

Exposure

Kristi Gable (gablek)

Melissa Khuat (mkhuat)

Vision

Problem: Avid photographers are always looking for interesting places for their photographs. Whenever a user wants to look for a new place to photograph, they often search for it online where results are not consolidated. For example, “Seattle Photography spots” comes up with multiple different websites with different suggestions.

Solution: Exposure will consolidate all information for both curated and user-submitted locations in one mobile application. Amateur and professional photographers will be able to find nearby locations while on-the-go, rather than having to research places beforehand.

Exposure--an Android application that fits the needs of the traveling photographer--allows its users to quickly search for nearby locations to capture photos.

Currently, *ShotHotspot* and *Stuck On Earth* are Exposure's competitors:

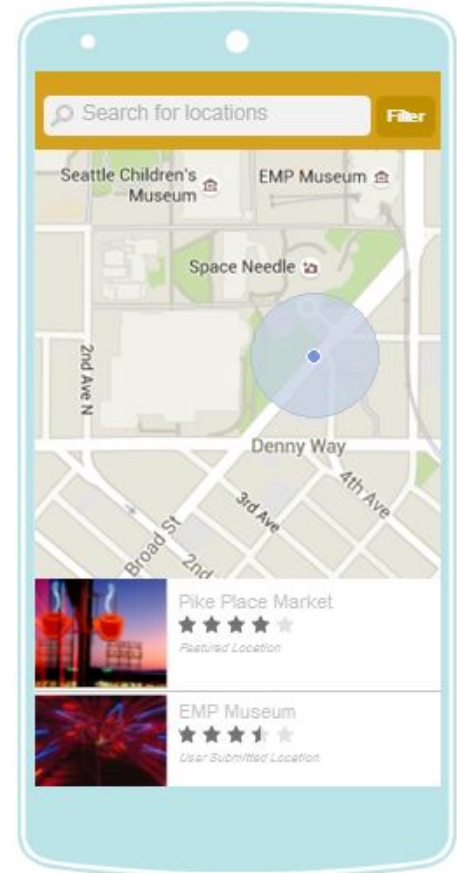
ShotHotspot is only available as a Web application, which is more suited for desktop browsers than smartphones or tablets; Exposure is a mobile app that allows photographers to find new locations while on-the-go.

Stuck On Earth is an application for both iOS and Android, but according to reviews, issues arise with connectivity: “The app suggests that a faster connection helps photos load faster. I tried viewing the photos using Wifi and on my computer and it still slow... the interface [also] needs improvement.” (Google Play Store)

Exposure will contain a clean, simple UI that focuses on performance and ease-of-use.

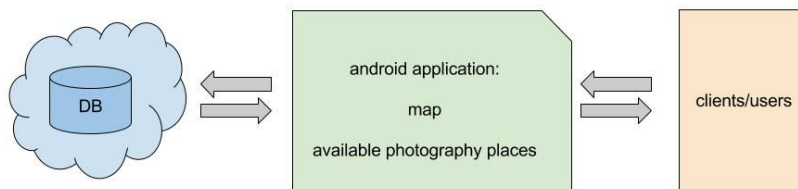
Compared to Yelp, TripAdvisor: Yelp and TripAdvisor target *tourists* and the *average adventurer* who are looking to explore areas. Exposure targets the *photographer*; interesting places to photograph are not necessarily places that are interesting places to explore. Thus, there are different use cases for the users of each application.

Scope: The first version will contain a map and list view of nearby locations that are curated by the application designers. Each location will contain a “details view” that shows the rating, category (landscape, architecture, portrait, etc.), address, and associated photos. Future versions will contain user-submitted locations, photos, ratings, and descriptions. Users will be able to login to an account and keep track of their “favorited” locations and share their posts with friends through a third-party application (Facebook, Twitter, etc). They can also earn trophies or badges based on their app involvement statistics.



Software Architecture

Exposure can be reasonably built in ten weeks and will contain three main components.



Starting from the backend, we will have a SQL database to store the information about the photography places (location, rating, category, etc) and the pictures that represent them. We want to utilize a product that focuses on returning results quickly and that can scale as our user base grows. One resource for us to consider using is Parse. It is a cloud-based product, which allows us to not have to store our own data locally, and is scalable. The next two parts are reliant on our database and receive direct input from the users. One of these is the map or location identifier of where the user is interested in. *Exposure* needs to be able to choose a location and get the photography locations at the same place or close to it. We will use the Google Maps API and take advantage of the functionality it has to do these calculations. The second part that interacts with both the database and user is the display of the details about each photography place. This includes both the list view once the location is selected and the details page that gives more information than the list.

The interesting part of this project will be designing our app to ensure that all parts are scalable. With the many features we want to add, this is definitely a challenge and will be crucial to the maintainability of our product.

Challenges and Risks

One notable challenge we anticipate while developing *Exposure* is that we need to learn how to work with the Android development environment in a relatively short amount of time. Although Android is based on Java, there are many new concepts that will be new to our team members, such as activities, fragments, content providers, and broadcast receivers. We will also spend a significant amount of time familiarizing ourselves with the frameworks that we are interested in using, including the Google Maps API and Parse framework. In order to tackle these learning curves, we will quickly start working with these tools as soon as the initial designs have been solidified; in addition, we hope to have at least one experienced Android developer on our team who can oversee production.

In regards to technical challenges and risks, *Exposure* must minimize battery drain and data usage, which were the two main complaints from users of one of the similar, alternative apps. Using location data and GPS technology may make this especially difficult. Also, once we start introducing user profiles to the system, maintaining the application will become increasingly complex; not only do we have to implement a user authentication service, but we also have to consider cases where users submit malicious or inappropriate content. How does *Exposure* filter through these unwanted photos? How does *Exposure* store an increasing magnitude of photos? Another addition that will take time to implement is the "User Profile", which includes the user's favorites and submissions. In order to confront these challenges, we will be combining two software lifecycle models--staged delivery and evolutionary prototyping--to guide our development of *Exposure*.