
Requirements Specification.

Author:
Thomas Keogh

Supervisor:
Dr. Myra Wilson

thk11@aber.ac.uk

Computer Science - G400

April 11, 2015

1.3 - Release

1 Functionality

The major aim for this project is to provide an android alternative to the already existing Access Aber web application. Currently the project exists solely as a web application which can be run on both iOS and Android through the use of a web browser. However the project is limited due to the time it was developed in and it is my aim to show that developing the application native to one environment can provide benefits which make the time and effort worth it. The idea is to not only port over the existing application but to improve upon it in ways that are seen fit. I feel that with the power given to a programmer within not just android, but mobile devices in general, the application can be made to be not only effective but intuitive and an asset to those who visit campus. The application will consist of several areas that will benefit the user and will heavily rely on the Google Maps Android API.

1.1 Route Finding

One of the main features to be included is that of a route finding system around campus. This differs from the web application due to the system actually finding a route, as it stands the application has around 200 stored routes which a user can select from. However these are displayed as a list and require the user to spend a large time searching through with no guarantee of there route actually being stored.

The android application will be developed in a different way. The user must be able to find there way around campus without having to search through 200 routes. The plan to combat this is to first have a much simpler way of route selection, ideally this will be two drop down boxes where a user selects where they are and where they want to go. This means that if we cover 20 locations, we just need 20 options instead of representing it as over 400 hard coded routes which is what would be needed in the current set up.

The second plan to combat the current issue is to remove most hard coded routes. All that ideally has to be implemented is for each building to connect to at least one other. If this is implemented correctly we can instead search for a Route rather than store them. This saves the user time and means that the application should be easily extendible.

One issue which arose in a meeting with the section of the University which initially set up the project was that while routes were shown not much else was. No information regarding the route and no indication of the difficulty of it, both of these would be of a lot of use especially to disabled users of the application. It is already known that colour coded routes should be possible within the application, combining this with a small window which shows information such as distance,elevation and steps on the route should fix this problem which has been highlighted.

1.2 Route Plotting

Another issue that arose within the meetings was that the plotting of a route was done manually, the only visualization the user inputting it had is whatever setting was provided through Google Maps. It was suggested that a suitable addition would be a route plotter which let the user walk a route and logged it for them.

Due to the planned nature of the system all that will hopefully be required is for the user to plot one route that connects to the graph of locations. By following this process we can fulfil another requirement that was brought up within discussions, the adding of information about a route. It may be possible to build a step free graph, thus allowing guaranteed step free routes. Some buildings are inaccessible so they would have to be omitted from this feature.

1.3 Location Based Help

A further requirement is the fulfilment of a help service for those on campus. Aberystwyth's campus can be a hard place to traverse along with confusing if it is your first time visiting. Due to this it is very possible that a user will require help to either exit a building or enter another, while help will be provided through another feature mapping the indoors of buildings will not be included in the application due to time constraints. To combat this a help service will need to be included, this will display to the user the closest person who may be able to provide some assistance to them.

Not only will this help the user if its needed, the presence of it should assure them even if it is not required. It has been noted previously that even some members of staff with mobility issues do not enter some buildings due to them being poorly labelled and other factors. This may be somewhat of a help to a complex situation.

1.4 Building Display

This is a feature that will be brought straight over from the existing application with only a few changes. It will provide a list of categories and let the user choose which they would like to view and then provide a display of where they are.

The feedback from the initial application is something which will be of great use here. There were several possible additions including pictures instead of just markers, something which is very possible and the inclusion of building names above names of lecture theatres as it can be unclear exactly where something is due to departments sharing buildings.

1.5 Multi Lingual Support

Due to this being a Welsh University it is required that welsh language support is provided for the likely event that a native welsh speaker uses the application.

This is an area which can pose problems. Android appears to have little to no support for the inclusion of Welsh language however new locales can be created and used which should help solve the problem.

2 Interfaces

2.1 User Interface

2.1.1 General User

The General User must have access to all features except the route plotter. The route plotter gives them no added actual use for their specific needs, however it is currently undecided if the project will be split up into two separate projects or just built as one and adapted from there. Having the Route plotter within the main application may confuse the user to a degree as unless you are aware of what its doing it will not appear to provide much of a benefit. It is likely it will be left in unless further use can be seen in splitting it up from the main part of the development.

Future development could focus on providing different User Interface based on the user type. This could be focused down further to provide different functionality based on a users mobility level. However in the time scale provided it is unlikely that this feature will be done now, if anything some level of customisation may be provided relating to route finding but not much else.

2.1.2 Staff Users

Staff users need access to all features, for addition to the system and to check everything is up to date. This is not a problem.

2.1.3 Encountered User Criticisms

Using the already completed user feedback results for the web application we can already start to set out some rough requirements for the look of the application and what needs to be shown. One major criticism was the fact that users are dropped straight into the map, this means that from there you have to work out what does what within the application. This also links to the problem of the buttons used currently being small and non intuitive which causes issues in itself.

This means its become a requirement to fix these base problems, most other problems highlighted link back to the confusing nature of the current User Interface. A huge improvement needs to be made to this, while some elements like the colour scheme are suitable for the purpose overall it is clear why users have issue with it. A possible solution is having a menu of sorts that is clearly labelled which then launches separate features meaning a simpler UI on each screen and no cluttering or extensive menus.

Other criticisms which effected user experience were related somewhat to the labelling of information within the site. It was not always clear if a labelled room was correct due to it appearing around other rooms for other departments. This is due to departments sharing buildings and is a fairly simple fix as all that needs to be done is to label each room with its respective information.

2.2 Google Services

The application will have to interface with several of Google's services. This mainly relates to interfacing with the map service which requires an API key, in this case API keys are free and will give us more than enough requests a day to develop the application. Using the Google Map service gives us a large range of benefits due to their thorough API, this allows for the development of elements like visualising a route on screen, showing a user there progress and markers for displaying building locations.

Having this thorough API is key to the success of the project, other possibilities have been explored including Open Street Maps but due to the developers past experience and what are seen to be gaps within the OSM API the Google Service has been chosen. While the service lacks some of the detail that may be seen as useful the benefits it gives more than make up for it and means that we can cover any gaps with our own implementations. The ability to customise elements of the maps is also a welcome addition and should help with the development of the labelling of buildings and displaying route paths and their grading.

2.3 Hardware

The application will also have to interface with hardware elements within the mobile devices. This mainly means the GPS module and possibly sensors to detect the gradient of a slope. Interfacing with the GPS module is made easy through the use of the Location Manager available through Android, this provides us with a periodic update of the users position allowing us to plot routes, display their position to them and to make estimations on who is best to contact for help.

It is also possible that the Camera will be used to help with plotting routes however it is more of a suggestion than a requirement. If this is implemented it means a picture would be displayed of both the start of the route and the end destination. Doing this will mean that the

user has a visual aid to help with finding their building as there could be some error with very close together destinations.

2.4 Interfacing with itself.

The application will also have to interface with itself in a way, the files created by the route plotter need to be ready to just add into the projects resources and work straight away. While this is less of a straight one to one interface the elements of the application need to be compatibly with each other, maybe an obvious requirement but necessary.

3 Performance

3.1 Offline Performance

There are several areas within the application where performance can be effected by what is to be completed. One major issue with this is that the application simply cannot work with no Internet Connection, at least not the first time round. The application has to load in the Google Map object and to do that and have the relevant imagery a connection is required. In this case the only real requirement is to at least notify a user of this and notify them if their is no current connection. A blank screen is shown by default when there is a lack of connection which needs to be remedied. It is suggested that a tech spike into caching imagery of the surrounding area is completed to make the application fully functional even when offline.

3.2 Searching Algorithm

A further area which could provide small problems, especially on older phones, is the searching for a route. However it is unlikely with the expected size of the graph that the algorithm created will cause too much effect. It is hard to set a requirement regarding the search, we need to it perform well but without an existing set up it is hard to test what we feel is a good range. However for the current stage of the project it is sufficient to say we need to test on a range of devices to find a good middle ground.

3.3 Possible Problem areas.

In addition to the performance issues previously we also need to ensure the performance of the GPS module, it is very possible that errors are made using cached values for the latitude and longitude but this should be a fairly easy fix. By guaranteeing new readings we improve our accuracy and provide the user with the route they actually plotted.

Performance wise we also have to take into account the responsiveness of the application. It is a common problem to encounter freezing and unclear interfaces within applications. To combat this steps need to be taken to not only improve code quality but provide a view that keeps the user updated on what is happening. Its not too much of a problem if it takes two seconds for a route to be found if the user is at least told about what function is being carried out.

One last problem area involves the performance on a range of screen sizes, the UI has to be coded in such a way that it does not depend on fixed sizes and instead relative ones. By doing this we ensure a consistent and appealing design despite device.

4 Attributes

4.1 Maintainability

A big issue with the application will be the maintainability of it; there is no guarantee the initial developer will always be around to change or fix it and as such this provides some restrictions on development. It means a high level of code quality needs to be maintained, while this is ideal in all projects it is maybe more so in this project as the chances of it being passed on could be quite high. This means full documentation, concise and clear comments and possibly JavaDoc but this will be decided at a later date.

Maintaining up to date information regarding locations of buildings and facilities is also something that needs to be considered. The issue here is whether we want to trust that a user can access the internet and gather the needed information from there or if it should come on device. Keeping the information on device means that an update is needed to change any of the information. Maybe not an ideal solution but possibly the best.

4.2 Expandability

This is a section with a large amount of requirements, the solutions to which will be fully considered in the design documentation due to the in depth analysis needed. The first problem is that its very possible the application will need expanding in future, because of this we need to make it as simple as possible to expand it for users that might not have large technical experience.

Due to this the code will have to read in information, and analyse it, using no built in constants. It needs to be developed in such a way that the files that contain the information guide the software as such. By doing this we keep a solid sensible structure to the information which will be easy to pull out with file reading algorithms in the code itself. The problem with this is that an error in a file could cause huge problems, however error checking in the file reader should help us solve that problem in somewhat of a sensible way.

4.2.1 Expanding the Graph

The big issue with expandability is the ability to add new locations to the route finder and as such expand the graph. It is possible that users with low technical expertise could be the ones responsible for expanding the project in future and as such the initial development has to make it easy for changes in the future.

To do this it is suggested that a simple set of instructions is clearly written up which describes in detail the process of adding in a new location. Ideally this will just mean creating a new file that contains the information of a locations name and a route into the graph of nodes. By implementing it in this way the maintainer does not have to understand the workings of the search or the traversal of a graph, just how to write out a basic file. A location file should ideally contain the start name, then details about its connecting nodes and the latitude longitude points which make up the link between the nodes.

It is essential that no code has to be changed to include these additions, relying on fixed variables in the code will cause large issues further down the time line of the project, beyond the initial building.

4.2.2 Expanding Choices

A further issue which arises with expandability is with the adding of new locations and how we can make it so they are recognised and added to existing menus in the application. This mainly

effects the selecting of the start and destination in route finding, the current plan is to load all possible places from a single file so adding to the menu would simply mean attaching a location to the end of a word file, something most people are capable of.

By having adaptive menu choices we keep further developers away from old code that may not be as intuitive as it might need to be. Having the application set up so that it can be expanded without the editing of existing code will be a real bonus as was described in the initial meetings.

4.3 Security

Security issues are something we do not really need to consider given the nature of the application. The user does not provide us with any personal information and there is no sensitive information included within the application. All data used will be publicly available and the code open sourced.

One way in which this might change in the future is the inclusion of directly uploading routes to the application and having devices load them from there. If this is developed in the future it is advised that great care is taken due to the damage that can be done through the inclusion of fake locations and misleading information.

4.4 Design

Design is an attribute that will need further research, this is due to Google releasing design guidelines to improve the quality of android applications due to an increasing opinion of iOS applications looking much cleaner and professional. However one consideration is access for those with vision impairment.

During the project time line and definitely in any further development a consideration to take is assistance to those with impairment. This can come in the way of multiple colour themes to help those with types of colour blindness as well as a possible sort of Sat Nav feature.

This feature would read out directions for the user, not only benefiting those with vision impairment but any general user. Maps can sometimes be a little unclear so if we provide instructions for when a user is at a specific GPS co ordinate then we know we are giving them enough information to properly follow the route provided.

4.4.1 Route Design

The design of the route displayed is also an important element with a range of requirements. First of all as previously described the user needs to be able to see the difficulty of the route. With the plan being displaying large routes made up of much smaller ones we will be able to display the difficulty of smaller sections than the whole path. This means users can see where the difficulty may lie and possibly find a way around it. With campus being fairly condensed with winding paths and roads its sometimes difficult to tell what is coming up. For disabled users to know that a path further on has a high incline could be valuable information.

To build on route design it is also required that the display is intuitive. This should be helped by the facilities provided by the Google Maps API. Being able to show a users location and orientation will be very useful especially if a custom arrow graphic is used rather than the standard dot. It is also important that the start and end points be clearly labelled to help with orientation.

4.5 Overall User Interface

The application must follow a consistent and appealing graphical interface throughout. This will help build up the users familiarity and help improve the intuitiveness of the design. This includes several guidelines which will be laid out in more detail in the design document but include -

- Options are selected from Expandable List View
- Pop up menus used for small actions on map screens.
- Custom Button XML to provide a theme appropriate display.

5 Design Constraints

The design constraints for the project are fairly obvious. The project must be android compatible and will be developed in the Android Studio IDE. It will be tested using both unit tests and black box testing with users who have no experience with the application.

It must be compatible with the lowest Android API level possible however what that currently is will not be clear until further on into development. The current aim is no higher than API level 14 which should be possible with the current estimated feature list.

6 Final Comment

The project will be developed as closely to this requirement specification as possible but may deviate due to the nature of software development, while this document details what is thought to be the current best practice it is possible that the final project deviates slightly from it.