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 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 means 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 was carried out to prove the perceived demand for a solution.

Through a survey conducted by the group, the number of people getting lost in RHB was high, and so too was the demand for a solution. Those 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 (by a sizeable margin), the group decided to design their potential solution as an Android application to cater to as many people as possible, and as they were learning to code using Java – a key programming language in Android development, the decision was almost made for them.

The app itself would consist of a rudimentary A\* algorithm dressed in a simple graphical user interface. The interface would be designed to allow the user to either navigate to and from anywhere in the building, or to simply view the map, using as few steps as possible.

The features of the application were divided into two main builds, the core build and the addendum build. The core consisted of what was determined to be both vital to the app and feasible for the team within the time limit, and the addendum build contained all features desired to enhance the experience.

## Addendum

* 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 could keep updated on each other's progress. While the lack of online logins has 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.

# Design

**Aim**

The priority of the project was to help users navigate the Richard Hoggart Building itself, which was the primary feature of what would be known as the project’s “core build”, with the rest of the campus as a lower priority. As Eduroam can be erratic at times depending on the user’s location, it was decided to allow the app to function offline. Also considering time constraints and other logistical limitations, the app would be developed exclusively for Android devices using the android studio, this would compliment the group’s course’s focus on the Java programming language and decrease the amount of work needed. A rudimentary A\* algorithm would be used to navigate between the campus rooms. Once the basics were in place, accessibility had come up as a legitimate issue for navigating the campus: When it came to programming, the application had to account for those who had difficulty using stairs, so a function for filtering out stairs and emphasising lifts and ramps was coded in.

All the user would have to do is find and select which room they were in, and which room they wanted to go to and toggle the accessibility filter. The app will display a route between rooms, displayed with a sequence of lines guiding the user from one room to another.

**Alterations**

Very little changed in the time between developing the core idea to the proposed implementation. But of the few changes, a prominent one was the decision to add a feature to view and look around the map freely. The research carried out had the effect of justifying most of the decisions made during the development phase.

The biggest major change to the app was the implementation of Fire exit access, which required adding another filter option in addition to accessibility.

**Core**

RHB –

The RHB building can be intimidating for new students, visitors and newcomers, its corridors are unintuitive, and its routes are remembered rather than figured out, more than any other building on campus, it is the Richard Hoggart Building where people got lost. Although the ground floor is a straightforward location with relatively few trouble spots, the upper floors were where navigation would get confusing.

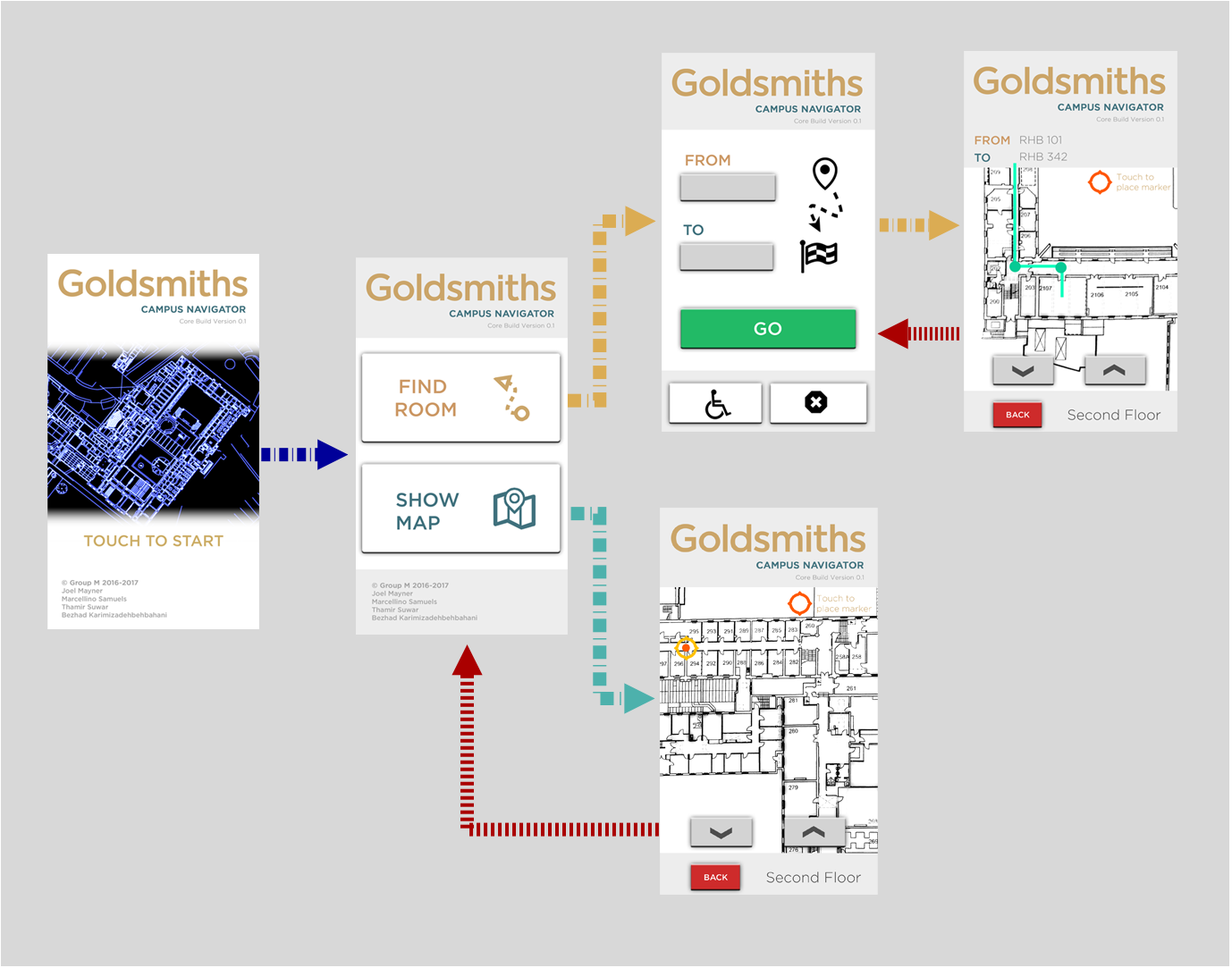
The main campus map was considered an important feature, even if it was not as vital to the implementation of the RHB building. As a relatively simple map, it would allow users to get almost anywhere on campus.

**Addendum**

The addendum build consisted of a list of items that were desirable but not considered vital if the group could not continue to develop beyond the core build the intent was for others to be able to add to the project.

GPS, room bookings implementation, map layouts for other buildings

**Development**



The design was intended to be clear and simple. The colour scheme was designed to complement the university’s font colour (Staníček). All icons were open source and chosen for their recognisability (). The user can pan and zoom in and out while viewing the map.

**Implementation**

Developing for the Android development studio was challenging, due not only to the group's unfamiliarity with the development studio environment but especially because of its reliance on the interactions between XML and Java code. The footprint of a single function, class or “activity” was considerably large, requiring the management of multiple and disparate files in the project directory for every single feature added. To compound development further, several inconsistencies and errors manifested within the Development environment itself, such as vanishing import statements and a deceptively unintuitive XML graphical UI. With the Graphical User Interface, adding a single element (such as a button or text field) unleashed a deluge of over-automated pre-sets which greatly restricted what lines could be added or removed without causing an error.

# Development Post-Mortem

Throughout the development of our application there have been a large variety of issue concerning the group work. The group member responsible for researching and testing was rarely available throughout the entire duration of the project and any work that was contributed was of a poor standard. This meant that the already large work load was then thrust upon an unprepared group. However, this is only the beginning of the turmoil that the group would face.

Unfortunately, due to unforeseen circumstances the group leader would be in and out of hospital leaving the project with no proper guidance or structure. This meant that the only available working members were the application designer and developer. As a result, meet the deadlines originally proposed in the Gantt chart became almost impossible and this lead to a code and fix style of development during the later stages of the project. The group was unable to properly utilise any development methods as a result.

Despite all this the developer and designer still managed to hold regular meetings as a two-man group to discuss the progress as well as how to deal with the huge workload. Communication between the designer and developer was good which allowed them to work together face to face and produce a good quantity of work despite the lack of members.

The majority of the meetings with the supervisor were orchestrated by the developer. For many of these meeting only the designer and developer attended.

Next time the group would be split into two separate group and a new project selected to create a more realistic workload for such a small group. More working meeting would be help to increase the productivity of the members and proper deployment and research tools would be used. Agile would be the best solution as it would force members to work together and produce frequent results.

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 could 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.