Contact Manager

For CS 4301.003 Android App Development in the semester of Spring 2018

Created by Zachary Aaron Greene from 4/01/2018 to 4/16/2018.

Current version: v4

Introduction

This application implements a database to display user inputted contacts via MVC Architecture. It contains classes for managing display items, database and database objects, sensors, and etc. The main UI renders via RecyclerView view type with a defined layout and given data; it is what asks the database for information. The map portion displays with a MapView.

I used a lot if tutorials to implement this project, and it was quite the learning experience. A lot of frustration to be had, but opened my eyes to higher level Java programming and that Android doesn’t have to be scary.

Warnings

Currently the project files in com.example.aaron.maptest2 are **not arranged properly**. This is because I have not tested the map functionality bug I ran into above with refactored classes into better sorted and more descriptive packages. I am afraid this might break the map feature (although this fear is baseless), and have intentionally left them all hodge-podged together, as I know the application works this way. I don’t want to unintentionally break anything irreversible through refactoring.

How to Use

1. When the application starts, you will get a screen that says "no contacts" and a button to create one in the bottom right.
2. When you create new contact, you will get a prompt with fields. Fill in the required fields with whatever you want, there is no input checking beyond that it exists. To select a field, click on it and it will bring up the appropriate android standard keyboard to fill in that field.
   1. As designed, currently First name, Last name, and Phone number are required fields.
   2. Further buttons are descriptive and do as they indicate they should.
3. On the contact screen, hold and scroll to view multiple contacts. Contacts are displayed with last name first, followed by first name. Contacts are also displayed with their creation date tool-tipped above the contact.
4. If you would like to view, edit, or delete a contact, press and hold on the selected contact until a new dialog box shows.
   1. From this new dialog box you may select the "view & edit" box for viewing and editing
   2. Select "show on map" to show the person's address on the map. Brings up a loading screen and then a map activity that center-zooms on their location.
      1. If loading succeeds, it just moves on to the map.
      2. If loading the address fails, the application will never transition to the maps page. From here, the user should use the back button on their phone.
   3. Select delete to delete the contact.
   4. Further buttons are descriptive and do as they indicate they should.
5. Notice that the list always displays in order of Last name, regardless of case. You can reverse the order by shaking the device. Kind of buggy, though.

Bugs and Missing / Future Features

* **Obsolete:** Part 1's final I/O for contact storage. NOTE: was never implemented, as part 1 was not completed.
* **Missing:** Currently, the app does not support part 3's "edit contact creation date" feature.
* **Missing:** Currently, the app does not support part 4's "distance from user location to contact address on map" feature.
* **Bug:** Currently, newly inserted contacts will not display their creation date. Upon restarting the application, they will be displayed.
  + I have not investigated how to fix this.
* **Bug:** Once you click on an entry to insert text in the Edit / View contact screen, you cannot back out until you have reached the end of the fields.
  + I do not know how to fix this, as it requires more knowledge about how editText works.

Resources

Initial Research

This document was modeled with help from an application designed back in 2008. The guided code is linked below. I realized that wasn't going to be much help as I had to redesign this under DataBaseHelper since older methods were depreciated. I wasted a lot of time until I realized this was the case.

Link: http://www.cs.trincoll.edu/hfoss/wiki/Tutorial:Making\_a\_Contacts\_Application

Key Resources

http://www.androidhive.info/2011/11/android-sqlite-database-tutorial/

This one helped a lot, many of these functions are adapted from Ravi Tamada’s guide. I cannot thank this guy enough for helping me get off the ground with this project.

http://jasonmcreynolds.com/?p=388

Thanks to Jason McReynolds. In particular this one helped a lot in demonstrating a successful way to use SensorEventListener, a custom interface, and some parameters I could reference for shaking.

Other Useful Resources

Many of the following links assisted in my development of this project, it’s difficult to link particular websites to code bits; most of this was just reference or tricks. Specifics are offered in the classes in the project.

https://developer.android.com/reference/android/database/sqlite/package-summary.html

https://developer.android.com/reference/android/database/package-summary.html

https://stackoverflow.com/questions/12015731/android-sqlite-example

https://stackoverflow.com/questions/43495549/cannot-install-repository-and-sync-project-in-android-studio

https://www.androidauthority.com/get-location-address-android-app-628764/

https://github.com/obaro/SimpleGeocodeApp/blob/master/app/src/main/java/com/sample/foo/simplegeocodeapp/MainActivityWithAsyncTask.java

https://www.journaldev.com/15676/android-geocoder-reverse-geocoding

https://www.androidauthority.com/get-location-address-android-app-628764/

https://stackoverflow.com/questions/14827532/waiting-till-the-async-task-finish-its-work

https://stackoverflow.com/questions/16252269/how-to-sort-an-arraylist

https://docs.oracle.com/javase/7/docs/api/java/util/Collections.html

https://docs.oracle.com/javase/tutorial/java/data/comparestrings.html

https://stackoverflow.com/questions/5271448/how-to-detect-shake-event-with-android

https://developer.android.com/guide/topics/sensors/sensors\_overview.html

https://developer.android.com/guide/topics/sensors/sensors\_motion.html

Special Thanks:

Special thanks to Android studio for highlighting all the discrepancies between the adapted code and my own with nice red bits, which make editing in my changes so much easier. I saved a lot of time with this feature. The error detection and auto-completion are robust and intelligent, and I saved a lot of time on this compared to past IDEs I have worked with.