Project Progress Report

Student: Dazhi Li

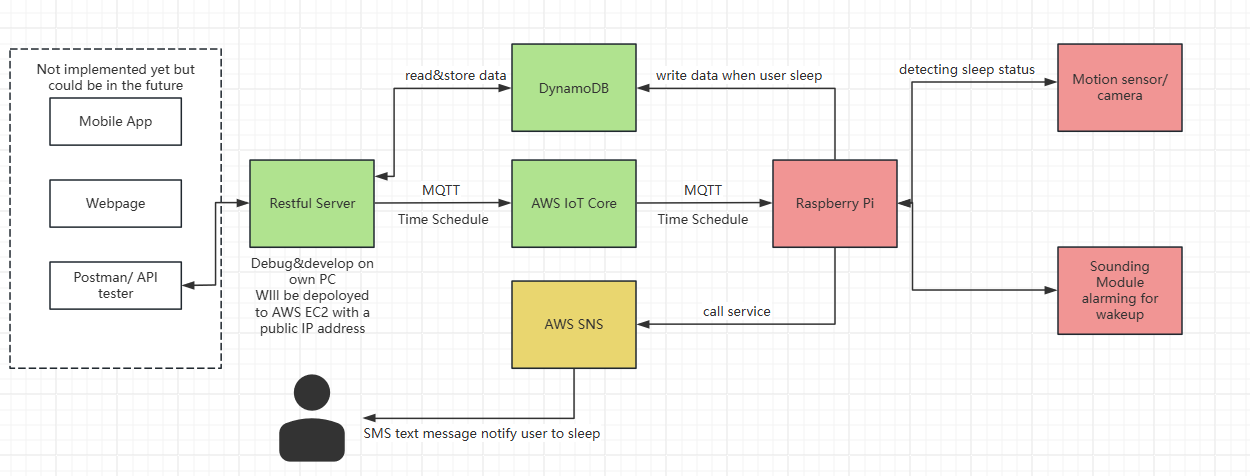
NetID: dazhili  
Student Number: 2400330

Project Name: Sleep habit forming alarm

# 1 Completed Tasks/Implemented Features

## 1.1 Workflow adjustment and details

I adjusted the whole work flow and added more details on how things will be implmented.



The figure above shows my new schema for my final project. On the left of the program is client-side application which is not included in this project. Just for references on further improving Human Computer Interface experience**. All modules marked green are those totally finished currently. Yellow stands for what are pending due to some situations. Red means not started yet.**

## 1.2 Restful Server

Restful server was implemented by Python, Flask and Flask Restful libraries. It currently supports interfaces below.

GET /alarms

Query all the alarms in the database

POST /alarms

Create a single alarm, body is json form. This request will generate a new alarm which is stored in the database. With the information that user provides in the request and previous user behaviors in the database, it will automatically generate a new time schedule. This schedule will be sent to IoT core by MQTT protocol.

GET /alarms/{alarm\_id}

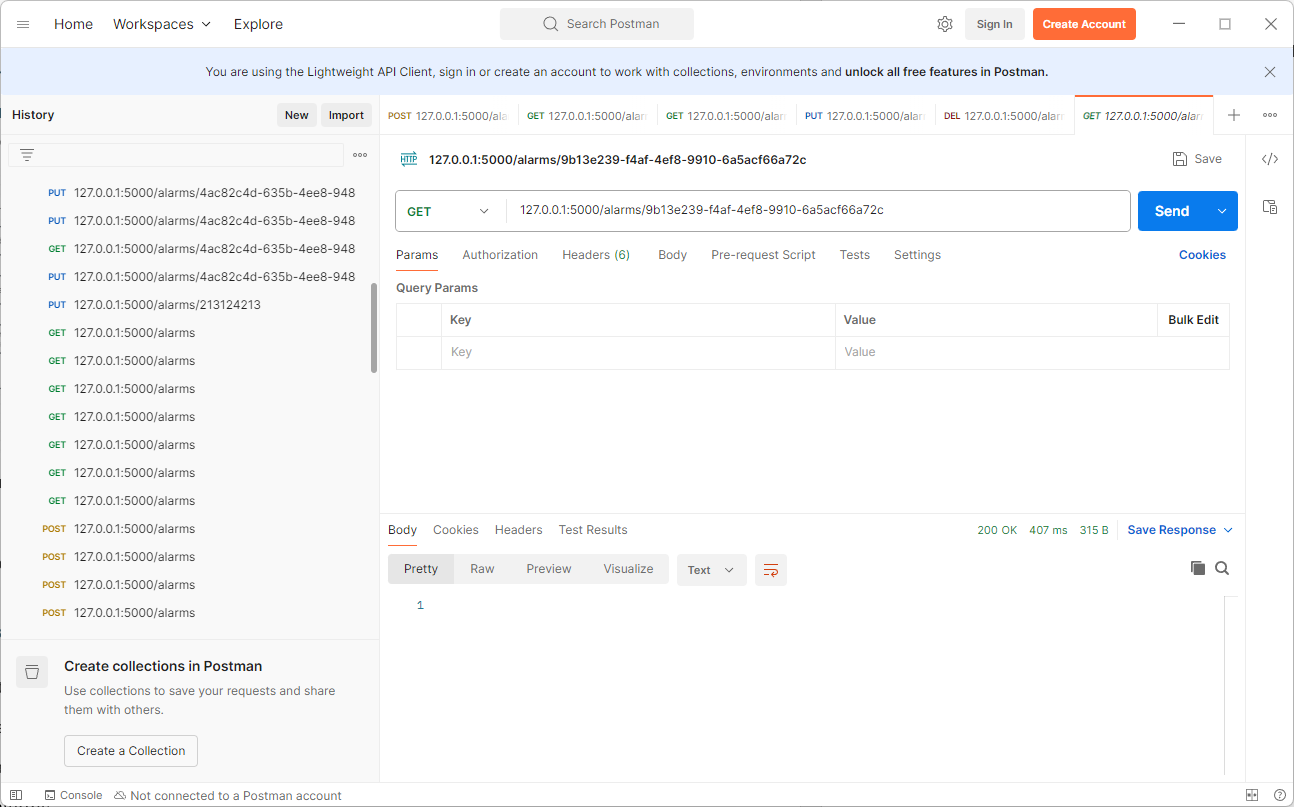
Query single alarm by alarm\_id, which is informed when alarm is created

DELETE /alarms/{alarm\_id}

Delete single alarm by alarm\_id. This will also send notification to IOT Core to cancel a time schedule.

PUT /alarms/{alarm\_id}

For simplicity, this interface was not fully implemented, user have to delete the alarm that user wish to modify and create a new one instead.



## 1.3 AWS Dynamo DB

This part will show how tables were arranged in DynamoDB

ALARMS

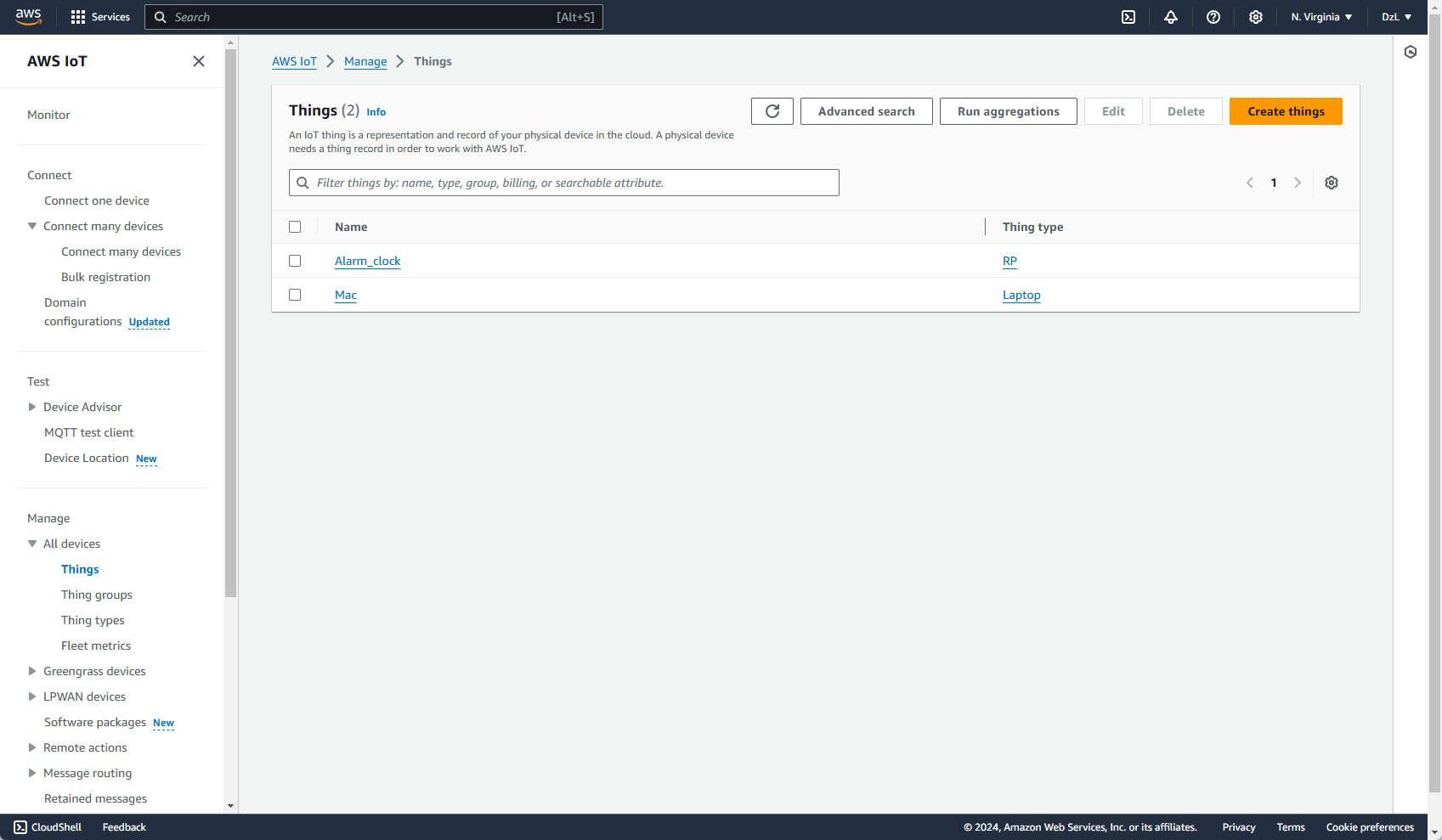
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID (Primary Key) | TIME (wake up time) | PHONE (notification destination) | REPEAT (only support daily repeat) | PREFER\_SLEEP\_TIME (user’s preference for sleep length) |

USER\_BEHABIORS

|  |  |  |
| --- | --- | --- |
| DATE (PK) | PRESET\_SLEEP\_TIME (preset sleep time calculated in time schedule) | ACTUAL\_SLEEP\_TIME  (real sleep time reported by alarm clock) |

## 1.4 AWS IoT Core

This part is quite simple as what I did in homework1. I added 2 devices in my AWS IoT core.

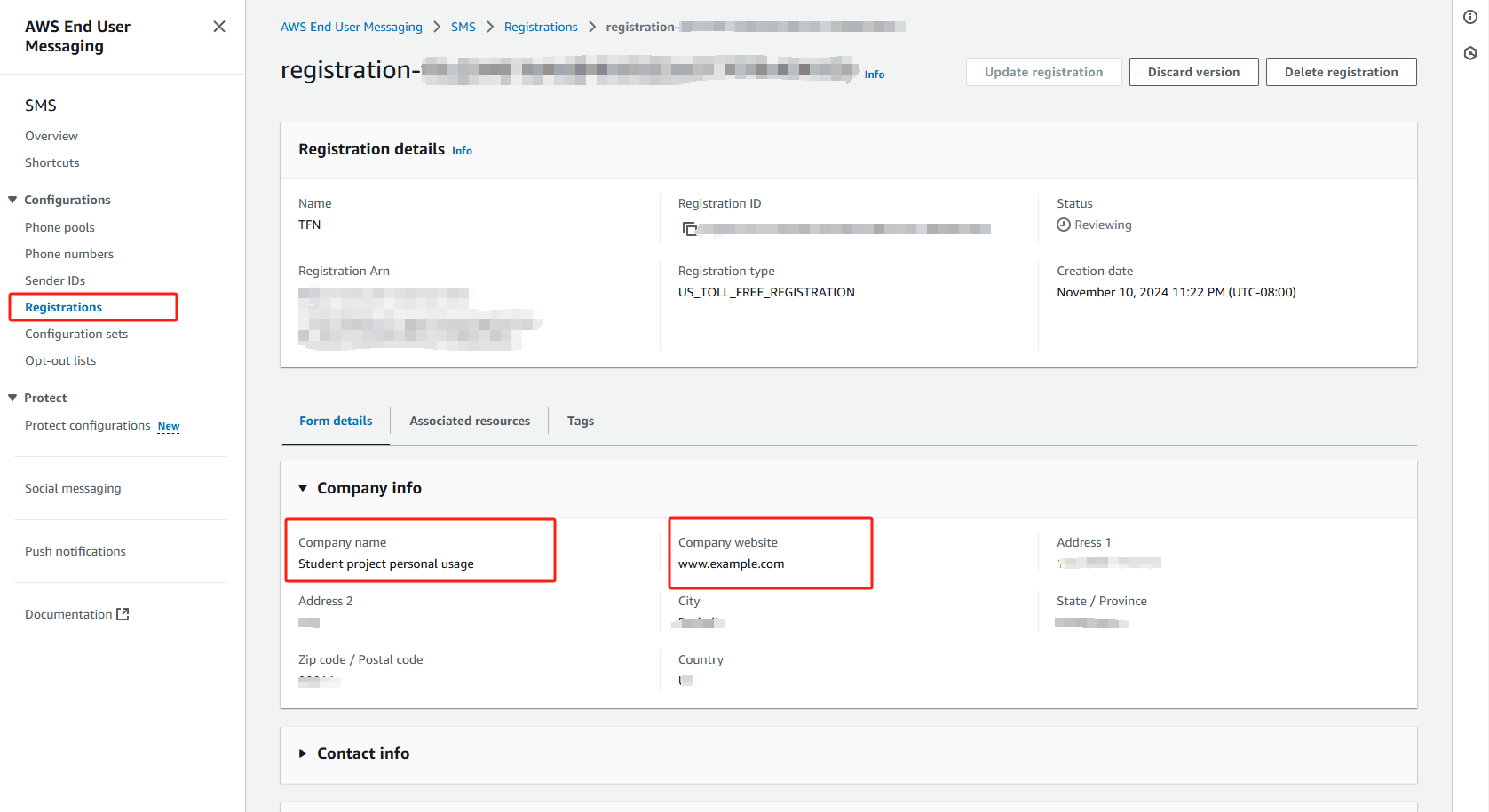


# 2 Encountered Problems

## 2.1 SNS Service needs an exclusive originator number

As far as I know, recently AWS added some regulations on SNS services that all messages sent from AWS need an originator/sender number which could identify the sender. However, in the past this is not necessary as AWS provided shared phone number as originator number before.

The problem is, requesting an originator number require a company name and company address so that I could use SNS service. However, I did not have any company registered under my name, all I could is sending a request to indicate that I only use this originator number for personal class project. I am not sure whether it will pass or not. If not, I have to consider another way that user could receive message from my raspberry pi.



## 2.2 Should I use raspberry pi as my alarm clock

As what we have discussed, raspberry pi is not the only choice to act an alarm clock in my final project. I considered and did a lot research on the webs. I found that using apple watch or any other forms of mobile devices is quite hard for me to program on those devices. Still, I am looking for better choices which could run 24\*7 and is easy programmable.

# 3 Planned Tasks

## 3.1 Coding another program running on alarm clock

The whole system contains two main programs, the first one is the restful server which has already been achieved. The second program is running on my alarm clock (currently design is raspberry pi), which could subscribe a time schedule from my IoT Core. Also, this program is responsible for storing user behaviors to the database.

## 3.2 Detecting user’s sleep behavior

My original plan is using a camera to automatically monitor the sleep time of a user. However, I might change my strategy to using a motion sensor/button instead. A button will be preferred because I think a motion sensor will too vague for my alarm clock to decide when is the precise time that the user sleep.