

**Suggest travel Location to user on basis of image recognition deep learning**

**Course Name**: Distributed Systems

**Professor**: Dr. Jinsong Ouyang

A Project Report

By

Paril Shah

Jaidipkumar Patel

Aaishwary Vadodariya

**TABLE OF CONTENTS**

1. Introduction
2. System Specification
3. Application Architecture
4. Application Screen Shots
5. New Technology Learned
6. Issues
7. Existing Traps
8. Conclusion
9. References

**Introduction**

The basic idea of this project is to develop a simple and efficient application which help the users to explore new places, attractions and destinations using posted images by the user in the past. The application constantly focuses on user directed search from the pool of images. Photos of these destinations or places guides new users to have better understanding through various perceptive. On the other side the active users who are visiting those places can upload their collection which can be used by some other user as reference in future.

We developed an application in iOS front end user interface it will manage a cloud database on which all users can upload their collections and similarly the same collections can be used as reference for users who are exploring new places. This iOS interface will provide various discrete categorical destination options like beach and forest etc. Along with country selection as primary search request criteria. These selected labels will initiate a search request in cloud database and return the relevant results. **User can then be directed to the location of the selected place, Maps can be used to direct the user to the location of the selected place.**

**System Specification**

**iOS API level :** iOS 10 and above

**Cloud Services :** Amazon Server, EC2 Server, RDS Server

**Back End :** J2EE with Spring and Hibernate

**IDE :** Xcode8.0 and IntelliJ IDEA

**SCM :** GitHub

**Application Architecture**

AWS

Rekognition

Request

Client

(iOS app)

Response

(RDS)

(RDS)

Request Response

Server

(AWS EC2)

Request Response

Database Server

(AWS RDS)

* **AWS Rekognition**

Amazon Rekognition is a highly scalable, deep learning technology developed by Amazon’s computer vision scientists to analyze billions of images daily. It uses deep neural network models to detect and label thousands of objects and scenes in the image. **We use Rekognition’s API for image classification in our project.**

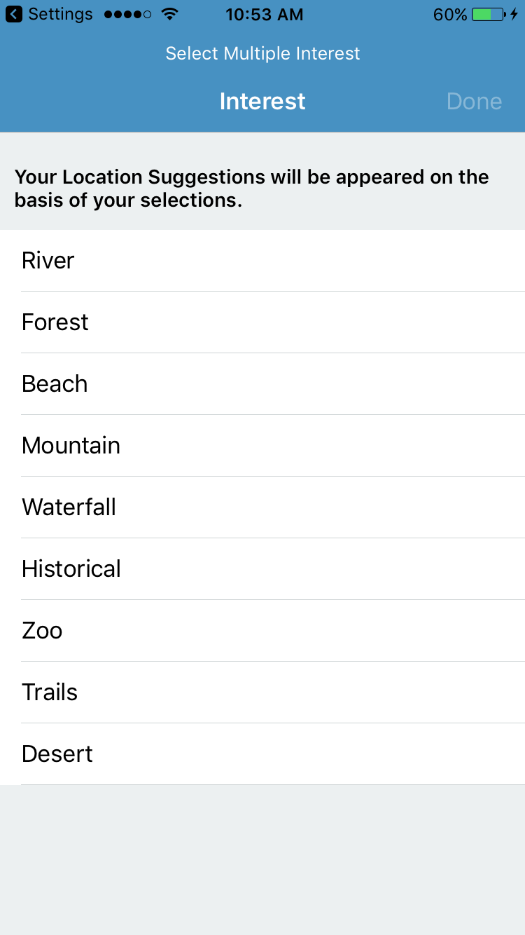
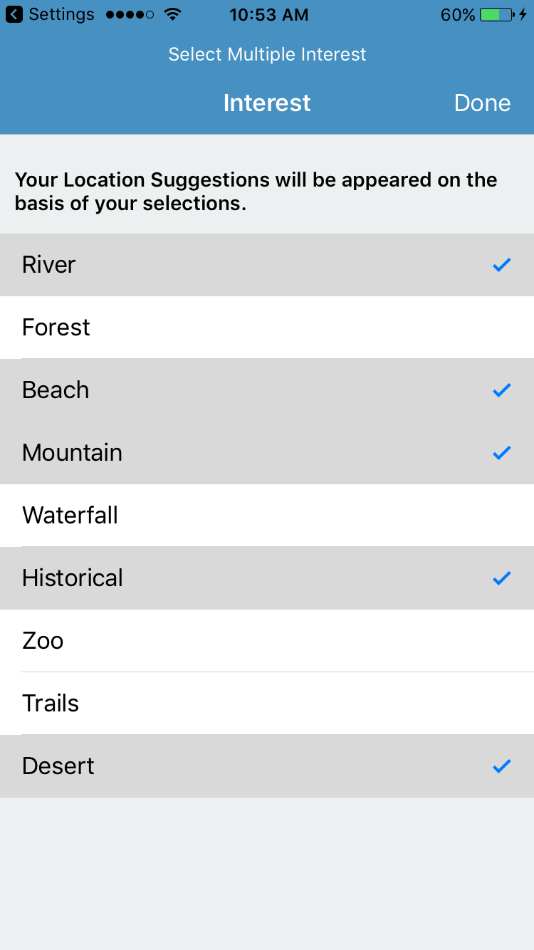
* **AWS EC2**

Amazon Elastic Computer Cloud is a web service that provides secure, resizable computer capacity in the cloud. As our project required a fair amount of resources to develop and run the application, we heavily used its web services to develop our project and has deployed the business logic on EC2 so that we can remotely access the code which adds easiness and efficiency to work together with team members.

* **AWS RDS**

Amazon Relational Database Service (Amazon RDS) makes it easy to set up and scale a relational database in the cloud. RDS provides six familiar database engines to choose from such as PostgreSQL, MySQL, and Oracle. We choose MySQL as our project back-end for Storing image data, travel location, categories etc.

**Application Screen Shots**

This is the first interface user interacts with, where they can select their various interest from the listed categories. User can also select multiple categories for more than one interest. Suggestions of related location and places are provided to user based on these categories selection.

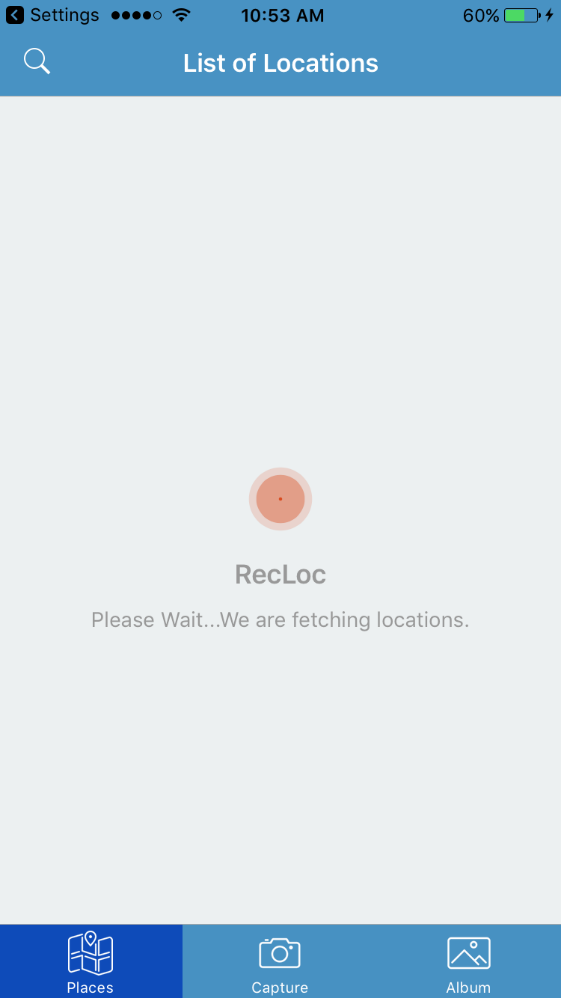
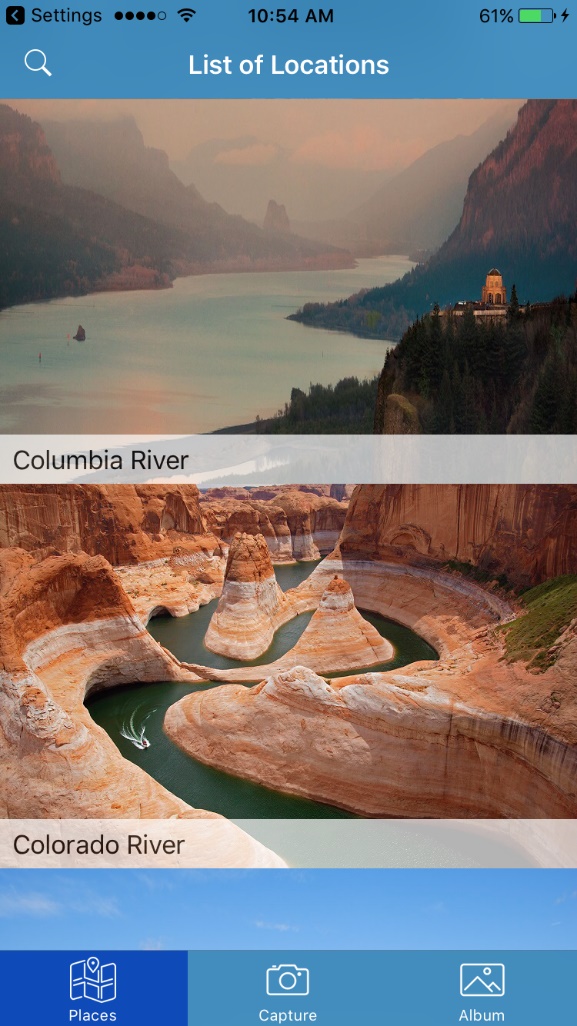
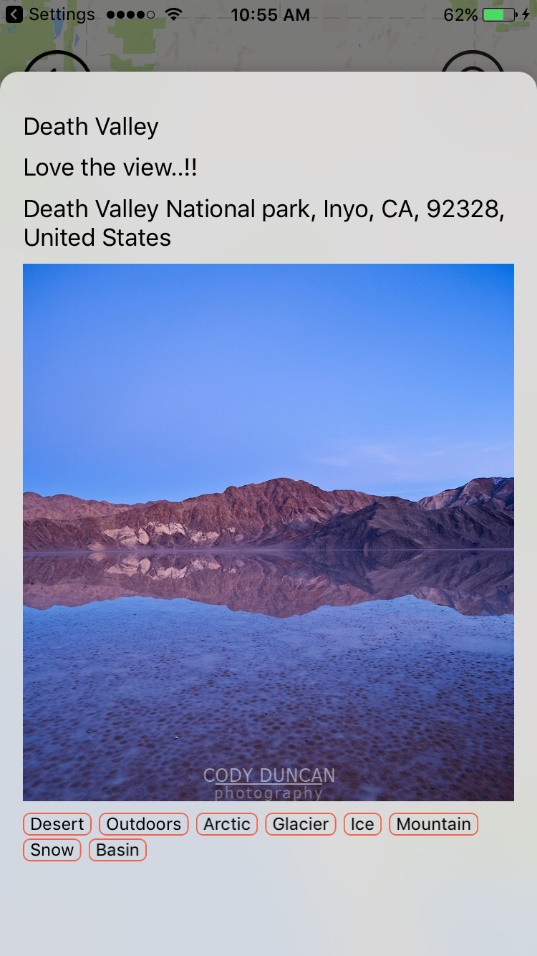
 

Image on the left is the loading session in progress where Suggestions are fetched based on previosly selected categories.

Image on the right displays the suggested locations for selected categories. Here River is selected as a category, so Columbia River, Colorado River are fetched as a result.

On clicking on any suggestion, user gets directed to an interface with 2 major objects. One is displaying map location of that Suggestion. Second is details of that Suggestion.

Image on the Left is an expanded version when user clicks on the location object. Here user can perform 3 actions

1. Freely explore nearby areas of that location.
2. Get the pointer back to the default location of that place ( by clicking on the pointer on top right corner of the screen).
3. Navigate back to the previous menu of Suggestion Selection list.

Image on the Right is an expanded version when user clicks on the details object. Here user can review various features about this place. For example its description, tags associated with this place and location’s exact address.

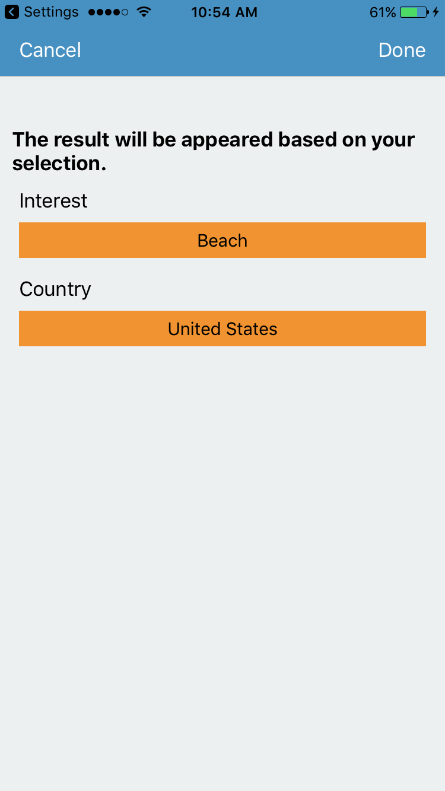
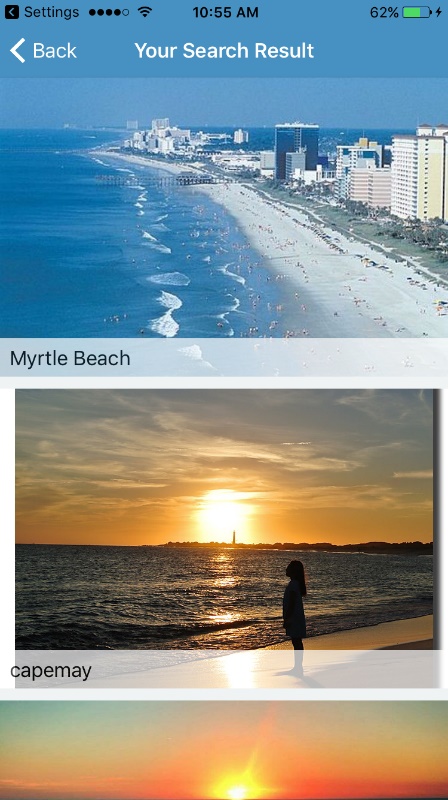
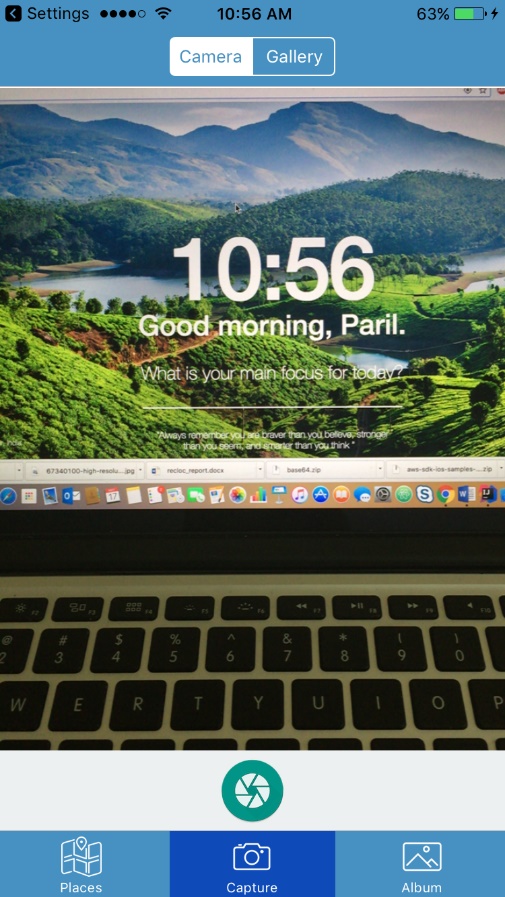
 

Image on the left shows the result of the selected options by the user where user has selected Beach as an Interest and United States as a Country.

When user clicks ‘Done’, related beach search results are provided for that specific country. (Image shown on the right).



Using camera is another cool feature of this App where user can click photos by using moving over to the camera tab on the center of the bottom pane. This helps user to take phots without opening camera in a traditional way. Integrating camera provides capturing photos without leaving the App.

Image on the right shows the screen after user has taken the photo using its camera, where they can go to edit mode for editing that image.

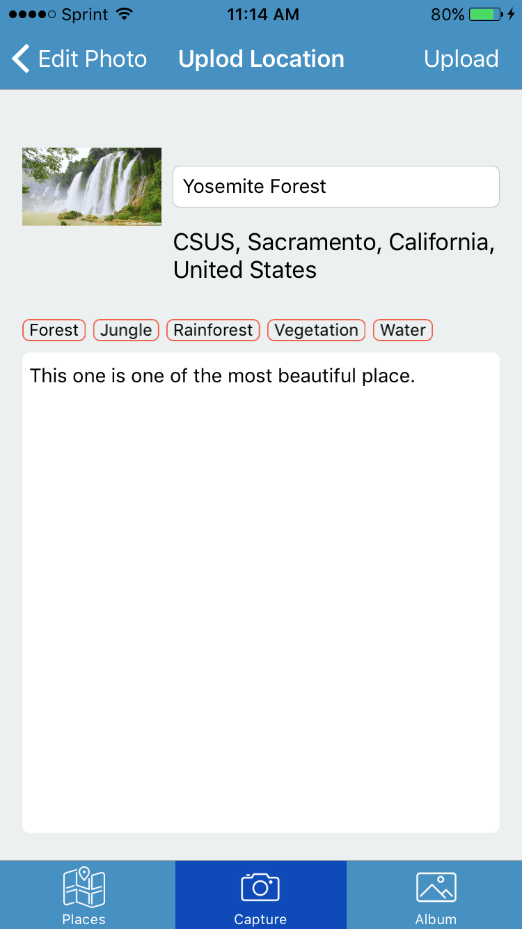
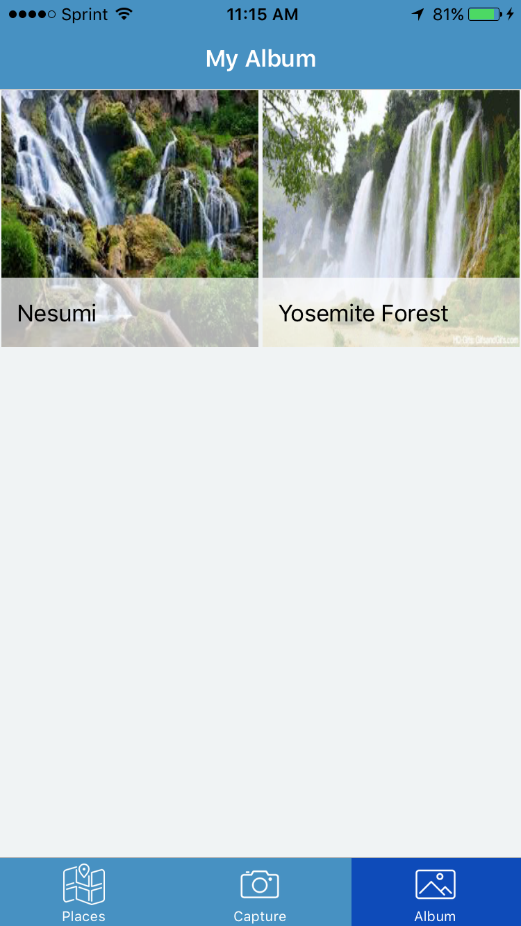


Image on the left displays the next phase after the image is ready to be uploaded. Here user can add description to that image and location.

Image on the right displays the screen of uploaded images. After selecting upload in previous interface, user can check its already uploaded images in ‘Albums’ tab at the right of the bottom pane. Here user can view images along with the location.

**New Technologies Learned**

- Spring/Hibernate [For creating backend]

- AWS EC2 setup [For Deployment]

- Amazon Relational Database Services (RDS) [Database server]

- AWS Rekognition SDK [For retrieving objects from the images using Deep learning]

**Issues**

While Installing Tomcat Server on virtual machine, there were many problems in accessing permission issues because of which setup of Tomcat server took time longer than expected. This happened during the manual installation of Tomcat server. Using Tomcat server as a Service and then installing solved the problem.

**Existing Traps**

**1.) Issue:** If 2 users add the same location, then 2 different entries are displayed to other users who try to access it based on location. This happens due to 2 different GPS location coordinates of 2 different users present in the same tourist place categorized.

**Possible Solution:** Merging all user entries which fall under specific predefined region of the same location.

**2.) Issue:** If users are adding image from Photo Album/Gallery of device, sometimes it is possible that the image does not has latitude and longitude. In that case, we are assigning default location of CSUS.

**Possible Solution:** In the future enhancement, we can give an option to user for adding location manually.

**Conclusion**

This Project helped us understand basic inside-outs of RESTful services, Amazon AWS technologies along with basics of AWS Rekogntion and EC2 services. Also, it provides useful solution to user to find location based on users’ own interest and suggestions posted by other users..

**References**

[1] <https://aws.amazon.com/rekognition/>

[2] <https://aws.amazon.com/ec2/>

[3] <https://aws.amazon.com/rds/>