

PhotoHound: Using Geographic Location, Pictures, and Adventure to Create a New Application for Social Media

Social media applications have become ubiquitous and they can take up huge chunks of our time. Most social media apps are used at home and can sometimes keep people from venturing outside. With this in mind, my team developed an application that would serve not only as an outlet for social media, but also as a fun way to get people outside, exploring the world around them. Samuel Robertson, Jack Murdock, Jordan Bates, and I conceptualized, designed, and implemented this idea in the form of an Android application. We chose this because we'd all had experience developing code for Android applications and it is easier to publish on the Google Play Store.

Our app was originally inspired by Geocaching. Geocaching is a real world game where people use GPS coordinates to find hidden boxes in urban, rural, and nature environments. There are several issues with Geocaching though. The boxes can be lost or destroyed, the boxes can sometimes be extremely difficult to find, and it can be inconvenient to purchase a GPS for the sole purpose of Geocaching. Our application took the basic premise behind Geocaching and revolutionized it for the next generation. Now, with our app people can explore in a similar fashion to Geocaching without having to purchase a GPS and without having to worry about difficult to find boxes that can be lost or destroyed. All they need is a smartphone and our free application to start creating and completing PhotoCaches. Our application also adds an avenue of advertising for businesses. Businesses and other organizations will be able to create sponsored PhotoCaches that will appear prominently for the users of the application. With these

new and updated features, we believe our application has automated and streamlined the basic idea behind Geocaching.

In order to create the potentially vast social community for our application we needed a server to hold all of the user information and PhotoCache information. We implemented a MySQL database that used server code written in DJANGO for commands to and from the server. We used a uWSGI which basically served as a load balancer so that all of the requests to and replies from the server wouldn't bottleneck and cause extremely slow processing when several requests/replies needed to be handled at the same time. Additionally, we used nginx as a proxy for our server. In order for the server to do all the things we needed to have a complex system of relationships between the many components of user's accounts and the PhotoCaches uploaded to the server. Some of these include the interests that a user has and the tags that are associated with PhotoCaches. Together these two components allow us to show a user all the PhotoCaches that have tags that match up with the user's interests. Without the server, there would be no way for our application to store all the PhotoCaches and for users to see all the PhotoCaches.

The actual Android application uses multiple activities so that the user can accomplish anything they might want within the application. When the app is first opened, the user sees a basic login page that allows a user to enter in their username and password to login to the app. There is also a prompt at the bottom of the page that allows the user to create an account if they do not already have one. On the page where a user can create an account, there are several fields. A few are required, namely the username, email, and password fields. There are also optional fields, namely the first name, last name, and description fields with some preferences at the bottom that are displayed as checkboxes. When a user inputs at least all of the required

fields, then the information is captured and sent to the server. After a user has either logged in or created an account they are taken to the streams page.

On the streams page, users can view PhotoCaches that have been uploaded to our server. The PhotoCaches are divided up into four different streams which can be viewed by swiping over to them. The four streams are Local, Popular, Personal, and To-Do. Each stream displays PhotoCaches fitting a different criteria. The Local stream displays PhotoCaches within 100 miles of the user's current location. The Popular stream is universal to all users and displays PhotoCaches that have been completed or added to To-Do lists the most. The Personal stream displays PhotoCaches that have tags that match the interests of the user. Finally, the To-Do stream displays all of the PhotoCaches that the user has added to their To-Do list. In each of the streams, PhotoCaches are displayed with their title, photo, author, difficulty, and description. If the user clicks on a PhotoCache then they will be brought to a page where they can view that individual PhotoCache with all of its details.

When a user is viewing a PhotoCache, they will see the information viewable on the streams page in addition to the latitude and longitude coordinates, the distance (in miles) and direction (cardinal direction) to the PhotoCache, and the tags for the PhotoCache. If the user clicks on the author of the PhotoCache on the view cache page then they will be taken to the Profile page for that user. Also, at the bottom of the view cache page there are two buttons that the user can click. One says "Add to To-Do List" and will send a message to the server that adds the PhotoCache to the user's specific To-Do List. The other says "Do This Cache" which will take the user to our custom camera activity. In this custom activity, a user will be able to see an overlay of the original image on top of their camera view. This will assist the user in lining up the perfect picture to match the original and this overlay has an adjustable opacity so the user can

see more or less of the original image. Once the picture has been, the GPS coordinates of the phone as well as the orientation (pitch, yaw, and roll) of the phone are captured and compared to the original coordinates and orientation of the PhotoCache. If the coordinates and orientation are close enough, then the user is taken to a reward page where they can view whatever reward that was set for the PhotoCache when it was created. After the user has viewed the reward, they are taken back to the streams page.

On the streams page there is a floating action button that can be clicked to create a PhotoCache. After being clicked, the user is taken to the camera where they can take a photo for their PhotoCache. When the picture is taken our app captures the GPS coordinates and orientation (pitch, yaw, and roll) of the phone so that it can be uploaded to the server. After the picture is taken, the user is taken to the create cache page where they will see the photo, the latitude and longitude, and fields where they can enter additional information for the cache. These fields are the title, description, tags, difficulty, and reward. Upon filling out all of the fields and clicking the create cache button at the bottom of the page, the user will be taken back to the streams page.

Throughout the app users will be able to click on a simple three-bar menu to navigate. This menu can take the user to their account page, their profile page, or the streams page. While viewing their account, a user will see all of the fields mentioned on the create account page. Here they will be able to edit any of those fields and then save their updates on the server. On the profile page, users will see the profile picture, username, number of created PhotoCaches, number of completed PhotoCaches, and description of user that the profile belongs to. All of the PhotoCaches that a user has created will also be displayed at the bottom of the page. If the profile belongs to the user then they can click on their profile picture to choose a new photo for

their profile. Another way that we made our app easy to navigate, was to have a back button in the top left of the app so that users can go back to the activity they were previously on.

Creating this application taught our whole team a great deal about Android development, real world Computer Science projects, and working in a team. In the future we plan to add a few more features to our app and then publish our application on the Google Play Store. We also plan to create an iOS version of the app to expand our user base.