

## Individual Report

### Lessons Learnt

Regarding the challenges and opportunities of collaborative work, this module built upon my previous experience within university. That is, I was able to take the opportunity to utilise my learnings from previous modules, for example taking a more technical role than in CSC8611, while also being presented with new challenges such as physical prototyping in a group rather than individually. This, in particular, provided various lessons, namely how one cannot be involved in every process and must expect to then pick up roles where you don't have an extensive knowledge of prior work. For example, I was put in charge of the circuitry despite having little knowledge/involvement in the physical structure, which meant I had to learn to analyse prior work done by others and communicate with them to derive requirements for the circuitry which could then be communicated to Ben from the MakerSpace for collaborative work on this.

Another lesson I learnt was related to my individual working in development; broadly, I had to learn javascript, as well as how to use nodejs, express and socketIO, and HTML and CSS to a lesser degree. Learning such a quantity of technical skills which I had covered little prior was a difficult task, and I think I could have done so more efficiently by taking an afternoon to watch brief tutorials on each of these before beginning development, as less trial-and-error would have been required. Still, I am happy to have gained these skills.

### Role within Group

Within the group, some roles were taken by nearly every group member, whereas others were split to ensure efficiency. For the former, we were all involved in the ideation process, discussing how to best implement our idea, and writing the final documentation and demonstration. My exclusive role was largely a backend developer, for which I worked alone finalising how to program our implementation, coding server.mjs with the aforementioned tools and assessing the best way of storing the necessary data. That said, a large portion of my time was also spent on the frontend due to the frontend designer coming from a UX background where they did not have the technical skill to work in javascript, which I used to dynamically label events in the user view and make that page responsive to GPIO button input. Finally, I was also in charge of circuitry, where I worked out how best to structure the components and wiring so that they would fit within our physical artifact properly and aesthetically, whilst considering the limitations of the Pi without a GPIO expander. This could not have been done without aid from Ben, who completed the soldering for me.

## **Technical Contribution**

In server.mjs, which I was solely responsible for, various functionality were implemented. Firstly, an express server was set up which could deal with the backend processes like intermittently checking events and listening for GPIO input, which had to be mapped to the correct numbers in the javascript library. Secondly, routing was used to allow for http requests to the addresses used for the user views of events and the booking form, passing in the events currently stored on the server as read from the json file and updated throughout runtime. Thirdly, many functions were used to achieve the aforementioned backend processes. For example, setDateTimes filled an array with the dates and times of all events, which could be used to determine which LEDs should be active at a given time. Finally, SocketIO was used to interface between the server and the client – namely, emitting messages from the server when a watched GPIO voltage changed to indicate a button being pressed.

On the frontend, a .ejs format was used iterate through events as passed and display their html, with custom values for name, date etc. Furthermore, <script> tags were used to write more complex javascript as required by the SocketIO messaging, allowing for refreshing the page for dynamic event updates, and allowing the page to be responsive to GPIO input as if one was using a mouse and keyboard.

Technical aspects of the circuitry also included considering resistors and grounding for components, well as assessing and setting the default values for components in the VSCode terminal while SSH into the Pi to ensure correct functionality.

The extent of my technical contribution is not limited to the above, as it included numerous other minor functionalities and considerations. As well as this, much of my time was spent debugging to ensure the system functioned correctly during demonstration.

## **Challenges and Learning Reflection**

During the group project, and module many challenges were faced, the learning from which can be applied to future projects. Generally, in future I would hope to have better communication within a group, since there were many times I felt “in the dark” about others’ progress in their work. For example, on the day of the PowerPoint being due, I expected the physical prototype to have already been largely built and just require my input of the Pi, but it was mostly incomplete, which left little time to complete many tasks and caused stress for the entire group. In future, I would like to alleviate this kind of problem by regularly checking in with those completing other tasks to mine and encouraging faster completion where necessary.

Jacob Peel 210077111 Group 2

Another challenge was in taking on the entirety of backend and majority of frontend development alone. This was an issue since I was expecting these to be my only jobs, but was later given circuitry in addition, causing stress. In future, I would like to split my workload with others where possible, so that I can be more responsive to challenges that may arise instead of having to dedicate all my time to one task.