CS5744 Project
Application Prototype Details and Guide

An Informal Writeup about v1 of the Photo Tracker Application Project

12/01/2013

Code Ninjas

Eileen Balci

Priyaranjan Parida

Table of Contents

About this Document
This document is a write-up meant for providing information about the current state of the prototype iteration 1 and how to run the application. It is meant as an informal document for guiding and informing the professors about the prototype application itself and about the project itself.

The Design Document is a much more professional document and is what detailed the design decisions behind the prototype application as a whole and was used to guide the development of the prototype application. While not all aspects were implemented from the design document, what hasn't been developed would be considered future work for the next iterations in the project. In the design document, future work is also specified that would apply to the next version of the application.

Project Artifacts
The following is a list of Project artifacts and their locations as well as various tools that were used for the project.

Artifacts and Documentation:
Design Document - Part of Zip file submitted on Scholar: Documents Folder
Prototype Application Details and Guides - Part of Zip file submitted on Scholar: Documents Folder
Source code - Backend Service and Android Application - Part of Zip file submitted on Scholar: Source Code Folder

If the Zip file, with the source code included, becomes too big for scholar, the professors are urged to access the source code via the Git repository listed under tools below.

For your convenience a link directly to the zip download of the source files is provided here: https://github.com/NinjaShark/PhotoTrackerApp/archive/master.zip clicking this link will initiate a download of the zip file.

Tools:
GitHub page for Code Base: https://github.com/NinjaShark/PhotoTrackerApp
Trello for Task Tracking: Private Page
Google Drive for Document Storage: Private Page

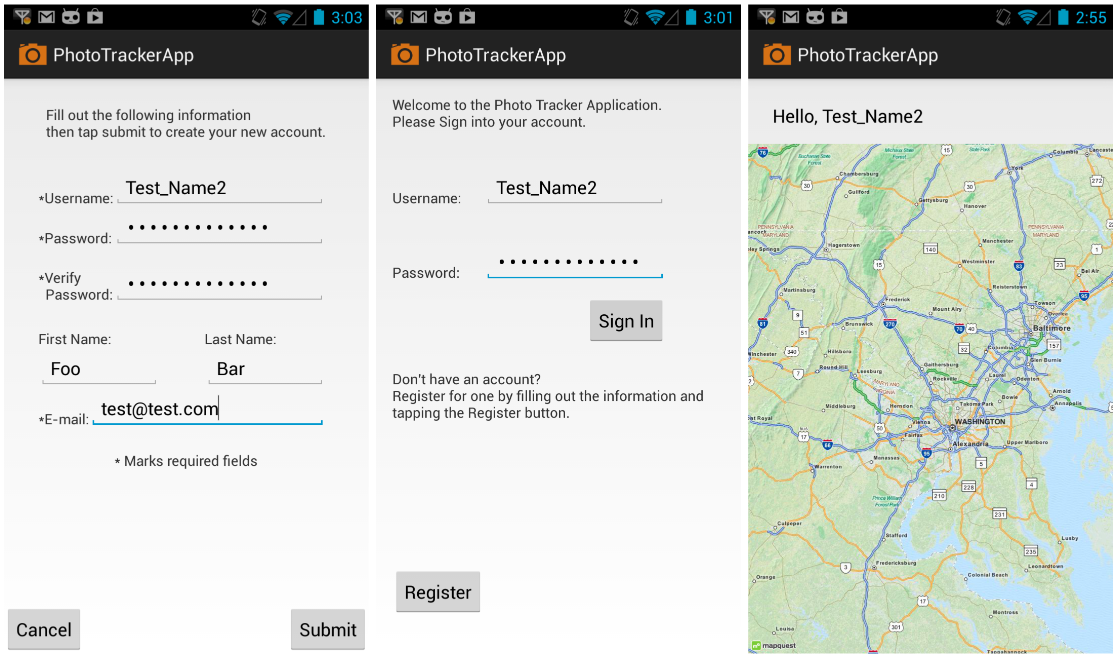
Prototype Current State
Following the design and mock-ups laid out in the Design Document, the team attempted at prototyping a version of this application. Knowing that the prototype would not be fully completed by the deadline, it is simple a look at what was accomplished in just a 2 week development time period (iteration 1). This section will go over what features were accomplished for the prototype application.

The Look and Feel

Mock-ups were drawn out and described in the Design Document. These mock-ups aided the look and feel of the application prototype. For the version one prototype, not all views or functionality was fully implemented due to time. Of the views implemented though, they followed the mock-ups pretty closely with the main difference being the Homescreen the user sees when they successfully login to the application. This view's User Interface (UI) was changed due to the fact that time was not going to allow for the functionality of the map view to be reached if all the views that were mocked-up were implemented.

In order to show what the Map view would have looked like, it was included in the Homescreen view instead. This was to allow for a demo of that capability at a minimum for this prototype.

Figure 1 is a series of three screenshots of the views that were completed with functionality for the prototype. The screenshots are following the order of a common workflow scenario of a user Registering, Logging in and viewing the Map.



Fill out the following information
then tap submit to create your new account.

*Username: Test_Name2

*Password: •••••••••••••

*Verify Password: •••••••••••••

First Name: Foo

Last Name: Bar

*E-mail: test@test.com

* Marks required fields

Welcome to the Photo Tracker Application.
Please Sign into your account.

Username: Test_Name2

Password: •••••••••••••

Don't have an account?
Register for one by filling out the information and tapping the Register button.

Hello, Test_Name2

Figure 1 Screenshots of the Registration view, Login view and Map view

Implemented Features

Due to time constraints, not all features laid out in the initial proposal were met for the prototype application, although they are discussed in the Design Document. The following features, however, were implemented in the initial prototype application.

- The user can create an account [Register]
- The user can login using a username and password entered when they registered [Login]
- The user can view a map [Map View]

Additional features that were implemented include input validation. For example, passwords had certain criteria that had to be met such as being at least 8 characters long, having at least 1 upper, 1 lower and 1 number or special character in the password. Usernames had to be alphanumeric but also allowed the user of underscores. Emails had to be of the proper Email format and the Username, Password, and Email were required fields.

The Registration and Login communicates to a backend server where the data is stored in a database. This is how the user is authenticated against the username and password they enter.

While it doesn't seem like a lot was implemented, it was surprisingly a lot of work over the two week time period that we had left to develop in.

Unimplemented Features

The following features that we wanted to implement but ran out of time where the following.

- The user can logout
- The user can take a picture
- The user can view a picture
- The user can view pictures on the map as pinned locations

This was the core of what our application was about, unfortunately it was also some of the most complex and by the time the registration and login functionality was completed, there was not enough time to start developing the camera functionality and image location extractor (to get the geolocation data from the jpg image using Exchangeable Image File (EXIF) data) and plot them on a map.

Future Features

Aside from the unimplemented features that still need to be completed in the next iterations, some future features for a second version have already been thought of. These features are not part of the initial project scope and would be something that would be implemented after the first version of the application is released.

These features would be the following:

- The user shall be able to add friends
- The user shall be able to make a photo as public, private, or “friends only”
- The user shall be able to make their map public, private, or “friends only”
- The user shall be able to view friends photos [if marked public or “friends only”]
- The user shall be able to view friends map

The implementation of such features would rely heavily on the backend service and database, but it is also a reason those components were made in the initial version. Technically speaking, with the first version of the application, the user probably didn't need to have an account and could have stored everything locally, but, because these future features were known, it helped make the design decision to add some initial capability so the future feature could be added with more ease.

What was developed by the team

The team developed the following classes and layouts with sources cited for additional code or references.

Classes:

- LoginActivity
- Registration Activity
- MappingActivity

  Utilized the MapQuest API and used their example code to help display the map which is cited in the code and also cited below:

  - http://developer.mapquest.com/web/products/featured/android-maps-api/documentation
- DatabaseMapping
- InputValidator

  Utilized different Regex strings for validation, all of which are cited in the source code comments but also cited below:

  - http://stackoverflow.com/questions/336210/regular-expression-for-alphanumeric-and-underscores
  - http://regexlib.com/REDetails.aspx?regexp_id=2204
  - http://commons.apache.org/proper/commons-validator/download_validator.cgi

Layouts:

- activity_login
- activity_register
- activity_homescreen

The following backend service code was developed by the team:

Interfaces:

- IRegistrationService

  Utilized javax.ws.rs package for exposing as RESTful Services
- ILoginService

  Utilized javax.ws.rs package for exposing as RESTful Services
- IAWSS3Service

Classes:

- RegistrationServiceImpl
- LoginServiceImpl
- AWSS3ServiceImpl

  Utilized the Amazon Web Service SDK to leverage Amazon S3
  http://docs.aws.amazon.com/AWSJavaSDK/latest/javadoc/index.html

How to Test the App

Because this is an Android application, there are two possible ways to run and test the application. This section will go over both ways to run the application.

Running the .apk File on your personal Android device

If you want to test the Android application on a device, you can email the .apk file to your gmail account associated with your device and download/install it from there. However, you much go to your Android device settings > Security > Unknown Sources and make sure the Unknown Sources checkbox is checked.

Once installed, to run the app, click on the application called PhotoTrackerApplication, this will start the application on the Login screen. If you have an account, login with your credentials, otherwise register for an account and then login.

This application does rely on having Internet connection, please ensure the device is connected to a WiFi or has access to the internet thorugh 3G, 4G/LTE or similar connections before proceeding. If you run into issues while using the application, please see the Known Issues section of this document.

Testing the Application on an Emulator

If an Android device is not available for use for testing, another alternative is using an Android Emulator. Typically this comes with the Android SDK and Android Developers Toolkit (ADT). If you already have your computer setup with the Android environment, you can start your IDE (usually Eclipse) and import the project. Once the project is imported, you can right click on it and select “Run As Android Application”, this will start the emulator and should install the application on the emulator for testing.

If your computer has Internet connection, the emulator should also be connected to the internet. If you run into issues while using the application, please see the Known Issues section of this document.

Setting up and Android Environment

If you do not have Android setup on your computer you must do the following before you can test the application. This is a tedious process and is time consuming.

1. You must install the Eclipse IDE from here: http://www.eclipse.org/downloads/ [1]
2. You must install the Android SDK from here: http://developer.android.com/sdk/index.html
3. You must install the ADT plugin through Eclipse: http://developer.android.com/sdk/installing/installing-adt.html
4. Once all that is done, you should now be able to create Android projects in Eclipse. If this can be done, you are ready to import the PhotoTrackerApplicaiton project into Eclipse and follow the instructions for testing in an Emulator.

[1] Eclipse Standard or Eclipse for Java developers are usually the most commonly used for Android development.

Testing the Service Application on Eclipse

In order to test the backend services, follow the instructions below. These are also tested when you run the Android application because the Android application connects the backend services for user registration and authentication.

Setting up and Service Environment

1. You must install the Eclipse IDE from here: http://www.eclipse.org/downloads/ [2]
2. You must install the AWS SDK from here: http://aws.amazon.com/sdkforjava/
3. You must install the Maven plugin through Eclipse: http://maven.apache.org/eclipse-plugin.html
4. Once all that is done, you should now be able to create Spring projects in Eclipse. If this can be done, you are ready to import the Photo Tracker Application project into Eclipse.
5. Build the application using Maven POM and deploy the generated WAR file from target folder with RunJettyRun server. RunJettyRun can be installed as plugin through Eclipse: http://marketplace.eclipse.org/content/run-jetty-run#.Up14-2Qadmk
6. Run the RegistrationClient and LoginClient class as Java Application to test the exposed RESTful services and its response.

Possible Third Option

If none of the above options are feasible, it could be possible for the Code Ninjas to demo the application over VNC. It will be slow, but the idea of the functionality implemented will be understandable.

If this option is desired please contact the Code Ninjas via their Virginia Tech emails.

Known Issues

This is a list of the known issues with the current prototype

- Login Failure when correct information is given
    - Sometimes the login will fail even though a valid username and password is entered.
        - By clicking sign-in a second time after the first failure, typically the application will sign you in and you will see the home screen with the map view
- Registration even if already registered
    - If you try to register with the same credentials the application will tell you that you registered successfully, but in reality, the server bounced back with a “USER ALREADY EXISTS” response.
        - This is not really an error, but it makes the user think that they registered again when in reality their first registration is still the same.
- Application Crashes with Force Quit

[2] Eclipse Standard or Eclipse for Java developers are usually the most commonly used for Android development.

    - Sometimes the application will unexpectedly quit, this can be due to a few reasons.
        - Make sure there is internet connection, currently there is no exception catching if the network is not active and thus the app crashes when it tries to connect to the server
        - Make sure the internet connection isn't proxied, sometimes proxies can cause the network connection to fail and this will crash the app
- No Logout feature
    - For this iteration, a logout button was not implemented on the Android side, even though it was supported in the backend services.

These issues were found during testing of the application. They were not fixed because of time constraints, but, if the project continued, the next iteration would start by fixing the known issues and doing regression testing first. Then new functionality would be implemented for that iteration.

Lessons Learned

This section highlights some of the lessons learned through doing this project with the addition of prototyping an application.

Time Management

The development team should have left more time for development, however the main priority of the project was the quality of the Design Document and that is where the majority of the teams time was spent.

The second priority was producing a quality prototype, thus focus and attention was spent on checking input validation and ensuring the application looked tidy and useable.

Design and Agile Engineering

Overall, having a design upfront before developing does help a lot. An idea of what was going to be involved in the prototype was clear early on and the team knew it would be very unlikely that all the mentioned features would be accomplished by the deadline. But the point of Agile Engineering is to be able to take steps and build upon work that has been done in the last iteration to continue advancing the functionality until the end goal is reached.

In this case, if the project was to continue, the next iteration would have involved adding the camera functionality and ensuring that worked by the end of the iteration. The next iteration after that would have focused on the user being able to view the photos taken with the camera and being able to retrieve the geo-location data from the images. The main importance is to have a functional prototype by the end of each iteration. While this prototype may still be incomplete, it is still functional for what has been implemented.

Quality
Quality is one of the most important outcomes of a software application. Without quality, users will dislike using software even if the software provides meaningful assistance to a users needs. While this was the first iteration of a prototype application, it was important to the team to ensure a level of quality that was higher than 'just a prototype'. While the views are simplistic, it was an attempt by the team to layout the components in a meaningful and usable way that would prove "user friendly".

The application style uses the built in Android standard. This gives the look and feel that is found across various Android applications. This helps keep a level of consistency across the application and across Android applications that allow a user familiarity when it comes to using Android applications.

While the Design portion of this project was the main focus, there was some attention to quality given to the prototype. However, there was no Quality Assessment given.

Documentation
Even though there was a short time span to do development on the prototype application. All the Android code was heavily documented and commented. This is because documentation is important for any software system. Not only does it help the current developers remember and understand what was going on in a piece of code, it also helps new developers pick up and understand the code as well.

Documentation of the code is a quality of itself. While the end user will not see the comments, the comments do help keep a level of quality in the code base itself, and that can be reflected back on the quality of the overall application as well.

Final Remarks
Overall this project was challenging and fun. It allowed the team to go through the process of designing an application to actually getting to prototype as much of it as could be done in a short 2-3 week iteration. This project not only involved the attention to the design document before beginning development, but it also involved the use of backend services along with an Android application that was part of the initial prototype. The team wishes they could have implemented more functionality for the prototype based on what was outlined in the design document.

During the project, it was important to the team to also keep in mind Quality. However, specific Quality Assurance metrics or assessments were not drawn out or implemented for use on the prototype. Once basis of Quality will be dependent on the end users like or dislike of the look and feel of the application, so the team supposes that the professors will be able to determine a level of quality of the prototype.

In the Android application comments, there are links to sources to some code that was used under various Free Use licenses. With open source growing, it is more common for developers to seek functionality that may have already been developed rather than “re-inventing the wheel”. This helps cut development time and cost if the functionality added integrates with the already developed code base.