Software Engineering Group 09 Project Maintenance Manual (Android)

Author:	Ben Weatherley [bew46]
Config Ref:	SE.GP09.MAN_ANDROID
Date:	8th May 2019
Version:	1.0
Status:	Release

Department of Computer Science
Aberystwyth University
Aberystwyth
Ceredigion
SY23 3DB

CONTENTS

1. INTRODUCTION

1.1 Purpose of this Document

This document is to provide information and answers to specific questions regarding the maintenance of the android application developed for our project.

1.2 Scope

This aims to outline any potential pitfalls or problems that could be encountered when maintaining/updating the software, and outlines areas that could most use improving or updating.

1.3 Objectives

The objective of this document is to provide answers, warnings and general information to be used by programmers attempting to maintain the android application.

2. PROGRAM DESCRIPTION

The application we have created allows a user to select a location and then displays a list of pubs from that location. The user can then view details about a selected pub. This is achieved by storing Pub objects in an array list. The user can also filter the list of pubs based on their attributes, which is achieved by copying pubs with matching attributes into a new array list and displaying that list. Finally, the user can check multiple pubs and begin a pub crawl with those pubs, a selected start time and a selected interval time, or alternatively create a random tour with a random number of pubs in the currently filtered list. The random crawl simply takes a random number of pub objects from the filtered array list of pubs, whereas the user specified crawl takes the checked pubs, placed them into a new array list and passes that on to the crawl activity.

3. PROGRAM STRUCTURE

The main structure of the program is that it is based on three sections: database communicator, controllers, and User Interface.

The database will retrieve necessary information from the database and check for updates if refreshed

Controllers takes care of the functionality of the whole program.

User Interface takes care of the interactions with the user and displaying the information.

As for the actual application there were 9 activities:

The SelectLocationActivity, which starts by pulling the locations from the database and displaying them on a list adapter, which a button onClickListener that fills an arrayList of pubs for the location selected.

The HomeActivity has no major methods, and instead works as a hub to redirect the user to other parts of the application.

The PubListActivity also has a listView, but it uses a custom layout (pub_list_layout) and therefore has a private class which extends BaseAdapter, and lets the program know what widgets are present on the listView. The only other method which does anything other than redirect the user is the createCrawl method, which creates an ArrayList of Pubs by adding all of the checked pubs in the listView.

The CreateTourActivity also has a listView as described above, but allows the user to move items up and down in the view, using two methods moveUp and moveDown which simply moves the selected item up or down one position in the arrayList and then refreshes the listView. The methods select startTime and selectTimeInterval use modular arithmetic to convert the time in minutes to an hour minute format.

The ViewTourActivity simply displays information about the current pub in the tour. The method updateTour removes the current item in the tour and then updates the page to display the new current pub.

The PubInfoActivity displays information about the selected pub when the activity is created and has no methods of note.

The FilterActivity allows the user to select desired attributes, and then filters the pubs based on these attributes. This is handled by the method doFilter, which adds pubs to the filteredArray list based on the returned value of the method Filter, which returns true if the pub in question matches the users desired attributes, and false otherwise.

The EmptyTourActivity contains the randomTour methods, which selects a random number between 1 and the number of pubs in the tour, and then adds that many pubs randomly to a pub crawl.

The CongratulationActivity, which has no real methods and just displays a simple congratulations to the user for completing the crawl.

4. ALGORITHMS

The most complicated algorithm (and one of the only algorithms) in our application handles the filtering of pubs based on the current values of each of the sliders, and just checks that each characteristic of the pub matches the users desired characteristics, if this is true, the pub is added to the array list of filtered pubs, if this is false for even one of the characteristics then the pub is not added.

Another algorithm is the algorithm of talking to the database. It must be run as Asynctask to be able to run in the background. Other than that, it pretty much the same as described in the *Design Specification [3]*.

5. MAIN DATA AREAS

Objects:

Pub:

Pub is currently the only object in our application, and stores important details of pubs, including the name, location, address, postcode, X co-ordinates, Y co-ordinates, description, an array of Booleans, and a boolean value to keep track of if the pubs checkbox is checked or not.

Some other important structures used throughout the application is a set of 3 array lists all of type Pub. One contains the full list of pubs at the current location, another contains the filtered list of pubs (this mirrors the full list if no filters are selected) and the final array list stores pubs currently in the user's pub crawl.

6. FILES

Our application does not create or read any physical files, and instead connects to a database stored on the Aberystwyth university server and reads information from there.

7. INTERFACES

Our application is using a database receiver interface. It acts the same way as our desktop application for this system. The two systems need to communicate with the database in different ways, due to differences in how normal java apps and android apps works.

8. SUGGESTIONS FOR IMPROVEMNETS

The first major improvement that could be made to this application is cleaning up the bad practices that were employed in its creation, namely the act of passing all the important data between each activity. These activities (where possible) should really be removed and replaced with popups that can still access the data stored in the main activity.

Another quality of life feature to make the code more readable would be to go through and rename all of the variables so that they follow a uniform naming structure, as we did not have time to do this at the end of the project.

A feature we were unable to implement was displaying the images stored (as a URL) in our database in our App. I did a bit of research on this problem and it seems it would be possible to convert the URL's to URI's and then display the URI's using image views.

A problem that was discovered when getting users to test our application that we did not have time to address is bigger buttons, as when loaded onto a bigger phone the buttons are too small and can often cause trouble for users.

A less important improvement would be moving things around on the UI slightly, as currently the "create a random pub crawl" feature can be a little hard to find as it is only accessible from visiting the "view pub crawl" option on the home screen when no pub crawl is currently in progress.

There is also a small bug in the program, where if you create a random pub crawl but do not confirm its creation (by choosing a start time and pressing the create crawl button) the pub crawl is still created, due to the array list being populated with items.

There is a bug where if the user selects random pub tour, and click the back button to try again, this results in the application always receiving the same number of pubs in the random tour. This is due to the app using random.nextInt that will not update in the previous activity's int and when used.

It may also be useful to create some automated tests so that they do not have to be run manually, which is the case at the present time.

Finally it would be useful to add some error messages to the program to alert the user as to why what they tried to do did not work. Currently if the user tries to do something 'illegal' (such as move an item up in a list view when its already at the top) the program will handle the action, but not report back to the user as to why the list did not change.

9. THINGS TO WATCH FOR WHEN MAKING CHANGES

Due to inexperience with android studio, this application has been programmed in such a way that new activities are called, and data is passed into them. Because of this, every button on the app that opens a new activity passes 3 array lists (the complete list of pubs, the filtered list of pubs, and pubs in the crawl) as well as the current locations name (e.g. Aberystwyth) and the start and interval times of the crawl.

Therefore, care must be taken whenever editing a button or creating a new activity, as if these values are not passed through then data will be lost which could result in no pubs appearing or the current crawl information being lost.

For details on how to improve this see section 8 of this document.

10. PHYSICAL LIMITATIONS OF THE PROGRAM

Requires an android version: 9 ("Pie") or older.

Recommended diskspace: 10MB, Requires: 5 MB

Recommended RAM: 60MB

11. REBUILDING AND TESTING

The application is simple to run now as it only imports the PosgreSQL library, will be located in the projects \lib directory. The rest of the imported classes are part of the standard library. As for tests no automated tests were created for the application as must be run manually in accordance with the test specifications.

Make sure to have the latest Gradle version.

REFERENCES

[1] Software Engineering Group Projects: General Documentation Standards. C. J. Price, N. W. Hardy, B.P. Tiddeman. SE.QA.03. 1.8 Release

[2] Software Engineering Group Projects – Producing a Final Report. C. J. Price. SE.QA.10. 2.2 Release

[3] Software Engineering Group 09 Projects – Design Specification. R. Reve SE.GP09.DESIGNSPEC. 1.1 Release

DOCUMENT HISTORY

Version	*CCF No.*	*Date*	*Changes made to document*	*Changed by*
0.1	N/A	06/05/2019	Original draft	BEW46
0.2	N/A	08/05/2019	Improved sections of functionalities	RUR7
1.0	N/A	08/05/2019	Added extra detail to the program structure	BEW46