**Final Report**

**Software Development for Mobile Devices**



**Ruidong Xu**

**Michael Oluwasegun Oshinaike**

**Yitong Ning**

**THE UNIVERSITY OF SHEFFIELD**

**DEPARTMENT OF COMPUTER SCIENCE**

**2017-18**

**Overview**

Our goal is to develop a photo management app based on Android platform. The functions of the whole project can be divided into seven parts as follows:

1. to visually browse previews of photos
2. to upload the pictures from the gallery
3. to take pictures using camera and save these pictures to the gallery
4. to create a database to save the information including title, description, date and locations (longitude and latitude)
5. to show pictures uploaded from the gallery or the camera on a map
6. to inspect the details of a specified photo including title, description, date and its location on a map
7. to upload the photos and their information to a server

These points are basically required by the assignment . We also did some extra features in our app. In the following sections, we will explain detailed about how we design the whole app and some extra features.

1. **Functions**
   1. **Visually browsing previews of photos**

In the main interface of our app, we adopted a grid layout to display all the photos. The photos the users take with the camera or upload from the gallery will all appear in the main interface and if they want to see more details about one specified photo, they could click that photo and it will become big and some other information such as longitude, latitude, title, description and so on will be displayed in that interface.

* 1. **Uploading pictures from gallery to our app**

We defined a floating button in the main interface and set onclicklistener for it. Once

our users press that button, they will jump to the gallery and choose the photos they want to upload to the main interface. Once our users upload the photos to the main interface, our app will automatically set markers for these photos in the map. These photos and their detailed information will all be saved to our database which ensure that after the users close our app and restart again, these photos will still there and will not disappear.

* 1. **Taking pictures with the camera**

We defined a floating button in the main interface and set onclicklistener for it. Once our users press that button, they will enter the interface provided by the system for taking photos. After finishing taking photos, they will go back to the main interface and the photo they just took will show in that interface. Photos will be saved to local storage and database created by us. This will ensure that after the users close our app and restart again, the photos will still be in our main interface and will not disappear. We strongly suggest our users to turn on the location service when taking photos if they want their photos to be marked in the map.

* 1. **Creating a database to save image information**

We used a Room database to store image and their information. Every time you take photos with the camera or upload photos from the gallery, all these photos and their information will be saved to database. We also show the detailed information such as title, description, date, locations and some other information to our user in the show details page from the data we retrieve from the database. Our users could also edit the title and description of their photos in the show details page.

* 1. **Showing pictures on a map**

Every picture no matter whether taken by the camera or directly uploaded from the gallery, our app will automatically set a marker for it in the map. Our users could also click the marker when they check the map. After they click, a small window will pop up to show them the details of this marker such as latitude, longitude and snippet. They could also change the current map to some other different types of maps such as none, normal, terrain, satellite and hybrid.

* 1. **Inspecting the details of an image**

After the user click on the specified photo from the main interface, the photo will become big, and we set a floating button in that page. When the users click that button, a new page, from where they could see the details of an image including title, description, date, locations in a map and some other exif information. They could edit the title and description and hide the location in a map. They could also upload their photos to server.

* 1. **Uploading and downloading photo from a server**

The application has the capability to upload images to the required server when the user views the image in full screen mode with a tap on the upload button. The application checks if there is internet connection and then sends the file else send the listens for internet connection and repeat the aforementioned procedure.

* 1. **Some extra features**
     1. **A splash screen**

Every time our users open our app, a splash screen will appear first before they enter the main interface. We added some animations for the splash screen. They will see a text move slowly from the bottom which writes ‘Welcome to AnPhoto’ with the background turning from light to dark.

* + 1. **Right and Left slip images when reviewing**

When our users view pictures in full screen mode a slide function is provided to transverse to the images from left to right and vice versa. Normally, they need to click the back button to go back to the main interface and click another picture to check it. This provides our users a more convenient way of reviewing the photo gallary. After they click the picture from the main interface and get to a show picture page.

* + 1. **Showing different types of maps**

We provided our users with different types of maps when they check the locations of their photos in the map. The menu we created in the right corner of the interface, they could change from the current type to some other types of maps such as none, normal, terrain, satellite and hybrid.

* + 1. **Showing more details about the picture using exif**

We provide some exif details which we felt were important to the users of our application, including the image resolution, size of the image file, filename and image orientation.

* + 1. **Creating a switch to lead you to settings of opening GPS**

In our application a location switch was required for the users to turn on location service in order to get geo-tags of photographs taken at different location. After our users slip this switch, a location menu pops up with a switch for the user to enable location service.

1. **Division of Work**

|  |  |  |
| --- | --- | --- |
| **Functions** | **Who finished it** | **Comments** |
| Visually browsing previews of photos | Ruidong Xu & Michael Oluwasegun Oshinaike&Yitong Ning |  |
| Uploading pictures from gallery to our app | Ruidong Xu & Michael Oluwasegun Oshinaike |  |
| Taking pictures with the camera | Ruidong Xu & Michael Oluwasegun Oshinaike |  |
| Creating a database to save images and their information | Michael Oluwasegun Oshinaike & Ruidong Xu |  |
| Showing pictures on a map and set markers | Ruidong Xu & Michael Oluwasegun Oshinaike |  |
| Inspecting the details of an image | Ruidong Xu & Michael Oluwasegun Oshinaike |  |
| Uploading and downloading photo from a server | Michael Oluwasegun Oshinaike |  |
| A splash screen with animation | Ruidong Xu |  |
| Right and Left slip images when reviewing | Michael Oluwasegun Oshinaike |  |
| Showing different types of maps | Ruidong Xu |  |
| Showing more details about the picture using exif | Michael Oluwasegun Oshinaike & Ruidong Xu |  |
| Creating a switch to lead you to settings of opening GPS | Michael Oluwasegun Oshinaike & Ruidong Xu |  |
| Documents | Ruidong Xu & Michael Oluwasegun Oshinaike |  |

1. **Library Planned**

EasyImage: We used EasyImage library to capture images from the gallery, camera or documents without creating lots of boilerplate.

Room: We used the room persistence library which provides an abstraction layer over SQLite to allow fluent database access while harnessing the full power of SQLite.

1. **Bibliography**

* <https://developerandroidguide.blogspot.co.uk/2017/05/upload-image-using-okhttp.html>
* <https://stackoverflow.com/questions/12455737/how-to-iterate-over-a-set-hashset-without-an-iterator>
* https://developer.android.com/reference/android/arch/persistence/room/RoomDatabase.html