Final Report

Group M – Campus Map Navigator

By Joel Mayner, Marcellino Samuels, Thamir Suwar, and Behzad Karimizadehbehbahani

# Introduction

The Richard Hoggart Building (abbr. RHB) is for all intents and purposes the face of Goldsmith’s University. Due to its winding halls and complicated upper floor layout, many who visit the building get lost at some point or another. RHB was built in 1843, long before it became a University of London, this implies that people have been getting lost in the building for over 170 years.

A collective of programmers and designers referred to as Group M decided to address this age-old problem. But for the solution to be effective, it had to be easily accessible, simple to use, and available to as many people as possible at the time they’d need it the most. However, before further development could be done. Research had to be carried out to prove the perceived demand for a solution.

It was discovered through a survey conducted that the number of people getting lost in RHB was high, and so was the demand for a solution. The solutions that did exist were considered lacklustre or non-existent beyond a few sporadically placed maps. Many people- mostly students, were still getting lost and there was a substantial gap in the market where a software based solution could be.

As most students surveyed were android smartphone users, the group decided to design their potential solution as an android application, and as they were learning to code using java – a key programming language in android development, the decision had almost been made for them.

We set out to form an easy to use interface, a fast and reliable pathfinding software, displaying a highlighted path from point A to point B. The group accomplished this and furthermore could implement A\* algorithm to increase the efficiency, we have also added a list of rooms and a pin drop feature to our application allowing users to navigate using their popular methods. Adding an interactive list of rooms to choose from, eliminated the issue of small phones or large fingers interfering with the button design and hindering the design useless.

Ambitions of the group regarding the application were to have the following implemented:

## Additional building maps

* Live Updates of room availability and events.
* Student timetable implementation
* Lectures and society logins to book rooms
* Indoor Positioning System(IOPS) implementation
* iPhone compatibility
* Text and email alert regarding upcoming events or lectures to users

Although we were not able to complete any of these ambitions we are proud of our achievements and our clean implementation of the solution as well as good communications to keep ourselves always in touch and connected with one another. Using Github and Whatsapp messaging to stay in touch and manage the project, we were able to keep updated on each other's progress. While the lack of online logins have restricted the features available on the map it also allows us to comply with the data protection act of 1998 ( we do not store any personal data nor share it with any other party as users are not required to enter any personal information on the application).

In this report, we are going to explore the final build: its features, resources and the research conducted to test the Alpha and Beta version of this software. We will justify our current design and features while exploring the problems we faced while developing this software.

# Development Post-Mortem:

In the past months, we have been able to have regular meetings every two weeks as well as keeping in touch online weekly; this helped a lot as members were at times ill or unreachable. Although we did not meet every deadline we have, we were able to put together a comprehensive effort considering our small size and the illness of the group leader.

We started by doing a risk management and creating a backlog of the features we needed for our project as well as defining our rules. We had much research done on the ideas and the possible designs of the application before starting the initial design process.

The timeline for the commencement of the coding process was by 28th of January and met earlier than planned; we were able to have focus groups and complete research about the optimal requirements of our design. We found out that a lot of the initial ideas we had were not user-friendly however fancy.

We had to decide between the different development methods, and we decided to use Agile Scrum with a few changes (forced and voluntarily). We started by having regular meetings every week online or in person (mostly online), exchanging ideas and codes.

By making adjustments to our plans and final requirements as well as creating a monthly backlog to allow us to start the design process while thinking about the most fundamental needs and aims of our project.

Our monthly backlogs began with a simple requirement to display a version of the university campus map on a computer device which would allow users to see the campus without having to acquire a campus map; achieved by the end of February 2017.

We continued using Scrum, pair programming and collective ownership all allowing us to deal with the lack of resources that our group specifically was facing ( lack of time considering other modules, illnesses and personal reason absences of team members, smaller group size ). The main two programmers on the team worked together to enable fast coding and problem solving; the debugging process also became much quicker. Suggestions and corrections of codes were easy to add as all group members had access to the code in real time on GitHub and were able to manipulate or suggest edits to certain parts of the coding. For the second month starting the end of February was to be able to add features such as navigation, simple Pathfinder. However, we faced several issues with this implementation regarding adding nodes to the map, and the scale of the project became much clearer. The work needed to be completed, and so it was accomplished. Pathfinding algorithm was implanted with minor complication, for instance, the Pathfinder was not able to understand the obstacles it was facing and seemed to suggest to the user to walk through walls and classrooms to arrive at their destination. So it was reviewed and talked about by the whole team, and we agreed that adding more nodes and choosing A\* algorithm plus defining obstacles for the algorithm was the best way forward.

After we had been able to fix this issue, we moved onto our next goal with a bit of a delay; we started by working on our broken navigation system, our second attention was to start creating a user interface with the smallest detail possible. No buttons designed, and the only available option to the user was to choose their current location and their destination from a list of room numbers. We had a simple working system by being able to implement the above.

The product owner(also the scrum master) continued by defining the next set of goals and prioritising them in the order needed. We had to develop the final features of a drop pin buttons and options into our design as well as being able to implement the idea into a mobile operating software. Unfortunately, the latter goal of implementation not met for a set of reasons which will be explained further in this document. While continuing to achieve our goals it became clearer for us that it was not possible to meet the deadline of 27th of April while keeping to our final requirements, this is when more time was required, and the Department granted the request.

As you can see in diagram 1( attached as a pdf file), our first Gantt chart and plan for the coding were most similar to the waterfall development. However, as we continued researching, we realised that we are not able to walk through the steps of the waterfall method of development because of the small size and the limited time would not have been able to review the code in time for release.

Perhaps the biggest issue with that method was the realisation that we could not change our goals until the application is ready which would cost a lot more resources, leading us to develop a new Gantt chart and going by the Scrum method.

Here are some of our planning and original for the development phase explained further:

Programming ground work- Was meant to start off the programming stage by choosing our programming language, Operating Software that it is to be run on and also start by finding a pathfinding algorithm that would help us along our way. This primary development was meant to last for at least a month

Core, Build- Started the implementation process to get the earliest version of our application running by implementing it on a mobile device, testing and debugging. This stage was set to end at minimum 25 days.

Addendum Build- The final stage of our progress which would have implemented the extra features that the user had requested, it was set to last for a smallest of 18 days and include implementation of additional building maps, Live Room Availability and event.

After this, the group would have had to start on writing a report. This very lengthy process did not give us enough time for any changes that we might have needed to make during the process, and it did not allow for the user to change their mind on the design required or any other aspects of the interface.

As it can identify from above, there was no time specified for changes made to our plan, and if any of the developers ran into a problem, they were not able to work with each other or ask for help on others. Due to tasks were designed to each person and was expected to be carried out without the need for a meeting to take place.

There was a vast documentation involved with that method that required to take notes as we go along and document every meeting which was time-consuming and inefficient.

If any issues arose during any of the stages, there was no resources assigned to deal with them and all the reviews and debugging of the codes was happening at the end of each cycle. Which could have caused massive issues during the development process as the coders were not aware of each other’s works, and any problems in the code would have left to the last stage for debugging.

Furthermore, the original schedule restricted the hands of the members regarding collaborating with each other on the work done.

By using Scrum, we created a flexible environment for our developers and allowed them to work in pairs and continuously debug the system. Daily meetings were organised online and weekly meetings to determine our progress and discuss the issues we are having were also held.

Every month we sat down and discussed the current estate of the application and if the goal to have a releasable version has met or not. The Scrum Master continued to write backlogs for every new Scrum cycle.

As can be seen in the above diagram we ran over some of our deadlines especially in the third scope and the second during the development process. In the first sprint, we had issues regarding sourcing the map and the process of coding it. By working collectively on this matter and assigning more time to the development, it was resolved. Furthermore, during the second Sprint, we encountered issues regarding the pathfinding algorithm and the nodding process which took longer than initially thought and required the whole team’s efforts to add all the nodes to every room in the main building. In the Third Scrum cycle we were approached by two issues one regarding the implementation onto Android operating software and second regarding the design’s compatibility with Android and if it would be still usable on smaller or larger screens.

General backlogs:

Small ( core) features: As the user, I want a map which I can use to view the campus

As the user I want the app to be available on a computer device

As the user I want the map to be clearly marked for rooms and amenities

Medium( Core plus) features:

As the user I want the app to have a navigation tool allowing me to find rooms using a list

As the user, I would like to be able to use a search feature to navigate through the campus

As the user, I would like to have a drop pin feature to make navigation easier

AS the user I would like to be able to see a path going from my current location to the destination assigned

As the user, I would like to be able to change the colours

As the users, I would like to be able to see alternative entries

Large (Addendum Build) Features:

AS the user I would like to be able to see live events happening around me as I am navigating or accessing the map

AS the user I would like to check the availability and schedule of rooms and labs in the campus via the application

AS the user I would like to be able to access this application via a mobile device

AS the user I would like to be able to add my current university timetable to the application and be reminded of the lectures and labs coming up

As the user, I would like to be able to navigate between different building and have access to the map of the streets adjacent to University

As the user, I would like to be able to talk to other users online

As the user, I would like to be able to Book rooms and library study areas using the application

Scrum cycle one backlogs:

As the user, I want a map which I can use to view the campus

As the user I want the app to be available on a computer device

As the user I want the map to be clearly marked for rooms and amenities

Scrum cycle two backlogs:

As the user I want the app to have a navigation tool allowing me to find rooms using a list

As the user, I would like to be able to use a search feature to navigate through the campus

As the user, I would like to have a drop pin feature to make navigation easier

AS the user I would like to be able to see a path going from my current location to the destination assigned

Scrum cycle three backlogs:

AS the user I would like to be able to access this application via a mobile device

As the user, I would like to be able to navigate between different building and have access to the map of the streets adjacent to University

AS the user I would like the symbols to be clear specially marking the accessible entries to the building

AS the user I would like to be able to change the colours of the pathway

As the user, I would like to be able to zoom onto the map

A visual representation of the time distribution:

As it can be seen from the above two diagrams, time was managed carefully giving the most amount of time to development and the coding process rather than spending much time on documentation the team was allowed to communicate without having to record every communication. If an issue arose surrounding a specific problem, team members were advised to only contact those who are relevant unless the member is not available.

Pair programming and collective ownership helped us a long way in being able to achieve some of our primary goals.

As can be seen in our schedule, we held daily, weekly and monthly scrum meetings.

Our daily scrum meeting was an opportunity for problems to be mentioned and taken a note of, a review of the progress was held on the daily basis during the scrum meeting. The meeting was flexible and allowed members to be absent if they are making progress on the work being done. If members required the help of another member, they were authorised to use the daily meeting to resolve an issue.

The weekly meetings were held to review everybody’s progress and provide solutions to the problems raised during the daily scrum meetings. The bigger issues that required a group work were either resolved at the weekly meeting or before that.

Our monthly scrum review meeting was to assess the outcome of the process and to suggest further development in areas needed. The monthly scrum meeting was also used to help debug any issues arising from within the application. The new areas for improvement as suggested by the user research would have introduced to the team by the project leader and scrum master.

Obstacles faced during the process:

As explained before one of the major barriers that we had to come over was the unusually small size of the group that we had. The workload was too high on the members, balancing the group work and other university coursework was a real issue. With members having to take up large chunks of work, specialisation fell, and it became harder to have members focused on the details. The leadership became a troublesome task as the team leader had to take on the role of the scrum master and the development team had to work simultaneously on design, documentation, debugging, UI programming, Map programming, algorithm implementation and much more. Keeping track of meeting details and minutes was also a task that was abandoned due to it being very time-consuming.

To resolve this issue, we should have had more timely managed meetings regularly requiring all members to attend instead of the flexible meetings that were held. Communication would have been improved if the role of the scrum master and the team leader were kept separate, additional research would have reduced the workload by making more fundamental goals and lessen the number of requirements. Other ways of tackling this issue would have been choosing a lighter project which did not require extensive mapping and nodding processes. The research during the design analysis stage should have included research into different pathfinding algorithms before the coding started which would have saved some time. The resources were managed to the satisfactory level during this project even though the lack of time, team members and absences of team members were major factors in us running over time.

Furthermore, the group leader and the scrum master had a medicine earlier in February that affected his ability to attend university; this increased the workload even more between the team members.