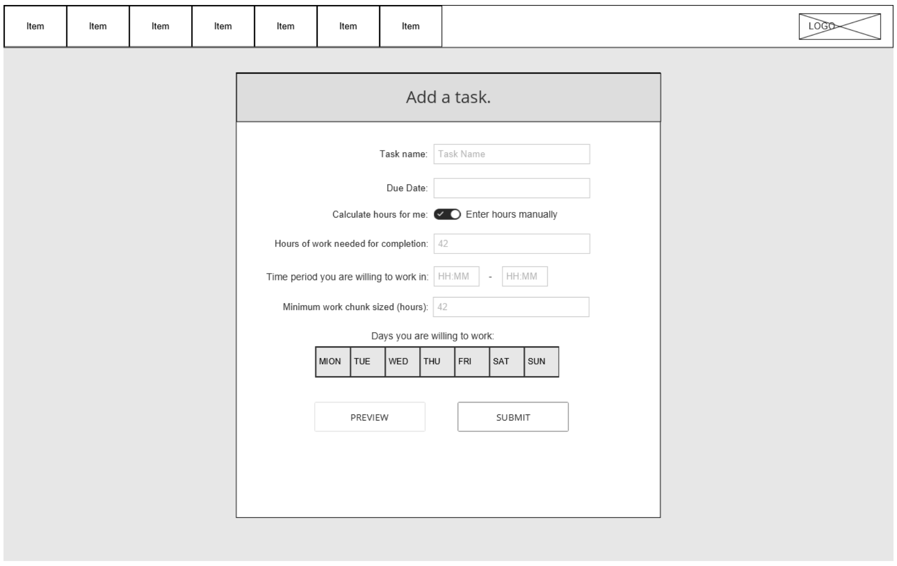
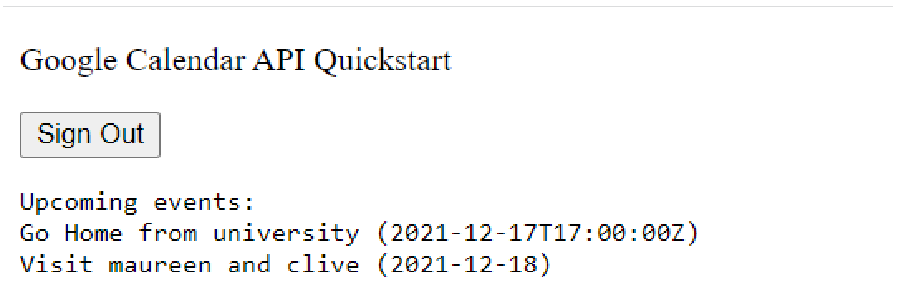


Figure 8: Wireframe for adding a task - user calculating hours.



```
Upcoming events:
Go Home from university (2021-12-17T17:00:00Z)
Visit maureen and clive (2021-12-18)
```

Figure 9: Google calendar connectivity test application made using a Google quickstart.

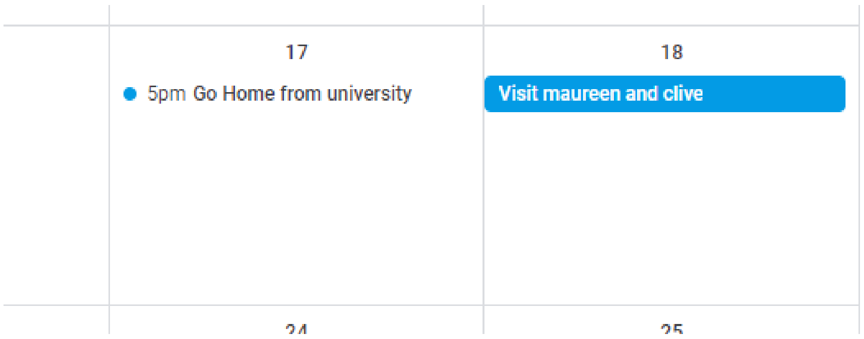
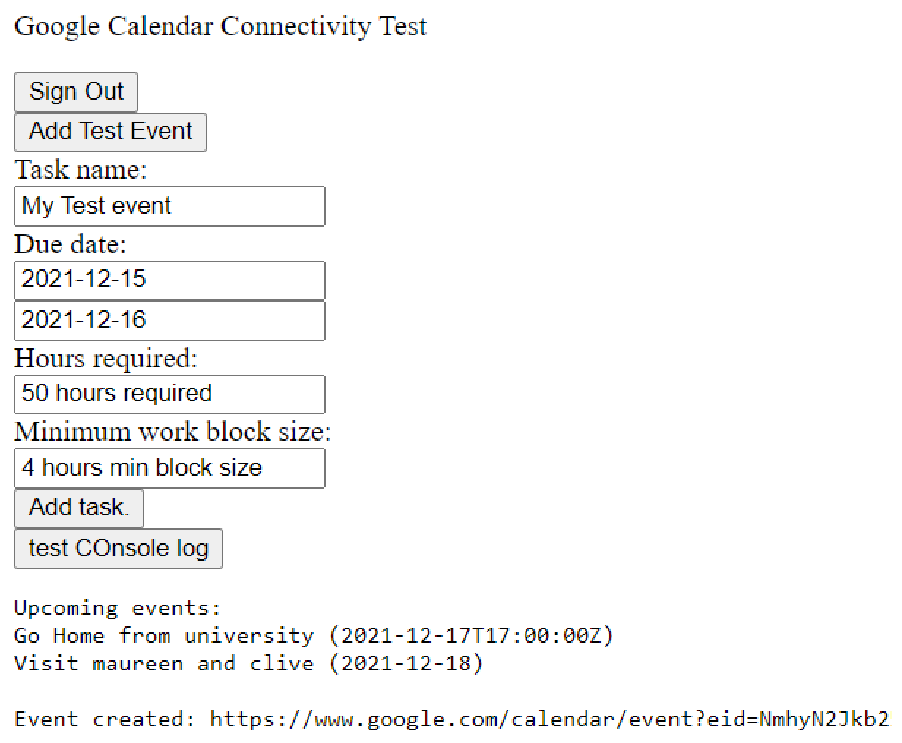


Figure 10: The upcoming events in my google calendar which were being displayed in the quickstart.



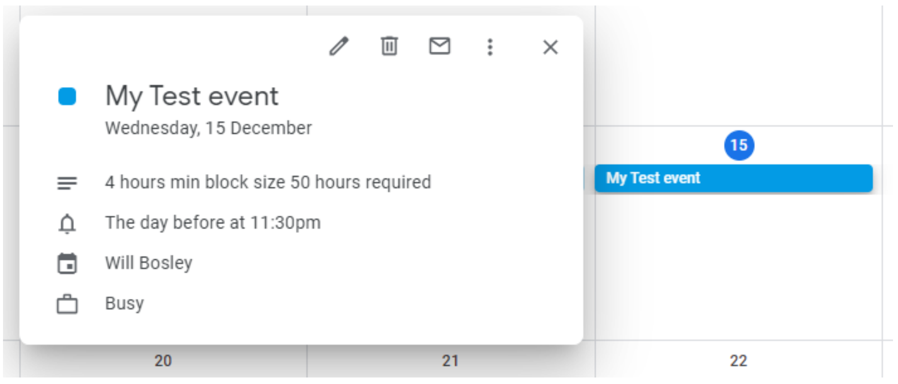
Google Calendar Connectivity Test

Task name:
My Test event
Due date:
2021-12-15
2021-12-16
Hours required:
50 hours required
Minimum work block size:
4 hours min block size

```
Upcoming events:
Go Home from university (2021-12-17T17:00:00Z)
Visit maureen and clive (2021-12-18)

Event created: https://www.google.com/calendar/event?eid=NmhyN2Jkb2
```

Figure 11: The test website for adding an event to a user's Google Calendar.



My Test event
Wednesday, 15 December
4 hours min block size 50 hours required
The day before at 11:30pm
Will Bosley
Busy

Figure 12: My updated Google calendar after using figure 11.