Adapting Access Aber making use of native Android functionality.

Final Report for CS39440 Major Project

Author: Thomas Keogh (thk11@aber.ac.uk)

Supervisor: Dr. Myra Wilson (mxw@aber.ac.uk)

17th April 2015

Version: 1.1 (Draft)

This report was submitted as partial fulfilment of a BSc degree in
Computer Science (G400)

Department of Computer Science
Aberystwyth University
Aberystwyth
Ceredigion
SY23 3DB
Wales, UK

Declaration of originality

In signing below, I confirm that:

- This submission is my own work, except where clearly indicated.

- I understand that there are severe penalties for plagiarism and other unfair practice, which can lead to loss of marks or even the withholding of a degree.

- I have read the sections on unfair practice in the Students' Examinations Handbook and the relevant sections of the current Student Handbook of the Department of Computer Science.

- I understand and agree to abide by the University's regulations governing these issues.

Signature ....

Date ....

Consent to share this work

In signing below, I hereby agree to this dissertation being made available to other students and academic staff of the Aberystwyth Computer Science Department.

Signature ....

Date ....

Acknowledgements

I am grateful to...

I'd like to thank...

Abstract

The aim for this project was to provide an alternative android native application to the current Access Aber web application, while also improving on the groundwork laid out by the initial work. This was done due to the help it can provide to users who are visiting the University as well as providing at least some assistance to those with disabilities. Aberystwyths campus can be a hard place to navigate and the benefits provided from a mobile device can benefit users to a large degree. The application was developed for the android OS using the Android Studio IDE and relies heavily upon the Google Maps API. It provides a range of services to the user, some of which were brought over from the existing application and improved, some new. These include route finding around campus, user refined displays of building locations, route plotter with application compatible outputs and a location based help system. This dissertation covers the original analysis, requirements, design and implementation of the project as well as details on the testing and design decision made throughout the development. The conclusion is an android application which met its original requirements specification and adds further complexity to the initial web version of the project. This includes route grading and the representation of this along with an extendable solution to route finding using Google Maps and graph searching techniques.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Background & Objectives

1.1 Background

1.1.1 Background and Preparation

Background research was compromised of analysing existing systems like the original Access Aber, previous work completed by the developer that used map technology within Android and general research into the map technology to gather a strong understanding of exactly what was possible.

A majority of time was spent analysing Access Aber and the currently existing criticisms of it, a lot of the problems were clear without the user feedback that had been provided but some criticisms were more subtle but easy to understand like how the application provides no main menu as such causing the user to feel lost from the start. It was also obvious that at a technical level, the included features were not overly complex, at least not to complete on the Android platform. Due to the application originally being developed on a system which allowed it to be run both on iOS and Android there probably were limitations from both sides which leads to having to develop around the weak elements in both platforms. As stated developing for a single platform should help us avoid these problems while also giving us the ability to take advantage of what is there. Due to this a large amount of research was completed into both UI design within Android as well as the technical benefits both Open Street Maps and Google Maps could provide the development.

Research was also completed into exactly how the mesh of locations would be created and represented, within the design specification it was concluded that a searchable graph was a possible solution and one worth researching further. This also meant analysing how other existing applications had mapped roads to graphs and the general area of map representation. While this gave a fair amount of answers further decisions needed to be made on both how to search the constructed graph and how to make it an extendible system for future additions. This was one of the main conditions brought up in meetings with the original 'customers' of the Access Aber application, it had to be functional after the initial developers had left. It was also described that it would be beneficial for the application to be developed in such a way that a possible future development that involved a central file store of the information for ranging versions of the application. Due to this further research was done on how we could leave the application in such a way that it facilitated the addition of this feature possibly without using it upon the current cycles completion.

Past work by the developer was also analysed for anything of use that may come out of it, the application analysed was proven to have functional working code and as such was a place where possible solutions could be found. This included implementation of maps in an Android environment, the logging of a route which is something that has been previously outlined as a key requirement and a fair amount of small features relating the monitoring of a users location and the information which can be gathered from that. Most information gathered from this however was fairly irrelevant, it was decided it was not the best place for referencing in future.

Finally the Android API and several Android libraries were examined for the benefits they could provide the development with, along with the search for several features that had already been selected for inclusion within the application. This ranged from simple research on Expandable List Views to research on the best way to present the application in an intuitive way, a lot of factors were gathered from the original feedback which guided a lot of the research. While a slide function for the screens was visually appealing the possibility of it confusing users was something that removed it as a possibility very quickly. Research was also performed into Google's Material Design program, a set of guidelines for good design within the Android environment. It contained a large amount of information relating to designing professional and good looking applications, while the information was too much to be fully applied in the projects time line some key ideas were taken from it relating to the design of elements within an application and general theme layout. Furthermore some sites were analysed for basic colour themes and how to implement a simplistic design without it looking unprofessional.

1.1.2   Interests

This project has been chosen due to a variety of reasons, revolving around both the developers interests and the possibility of the project leading to something that can have real world influences. The main source of interest is that the project allows for a development environment that takes input from the real world, mobile devices are of a great deal of interest to the developer due to the data that can be gathered from them and the manipulation of this data. Mobile devices are steadily changing the way we live, with applications that not only help us find where to go but tell us what we can do when we get there. Where other people have liked and even what is best suited to us based on past choices we have made and the device has noted. While some systems outside of mobile development clearly have massive real world influences it is felt by the developer that the easy entry and possibilities provided with mobile development make it one of the most accessible and revolutionary platforms.

With the application in concern providing information to the user it also means we can filter what information is shown based on user details. If the user says i cannot manoeuvre stairs for whatever reason, we could implement a function to provide paths with no stairs, this could obviously be expanded to include searching based on a range of filters. This could also be condensed down to a single graph, with different links becoming available with different information. This kind of possibility is what makes mobile such an interesting platform, not only can we show a map and a route through it, we can then guide a user through it, even ask them for information regarding it once they are done. Possibilities provided are wide and interesting, with so many choices available the future of the application is impossible to predict because of the many paths it could possibly follow.

Another source of interest is the real effect this application could have for an organisation such as Aberystwyth University, the confusing and difficult to traverse campus is not a problem

that can be resolved easily. With everything on campus set in stone it is simply not possible to move buildings and locations around. However with technology we can facilitate the use of campus without any physical additions to it at all. This means we can change the physical use of something with technology that does not itself directly effect or control the environment it is deployed in. An interesting possibility would be to examine users manoeuvring of campus before the application is introduced and afterwards, as well as the effect it has on how comfortable disabled users feel on campus. With the application very literally changing the way a user traverses the campus it is possible that some of the more unused walkways could become more travelled due to their inclusion in the application. Further research will have to be done on most effective travel paths around campus, it is likely that the ones currently most travelled are the quickest but that is not definite.

1.2   Analysis

A broad analysis of the problem with the currently existing Access Aber application is that while it provides solutions, they are long, problematic and make the user follow a convoluted process that may not be obvious to the common person, the exact person we are aiming to help with the application. A prime example of the problems posed is in the route finding within the application, it exists but requires the user to scroll through a huge list of possible routes and then takes them to a summary screen before finally letting the user see the route they need to walk. There are some very obvious, stand out ways, that this can be improved including the implementation of simple features like drop-down boxes for destinations and the removal of the summary screen.

Most of the problems in the application come from simple solutions to more complex problems, simple in the way that they do not fully cover the problem or provide a real benefit to the system. This appears throughout the application and based on the information provided seems to have been caused by the short time period it was developed in. One aim for this project is to provide a much more polished result with each problem solved properly rather than meeting requirements with a 'it will suffice' mentality.

Access Aber's initial form will be treated as a kind of groundwork for development, while no code will actually be taken from it the features will be adapted and built upon. This way we can build not only a port of the application to Android we can build a better version which in turn can act as ground work for any future work on the project, be it with the Android application or another new version on another platform. Having this groundwork and the feedback on it means that the some of the problems that need fixing are already highlighted for us, while the time line of the project may not be long enough for some of the features to be fixed most can at least be addressed.

A step by step guide which details various information such as major features and activities can be seen in the Design Specification (Section 2). The following requirements are what has been decided as key to the success of the application, they have been selected due to either their use to the user or the fact that they have always been included in the project and these aim to be improved versions.

1.2.1 Breakdown of Requirements

After much analysis the problem has been broken down into a set of requirements, some of the process to arriving at this point can be found in both the Requirements specification with further information in the Design Specification, however the reasoning will be fully discussed here. Each requirement will serve as a benchmark for the project, requirements were set as realistically as possible considering the time given and the work that needs to be completed.

1.2.1.1 Route Finding

Route finding is seen as by far the major feature within the application, not only is it the most technologically advanced it is also the most useful for our prospective users. As detailed, a graph system had been highlighted as the best way to achieve a representation of campus for us to apply a search algorithm to, with Breadth First Search (BFS from now) seen as a possible solution for the time being. This feature also has some planned extensions to it, firstly the path shown to the user should contain extra information pertaining to the path. This includes information about the difficulty of the route and features along the route such as stairs, it would be beneficial to have information for separate segments beyond their rating but this is a stretch goal and not something should realistically be expected. Due to the expected structure of the graph the implementation of displaying rating of segments within the paths should be attainable, with large routes being made up of many links between nodes it is almost to be expected in a system such as this.

It is key that the route finding solution be easily expandable, this is due to the maintainability of the application being key to its success. There is no guarantee that the original developer will be around for future expanding if the application is deemed fit for purpose and further developed, due to this it must be possible for the adding of new locations to be possible. An ideal solution would use a central file store to pull location information from, however the problem here which has been previously covered is that there is no guarantee the user has access to internet when using the application. Due to this a sufficient solution should see that the extending of route finding should be achievable by a novice programmer, ideally a common user if they are provided with clear instructions. A set of guidelines should be produced if the project is taken further.

A further stretch goal would be an implementation of route finding based on a sort of filter, whether this be a routes rating or features on the route. A current rough idea of this can be found in the Design Specification (2.1.1), this highlights the idea of using a separate graph altogether, this way if we make the search algorithm so that all it needs is a set of files in a set way we can have a variety of graphs present within the application. An ideal solution would use a consolidated graph, with different routes between different Nodes being considered, however this is seen as problematic in the current time frame.

A final possible addition, though not likely to be achievable in the time frame, is to provide access from anywhere to somewhere on campus. This means having a user be guided into the graph from areas outside of it, meaning dynamic routes would have to be created. Initial research shows this could be possible using the Google Directions API however it is likely that with other development ongoing this is likely to be feature developed outside of the initial time frame. However the use of this feature would be a real benefit for the application, a simplistic solution would have the user guided to the nearest node and then the search algorithm would take it from there. It is even possible that in future the whole graph would be replaced with the Directions API, this however would require the user to have access to the internet at all times and it is simply some-

thing that cannot be guaranteed. Due to this the hard coded graph and search related to the graph is likely to be present for the lifetime of the application due to the required off line capabilities.

1.2.1.2   Route Plotting

This desired features comes as a result of the original meetings, the 'customers' were unhappy with the current system of plotting routes, which were all done manually. Part of the solution is already achieved through the use of a graph, a location only needs to be linked to one other to have a route available to anywhere on the graph. However this should be further developed for the application to be seen as fully functional. The application should include a feature that lets a user log a route and have it printed out to a file compatible with the current route finding solution, this way the person responsible for the future maintenance of the application can visualize a route before they add it to the current system. A route should be able to be visualized while it is being created, so that the user is not blindly adding points. GPS co ordinates received from some devices can be far from where the actual user is. Functionality within the Google Maps API allows us to show the user where the phone thinks they are, this should help them plot accurate points.

The output file, the files used by the route finder, should also be human readable. This requirement is not as key as the others but should be achievable as the file reader will be coded for this exact purpose. Having a human readable file provides various benefits, mostly that of being easily editable by anyone who wishes to change information in the future. Its negatives are few, while a non human readable may be able to be smaller the benefits time wise will be minimal. This also means that route plotting using the in built feature is not required, files can still be made by hand due to the standard format to be laid out.

Finally the Route Plotter should provide a grading for the route that has been laid out by the user. This is the grading that will be displayed by colour on the route finder, while the initial rating is likely to be numerical this will be translated into the red, amber, green system that has been mentioned in the previous meetings. Having this easily identifiable system provides the user with a sense of understanding without having to state the feature on screen and cause cluttering in the UI. Ideally the route plotter will take into account cumulative distance, steps on the route and elevation. However after initial research elevation is not only sometimes unreliable but a complex overall problem, to do with both the misleading information it can provide and the problems in finding elevation. However an ideal solution would take all three factors into account fairly.

1.2.1.3   Building Display

Building display is a feature currently implemented fairly well within the original application and is something we can mostly just adapt, however there are improvements that can be made, mostly found due to the user feedback. This feature should allow for users to highlight building types and have them displayed on a map of campus, this way a user can browse around what they want rather than have to try and guess where certain areas of interest are. A user should also have there location displayed to them so that they know where a building is in relation to them.

One criticism that has come from the user feedback is that sometimes the display can be less than clear, this includes building departments sometimes not being named and just a general lack of clarity. A good example is buildings named after people just having the persons name, for example 'Hugh Owen'. Problems like this should be fixed for the requirement to be met. Again a

facility for easy addition to this feature should be added, preferable through on device files for the time being due to the previously highlighted problems with a central file store.

1.2.1.4   Location Based Help

Location Based help is seen as the easiest implemented feature, something that the web application also currently lacks. It is seen as more of a test than the other features which seem solid and long lasting, the resources available to the University may not be able to facilitate the full deployment this feature may require. In short the feature should be able to provide the user with the details of the closest person who could provide help, currently this is limited to the three libraries which have their accessibility officers details listed on line. However other buildings will have persons responsible for this its just their details need to be retrieved from either the departments the buildings are home to or the University.

A possible change to this implementation is the addition of porters to the help listed, if all three locations are too far away, a porters could be contacted instead. However a dialouge would have to be opened up with the University over whether they find these practices acceptable. For this application they will be implemented as a proof of concept. But in future they could either be removed or only available on set days when the demand for the services is expected.

A further idea which would have to be completed after the projects time line is to have two separate applications on open days, one for visitors which is the one being developed in this project and one for the helpers hired by the University for the day. The application provided to the Helpers would be linked with the Help function shown here, on Open Days the application would take a users request for help and give one of the Helpers their location. By doing this we provide a way for confused users or just users looking for a personal guide a way to acquire both of those. This could be completed in a fairly short amount of time with a developer who has experience with what is likely to be a PHP and SQL back end. However this is a future plan and not something to be expected from this application.

1.2.1.5   UI Design

Interface Design itself contains a set of requirements that can be mostly boiled down to one key aspect - an intuitive interface for the User. Currently there are several problems with the existing interface that can be fixed fairly easily, like the lack of a menu and unclear buttons. However a fully intuitive interface will take time to develop, testing and a good amount of inspiration from other sources. Within this project it is the aim to have a clean, practical layout that most users can just pick up and use without and previous knowledge of the layout. By doing this we guarantee less confusion and problem free use of the project, it also means we would get less users wanting to use the Help feature which if implemented would be a big bonus for the University and development team.

This requirement can also be broken down into further aspects than just 'intuitive design'. Firstly the theme used within the application must be consistent without, it would also be beneficial if it coincided with the Universities colours of yellow and purple but this will not be treated as a requirement as of now, due to the restrictions it places on the development team. Buttons should be labelled clearly with reactive designs to users touch, there are few issues worse than no responsive buttons that do not give the user the feeling of actually starting a process. The design should also

be simple with non overpowering colours, we want the user to feel comfortable and a confusing sea of bright colours over a map is not a good way to achieve this.

Future goals could be varied colour schemes based on colour blindness, with the application aiming to help those with disabilities ignoring those with colour blindness is a large oversight and is something that should be considered in the future. It may be possible to implemented in this version and is something that should be strived for. Another feature which is seen as a key requirement is Welsh language support. With the application being developed for a Welsh University the inclusion of this is key, current research points to this being more than possible despite Android not natively supporting Welsh as a current locale. This can be programmed in however meaning it should be achieved in the current time frame.

1.2.2 Itemized Requirements

Following is the list of requirements for the project, some requirements can also be seen in the requirements document however these detail a more up to date version with more focus on smaller details.

- FR1 - Route Finding - Guiding a User around campus, including the display of route grading and a no steps graph.

- FR2 - Route Plotting - Users must be able to plot their own routes around campus and print them to an application compatible file.

- FR3 - Location Based Help - Provide details of possible help available, possibly include the addition of contacting porters.

- FR4 - Building Display - A display of related buildings to be filtered by the user, contains accurate information on buildings and the departments in them.

- FR5 - Multi Lingual Support - Welsh translations provided in the application and a user choice on what language they would like.

- FR6 - A clean and simplistic UI with clearly labelled buttons and a mostly responsive design, possibly including colour blind themes.

1.3 Process

You need to describe briefly the life cycle model or research method that you used. You do not need to write about all of the different process models that you are aware of. Focus on the process model that you have used. It is possible that you needed to adapt an existing process model to suit your project; clearly identify what you used and how you adapted it for your needs.

Chapter 2

Design

You should concentrate on the more important aspects of the design. It is essential that an overview is presented before going into detail. As well as describing the design adopted it must also explain what other designs were considered and why they were rejected.

The design should describe what you expected to do, and might also explain areas that you had to revise after some investigation.

Typically, for an object-oriented design, the discussion will focus on the choice of objects and classes and the allocation of methods to classes. The use made of reusable components should be described and their source referenced. Particularly important decisions concerning data structures usually affect the architecture of a system and so should be described here.

How much material you include on detailed design and implementation will depend very much on the nature of the project. It should not be padded out. Think about the significant aspects of your system. For example, describe the design of the user interface if it is a critical aspect of your system, or provide detail about methods and data structures that are not trivial. Do not spend time on long lists of trivial items and repetitive descriptions. If in doubt about what is appropriate, speak to your supervisor.

You should also identify any support tools that you used. You should discuss your choice of implementation tools - programming language, compilers, database management system, program development environment, etc.

Some example sub-sections may be as follows, but the specific sections are for you to define.

2.1   Overall Architecture

2.2   Some detailed design

2.2.1   Even more detail

2.3   User Interface

2.4   Other relevant sections

Chapter 3

Implementation

The implementation should look at any issues you encountered as you tried to implement your design. During the work, you might have found that elements of your design were unnecessary or overly complex; perhaps third party libraries were available that simplified some of the functions that you intended to implement. If things were easier in some areas, then how did you adapt your project to take account of your findings?

It is more likely that things were more complex than you first thought. In particular, were there any problems or difficulties that you found during implementation that you had to address? Did such problems simply delay you or were they more significant?

You can conclude this section by reviewing the end of the implementation stage against the planned requirements.

Chapter 4

Testing

Detailed descriptions of every test case are definitely not what is required here. What is important is to show that you adopted a sensible strategy that was, in principle, capable of testing the system adequately even if you did not have the time to test the system fully.

Have you tested your system on real users? For example, if your system is supposed to solve a problem for a business, then it would be appropriate to present your approach to involve the users in the testing process and to record the results that you obtained. Depending on the level of detail, it is likely that you would put any detailed results in an appendix.

The following sections indicate some areas you might include. Other sections may be more appropriate to your project.

4.1   Overall Approach to Testing

4.2   Automated Testing

4.2.1   Unit Tests

4.2.2   User Interface Testing

4.2.3   Stress Testing

4.2.4   Other types of testing

4.3   Integration Testing

4.4   User Testing

Chapter 5

Evaluation

Examiners expect to find in your dissertation a section addressing such questions as:

- Were the requirements correctly identified?

- Were the design decisions correct?

- Could a more suitable set of tools have been chosen?

- How well did the software meet the needs of those who were expecting to use it?

- How well were any other project aims achieved?

- If you were starting again, what would you do differently?

Such material is regarded as an important part of the dissertation; it should demonstrate that you are capable not only of carrying out a piece of work but also of thinking critically about how you did it and how you might have done it better. This is seen as an important part of an honours degree.

There will be good things and room for improvement with any project. As you write this section, identify and discuss the parts of the work that went well and also consider ways in which the work could be improved.

Review the discussion on the Evaluation section from the lectures. A recording is available on Blackboard.

Appendices

Appendix A

Third-Party Code and Libraries

If you have made use of any third party code or software libraries, i.e. any code that you have not designed and written yourself, then you must include this appendix.

As has been said in lectures, it is acceptable and likely that you will make use of third-party code and software libraries. The key requirement is that we understand what is your original work and what work is based on that of other people.

Therefore, you need to clearly state what you have used and where the original material can be found. Also, if you have made any changes to the original versions, you must explain what you have changed.

As an example, you might include a definition such as:

Apache POI library The project has been used to read and write Microsoft Excel files (XLS) as part of the interaction with the clients existing system for processing data. Version 3.10-FINAL was used. The library is open source and it is available from the Apache Software Foundation [2]. The library is released using the Apache License [1]. This library was used without modification.

Appendix B

Code samples

2.1 Random Number Generator

The Bayes Durham Shuffle ensures that the psuedo random numbers used in the simulation are further shuffled, ensuring minimal correlation between subsequent random outputs [6].

```c
#define IM1 2147483563
#define IM2 2147483399
#define AM (1.0/IM1)
#define IMM1 (IM1-1)
#define IA1 40014
#define IA2 40692
#define IQ1 53668
#define IQ2 52774
#define IR1 12211
#define IR2 3791
#define NTAB 32
#define NDIV (1+IMM1/NTAB)
#define EPS 1.2e-7
#define RNMX (1.0 - EPS)

double ran2(long *idum)
{
  /*----------------------------------------------------*/
  /* Minimum Standard Random Number Generator           */
  /* Taken from Numerical recipies in C                 */
  /* Based on Park and Miller with Bays Durham Shuffle  */
  /* Coupled Schrage methods for extra periodicity      */
  /* Always call with negative number to initialise     */
  /*----------------------------------------------------*/

  int j;
  long k;
  static long idum2=123456789;
```

```
static long iy=0;
static long iv[NTAB];
double temp;

if (*idum <=0)
{
  if (-(*idum) < 1)
  {
    *idum = 1;
  }else
  {
    *idum = -(*idum);
  }
  idum2=(*idum);
  for (j=NTAB+7;j>=0;j--)
  {
    k = (*idum)/IQ1;
    *idum = IA1 *(*idum-k*IQ1) - IR1*k;
    if (*idum < 0)
    {
      *idum += IM1;
    }
    if (j < NTAB)
    {
      iv[j] = *idum;
    }
  }
  iy = iv[0];
}
k = (*idum)/IQ1;
*idum = IA1*(*idum-k*IQ1) - IR1*k;
if (*idum < 0)
{
  *idum += IM1;
}
k = (idum2)/IQ2;
idum2 = IA2*(idum2-k*IQ2) - IR2*k;
if (idum2 < 0)
{
  idum2 += IM2;
}
j = iy/NDIV;
iy=iv[j] - idum2;
iv[j] = *idum;
if (iy < 1)
{
  iy += IMM1;
}
```

```
   if ((temp=AM*iy) > RNMX)
   {
     return RNMX;
   }else
   {
     return temp;
   }
}
```

Annotated Bibliography

[1] Apache Software Foundation, "Apache License, Version 2.0," http://www.apache.org/licenses/LICENSE-2.0, 2004.

    This is my annotation. I should add in a description here.

[2] ——, "Apache POI - the Java API for Microsoft Documents," http://poi.apache.org, 2014.

    This is my annotation. I should add in a description here.

[3] H. M. Dee and D. C. Hogg, "Navigational strategies in behaviour modelling," *Artificial Intelligence*, vol. 173(2), pp. 329–342, 2009.

    This is my annotation. I should add in a description here.

[4] S. Duckworth, "A picture of a kitten at Hellifield Peel," http://www.geograph.org.uk/photo/640959, 2007, copyright Sylvia Duckworth and licensed for reuse under a Creative Commons Attribution-Share Alike 2.0 Generic Licence. Accessed August 2011.

    This is my annotation. I should add in a description here.

[5] M. Neal, J. Feyereisl, R. Rascunà, and X. Wang, "Don't touch me, I'm fine: Robot autonomy using an artificial innate immune system," in *Proceedings of the 5th International Conference on Artificial Immune Systems*.   Springer, 2006, pp. 349–361.

    This paper...

[6] W. Press *et al.*, *Numerical recipes in C*.   Cambridge University Press Cambridge, 1992, pp. 349–361.

    This is my annotation. I can add in comments that are in **bold** and *italics and then other content.*

[7] Various, "Fail blog," http://www.failblog.org/, Aug. 2011, accessed August 2011.

    This is my annotation. I should add in a description here.