An IoT based solution to device tracking

By: Dhruv Gupta

Advisor: Anu Bourgeois

Presented: 4/20/2020

Use Case

Software System to enable students to check if students are in their office

Overview

Architecture

Implementation

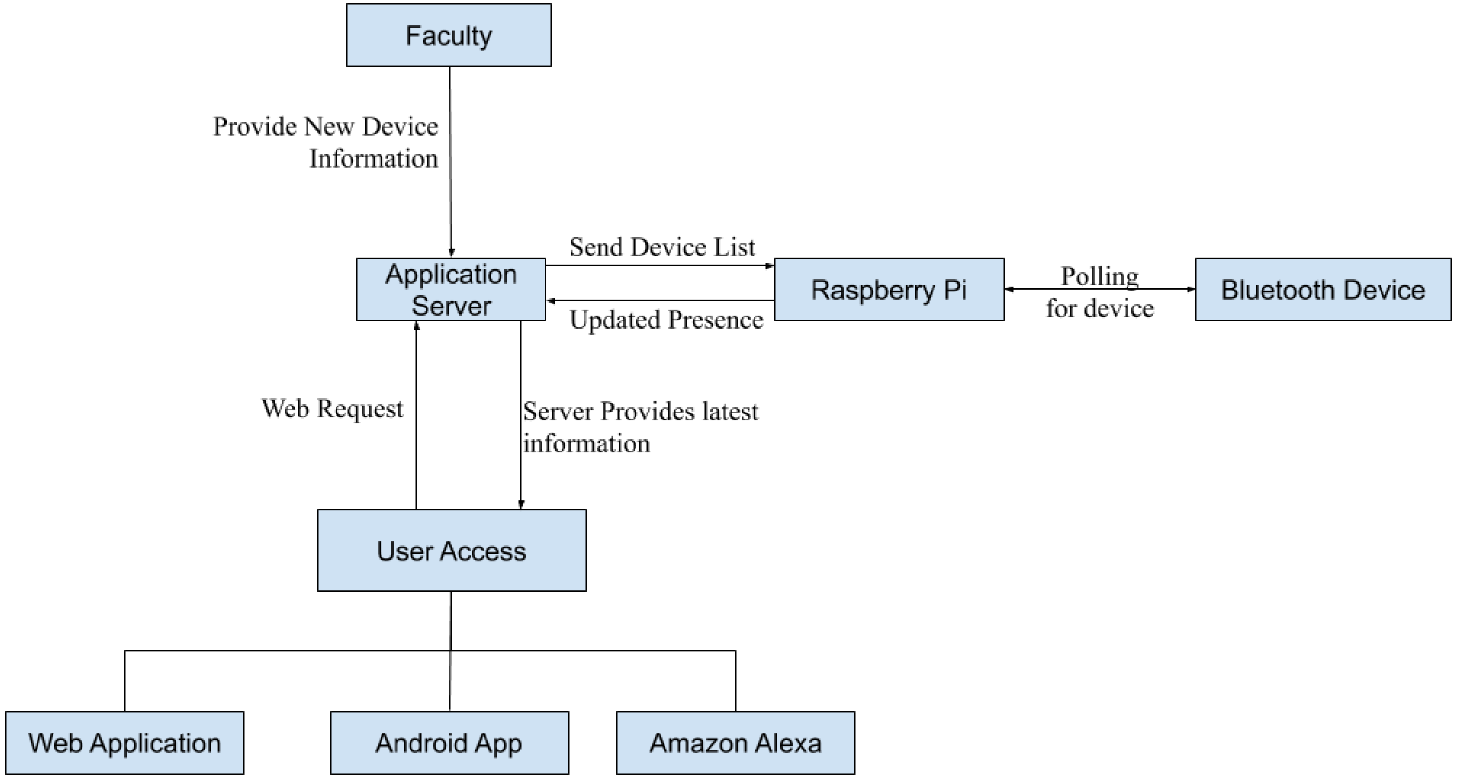
- Raspberry Pi
- Web App
- Web API
- Amazon Alexa
- Android Application

Future Work

Undergrad Projects

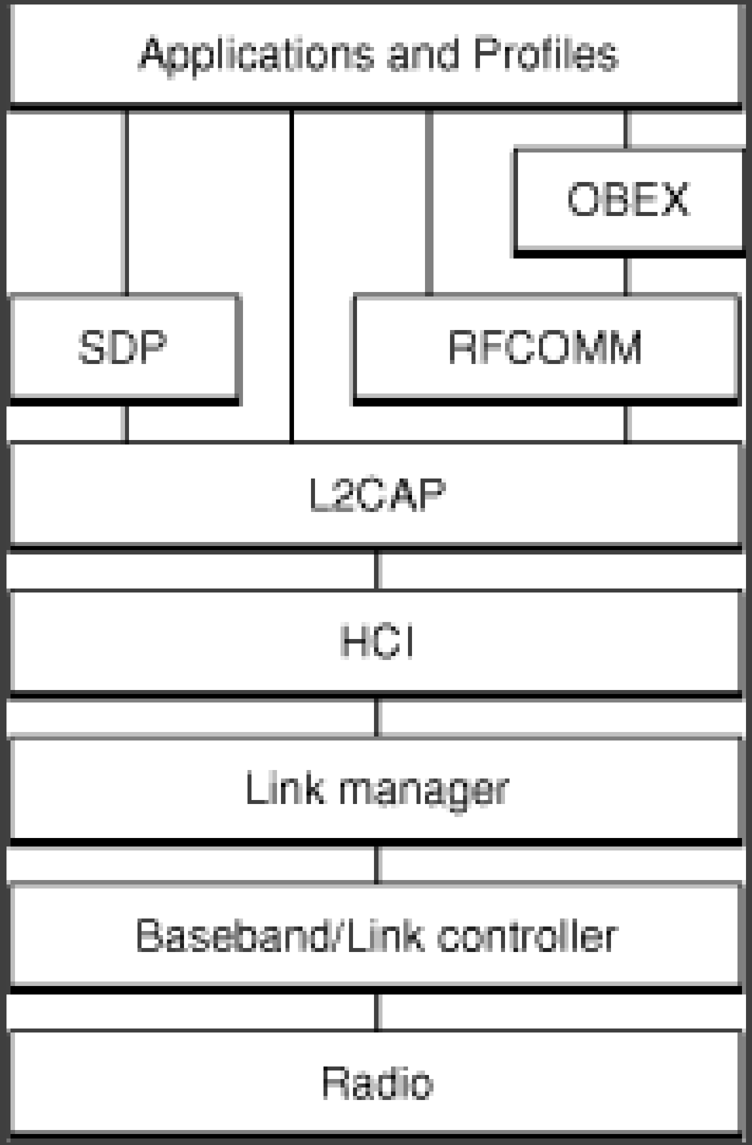
DEMO

Architecture



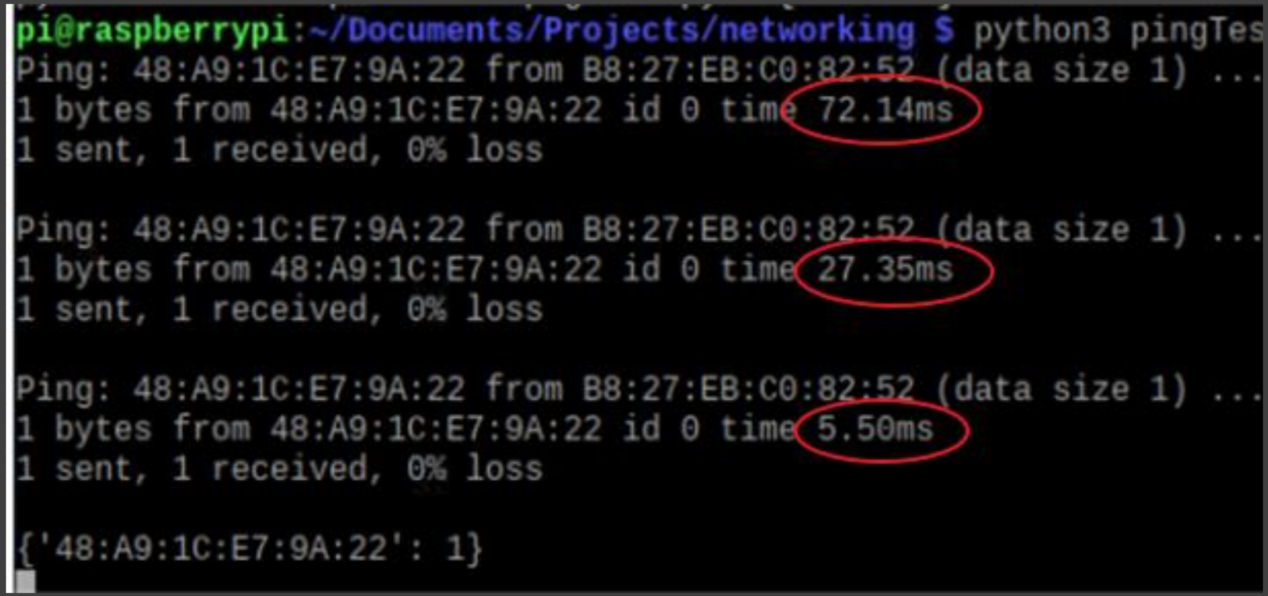
Bluetooth Protocol Stack

- The RFCOMM protocol has the same service and reliability guarantees as TCP
- L2CAP employs the concept of channels to track the source and destination of data packets. The L2CAP layer is a required part of every Bluetooth system. Responsible for providing connectionless and connection oriented services to the upper layers of Bluetooth communication stack
- Every Bluetooth device has a unique identifier similar to the Media Access Control (MAC) address in the TCP/IP paradigm known as the device address or Bluetooth Address. This is a globally unique 48-bit address and the address spaces are managed by the IEEE Registration Authority.
- The L2ping is similar to the ICMP ping in the internet protocol suite



Raspberry Pi

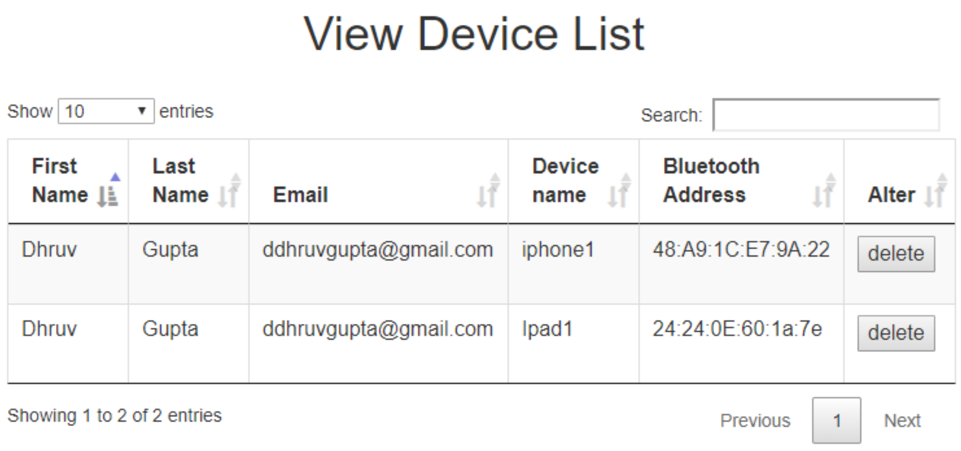
- The Raspberry Pi acts as the sensor which will connect to the Bluetooth devices.
- Since the Raspberry Pi is small, affordable and runs on a Linux operating system distribution called Raspbian, it can be configured to provide high reliability and availability as a distributed sensor bed.
- The Pi runs a python script at a configured frequency to ping all the devices in its device list and updates their presence in a hash table.
- The ping is performed thrice every cycle, with the mode of the results being taken as the result to provide accurate results.



```
pi@raspberrypi:~/Documents/Projects/networking $ python3 pingTes
Ping: 48:A9:1C:E7:9A:22 from B8:27:EB:C0:82:52 (data size 1) ...
1 bytes from 48:A9:1C:E7:9A:22 id 0 time 72.14ms
1 sent, 1 received, 0% loss

Ping: 48:A9:1C:E7:9A:22 from B8:27:EB:C0:82:52 (data size 1) ...
1 bytes from 48:A9:1C:E7:9A:22 id 0 time 27.35ms
1 sent, 1 received, 0% loss

Ping: 48:A9:1C:E7:9A:22 from B8:27:EB:C0:82:52 (data size 1) ...
1 bytes from 48:A9:1C:E7:9A:22 id 0 time 5.50ms
1 sent, 1 received, 0% loss

{'48:A9:1C:E7:9A:22': 1}
```

Web Application

- Built in PHP with jQuery, BootStrap and mySQL database
- Passwords hashed with MD5
- Email verification for user authentication
- Only @gsu or @student.gsu allowed
- PDO for preventing SQL injection
- Search functionality via DataTables
- Hosted on Amazon EC2 instance
- APIs for interacting with other applications



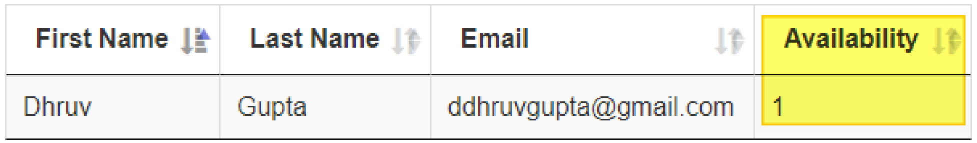
View Device List

Show 10 entries

Search:

First Name	Last Name	Email	Device name	Bluetooth Address	Alter
Dhruv	Gupta	ddhruvgupta@gmail.com	iphone1	48:A9:1C:E7:9A:22	delete
Dhruv	Gupta	ddhruvgupta@gmail.com	Ipad1	24:24:0E:60:1a:7e	delete

Showing 1 to 2 of 2 entries



First Name	Last Name	Email	Availability
Dhruv	Gupta	ddhruvgupta@gmail.com	1

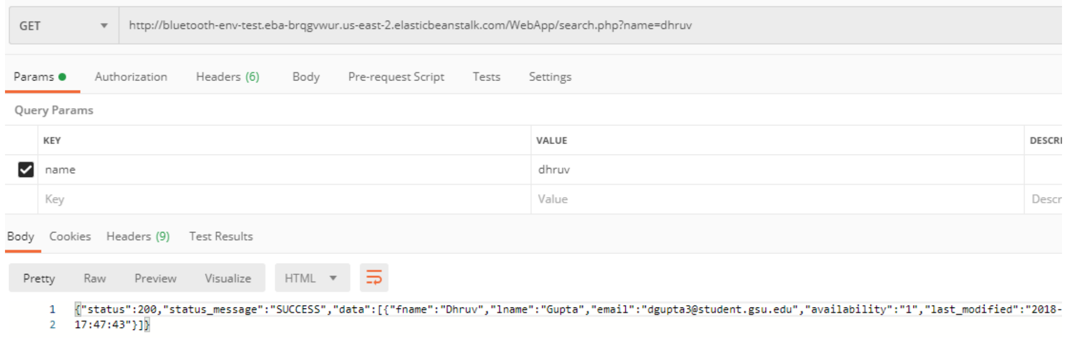
Web API

Search API

- Serves requests from a mobile application and Alexa devices.

Update API

- API for Raspberry Pi devices to update information about devices and get an updated list of devices.

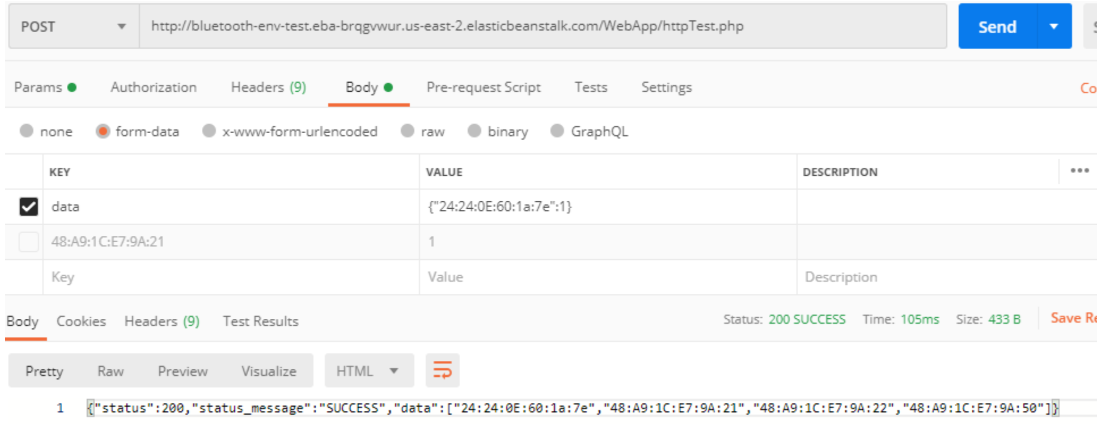


GET http://bluetooth-env-test.eba-brqgvwur.us-east-2.elasticbeanstalk.com/WebApp/search.php?name=dhruv

Query Params

	KEY	VALUE	DESCRI
☑	name	dhruv	
	Key	Value	Descr

```
{"status":200,"status_message":"SUCCESS","data":[{"fname":"Dhruv","lname":"Gupta","email":"dgupta3@student.gsu.edu","availability":"1","last_modified":"2018-
17:47:43"}]}
```



POST http://bluetooth-env-test.eba-brqgvwur.us-east-2.elasticbeanstalk.com/WebApp/httpTest.php

○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

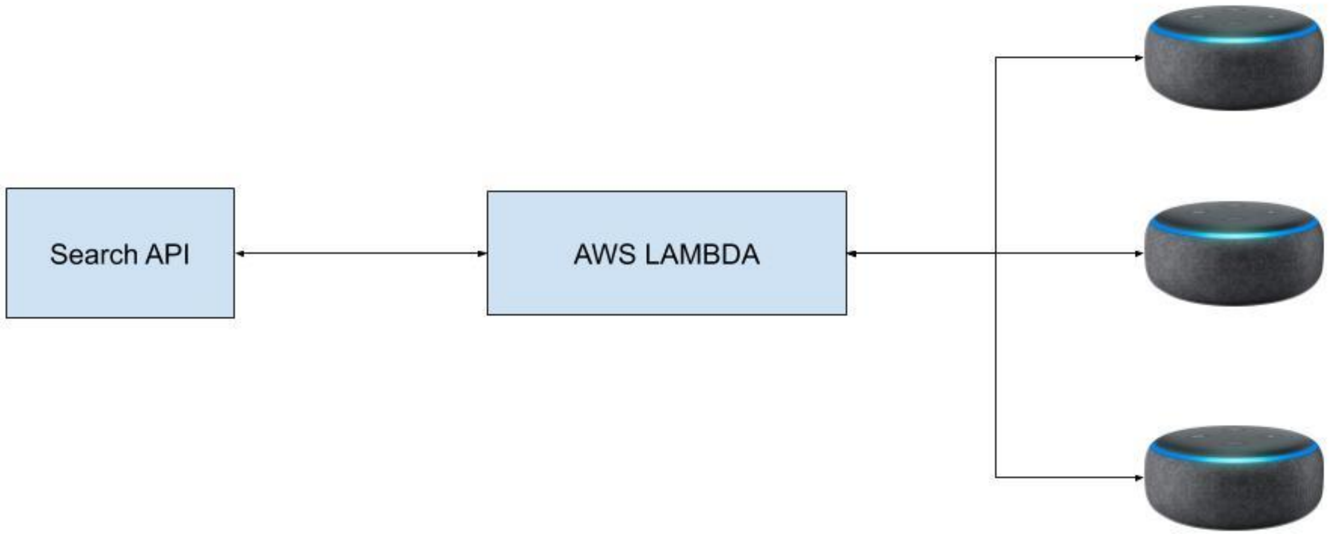
	KEY	VALUE	DESCRIPTION
☑	data	{"24:24:0E:60:1a:7e":1}	
□	48:A9:1C:E7:9A:21	1	
	Key	Value	Description

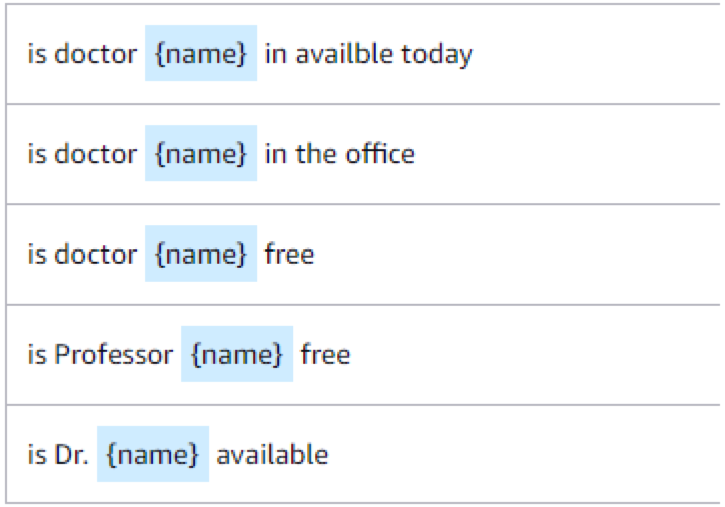
Status: 200 SUCCESS Time: 105ms Size: 433 B

```
{"status":200,"status_message":"SUCCESS","data":["24:24:0E:60:1a:7e","48:A9:1C:E7:9A:21","48:A9:1C:E7:9A:22","48:A9:1C:E7:9A:50"]}
```

Amazon Alexa

- AWS Lambda function created to act as parent node for Alexa requests
- Developed in Node.js, contains event handlers for different interactions and inputs
- Alexa is pre trained to be able to understand names without developer support
- Custom name class can be created but would require constant updates with faculty changes





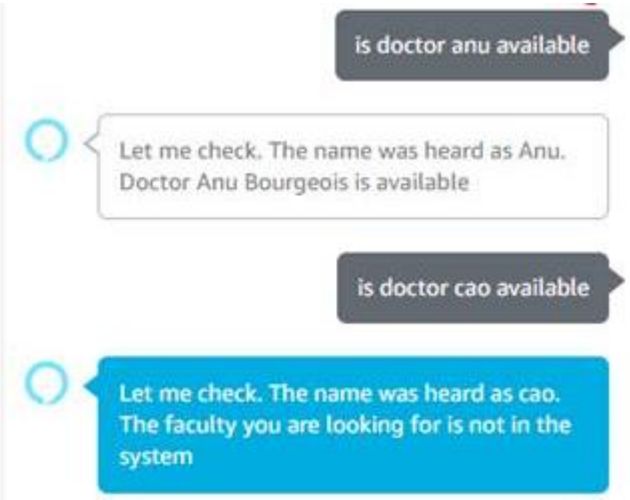
is doctor {name} in availble today

is doctor {name} in the office

is doctor {name} free

is Professor {name} free

is Dr. {name} available



is doctor anu available

Let me check. The name was heard as Anu. Doctor Anu Bourgeois is available

is doctor cao available

Let me check. The name was heard as cao. The faculty you are looking for is not in the system

Amazon Web Services

Solution hosted on Amazon EC2 instances using Elastic Beanstalk

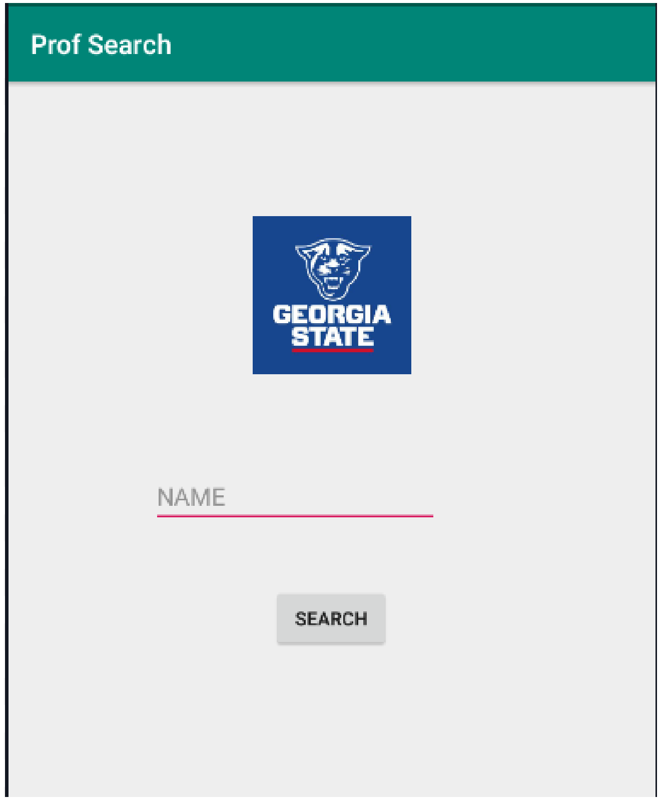
EC2 instance linked to GitHub Repository via CI/CD Pipeline

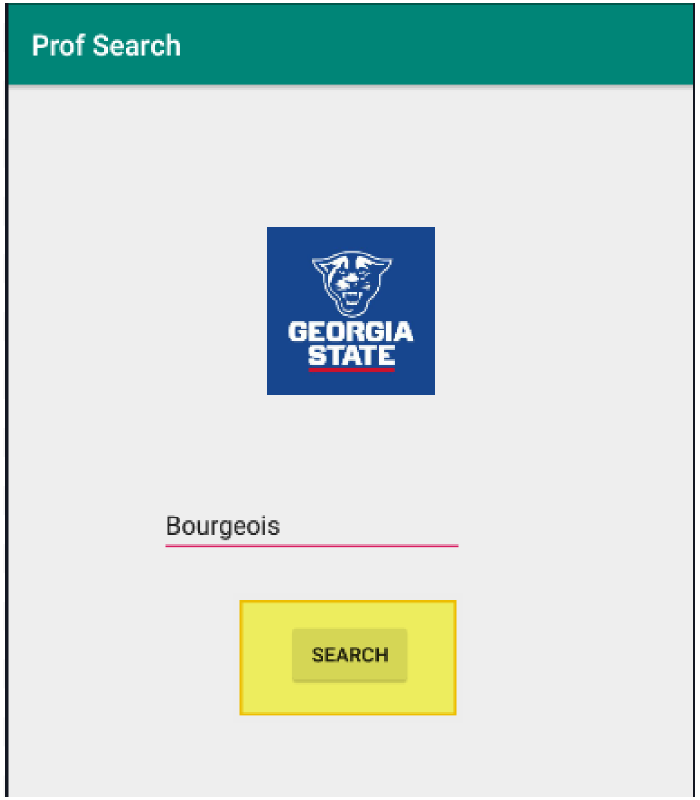
Internal communication between services can be secured

Eg. Lambda function can communicate with EC2 instance securely

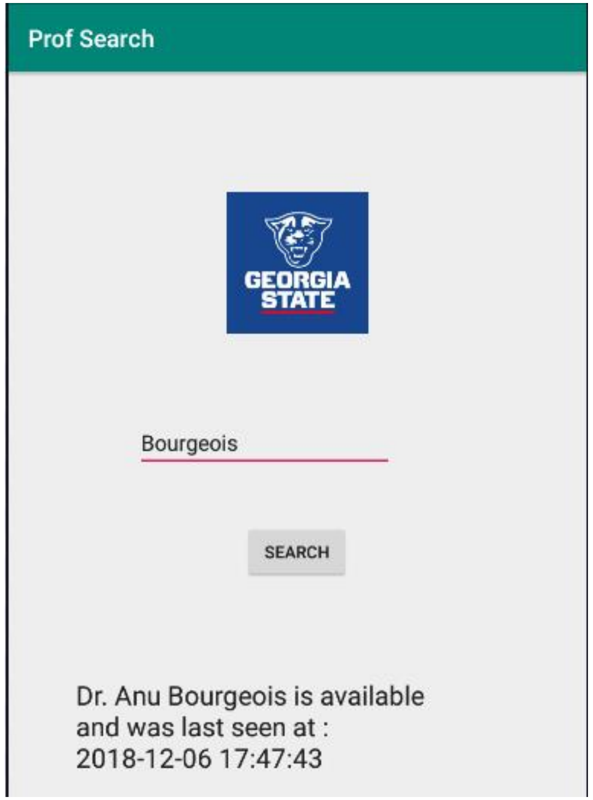
mySQL RDS instance is secured and can only be reached via EC2 server

Android Application





Bourgeois



Bourgeois

Dr. Anu Bourgeois is available
and was last seen at :
2018-12-06 17:47:43

Future Work

Improve Landing Page and add NavBar

Add OAuth 2.0 to API

Implement SSL communication between all components

Implement Batch Processing Queuing system to enable scaling of Sensor Network

Implement Caching mechanism for preventing too many calls to database

Database Sharding based on department

Mobile application to enable professors to update availability or via Alexa

Appointment making ability via Alexa

Projects for undergraduates

Build an Alexa based Client using available APIs

- Good OS background knowledge required
- Eg. Ask Alexa to find Campus Dining Options

Build an Android Application

- Requires background in OS and Mobile Apps
- Eg. Building an application to trigger home automation systems

Raspberry Pi Application for sensing

- Python and other tools easily available on Linux, Great community support
- Good introduction to Linux OS, Networking and working in resources constraints
- Sensors can interface with GPIO pins
- Eg. Home Automation Solutions, Health Tracking, Controlling Media Devices

Database Concepts

- Explore how to store real time data and database scaling concepts
- Good introduction for indexing and database partitioning concepts

DEMO