

CS 4366: Senior Capstone Project  
Dr. Sunho Lim  
Project #3 - Software Design Specification - Project Report  
EmergenSeek

Suhas Bacchu Derek Fritz Kevon Manahan Annie Vo Simon Woldemichael

February 27, 2019

**Abstract**

In this report, we describe and detail the layout and design of EmergenSeek. The structure of the system, at a high-level, consists of two components; a mobile client, developed using a Dart-based, cross-platform mobile framework called Flutter, and a backend API, developed using Amazon Web Services and the Go programming language. Each of these components is also broken down into their own modules and components. Respective design documentation, including UML class, use case, and sequence diagrams, are included as well.

## 1 Design Documentation

In this section, we will discuss the various details of our design documents.

### 1.1 UML Class Diagram

### 1.2 UML Use-case Diagram

### 1.3 UML Sequence Diagrams

## 2 Client Analysis

In this section, we will enumerate the various modules and components responsible for keeping the frontend functional.

## 3 Backend Analysis

In this section, we will enumerate the various modules, components, and objects responsible for keeping the backend functional. It should be noted that there is no literal `class` instance within the Go programming language, but objects may still be enumerated through the `struct` qualifier. Class attributes are defined as `struct` fields.

### 3.1 Models

The models of this application contain Go structures which replicate

### 3.2 Subsystems

The backend of the application will utilize AWS Lambda, a serverless, Functions-as-a-Service offering to run our Go code, DynamoDB to store user data and information, the Twilio API to provide the system with programmable SMS and voice communications, and Google's Geocoding and Maps APIs to better define the location of users and various emergency services. The subsystems of the application are divided up into single-responsibility, service-oriented architecture. This means that the backend consists of several Lambda functions and each function is responsible for only one thing. In total, at the time of this report, there are *six* Lambda functions. Here is a breakdown of their functionality. Functions are prefixed with `ES` and suffixed with a meaningful name which explains their role.

- 3.2.1 Lambda Function 1
- 3.2.2 Lambda Function 2
- 3.2.3 Lambda Function 3
- 3.2.4 Lambda Function 4
- 3.2.5 Lambda Function 5
- 3.2.6 Lambda Function 6

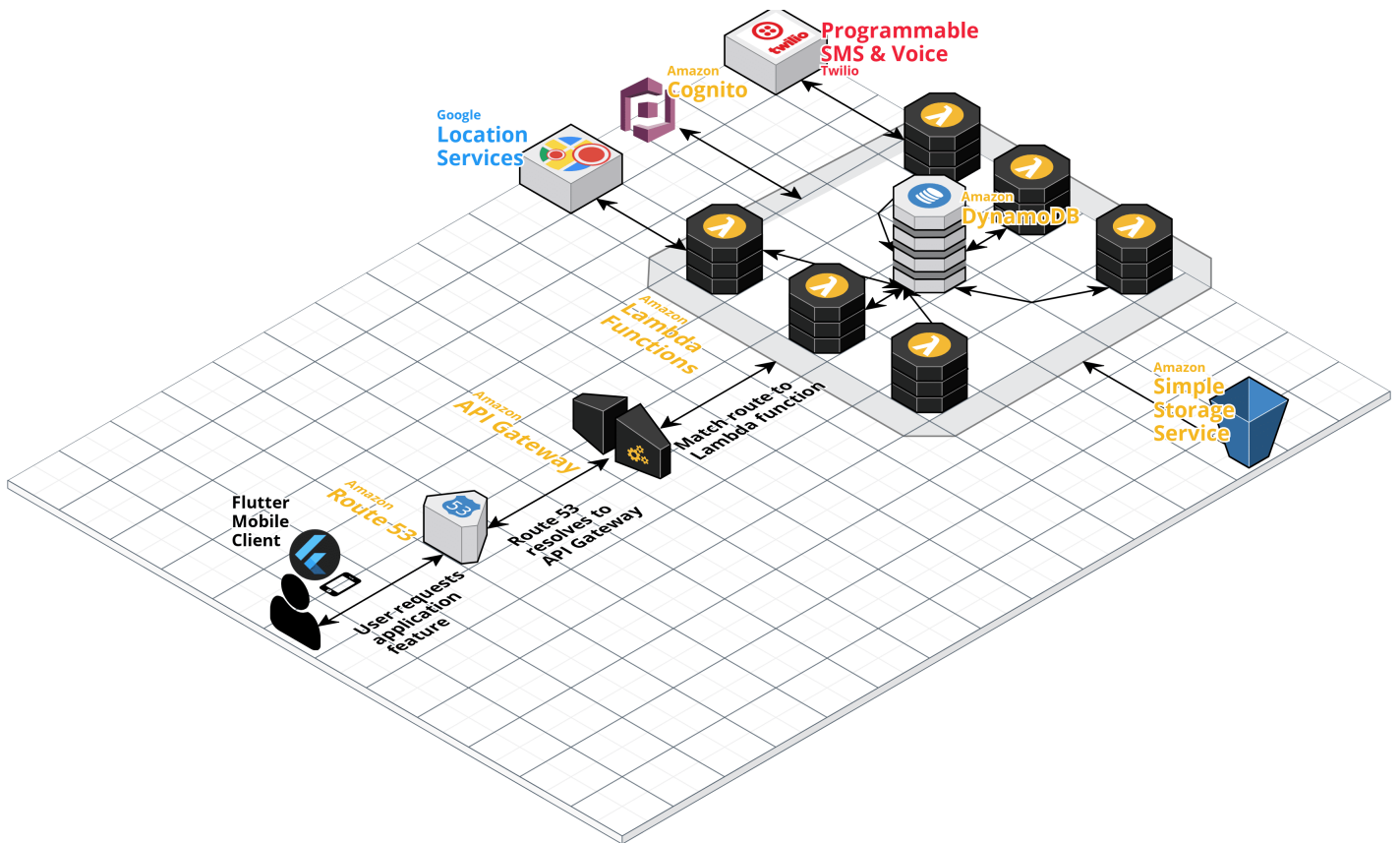


Figure 1: Cloudcraft [1] diagram of the AWS resources and external APIs necessary for the backend.

## References

- [1] Cloudcraft <https://cloudcraft.co>