2017

FIT3140 Assignment 3

REPORT

DEKEL PILLI & JAMES PICKERING

Table of Contents

Spike specifications

Our work consisted of two separate spikes, as described below.

Non-technical explanations

Spike 1

The first involved making a program that notifies an online database when it detects a set of motions, and a program that sends an email to a pre-defined address with details about the latest motion and about the motions received so far in general. If both of these programs are running at once, every set of motions that is detected should lead to at least one email being sent.

Spike 2

Our second spike involved us replacing the email-sending program that must be active on a local device to work, with one that runs on the cloud, independently of a local machine. This will perform the same function, but it means that only the motion sensor needs to be running to achieve the same functionality.

Technical explanations

Spike 1

Our first spike included a program that reads from an Arduino Uno motion sensor. On receiving a motion, the program should record the time of its start and its end, and send that data to a Firebase Database. The second program will read this data whenever the firebase is updated and send an email with information about all of the data in the firebase, and another email if the difference between the two times is at least 5 seconds.

Spike 2

For the second spike, the first program remains unchanged. The second program, instead of having to run locally, will be changed to be server-less. This will use Firebase Functions to listen for any additions to the Firebase Database.

Spike review

While the tasks for both spikes have been completed, both were done a day later than the deadline. While this is not desirable, we did plan the deadline for these spikes with the knowledge that being late is a possibility, and left enough leeway to still complete the assignment in time, without much stress. Due to past assignments, individual works, and lab exercises, we were both familiar with the toolchain used (Arduino, Node JS, Firebase DB & Firebase Functions), so we did not need to practice for the assignment.

These spikes were planned in close consideration of the requirements, with the first spike allocated for completing all of our on-promise functionality, and the second spike dedicated to converting the necessary parts to be server-less. This worked well in practice.

Alternatives

While there are many motion detection devices available, alternatives for the Arduino were not explored as it is the only one we have free access to.

While Node JS is a popular option for Arduino communication, Python would have also been a suitable option. Both team members are comfortable using Python and Python is also able to handle all Firebase Database functions well. Java also fit all of these criteria. Despite these options, our decision to use Node JS was easy, mostly due to the fact that it was the language we used for

assignment 2, which also involved Arduino and Firebase. This allowed us to guarantee our proficiency with the interaction between these tools and reuse some of our old code.

While there exist alternatives to Firebase DB (other cloud DBs) and Firebase Functions (Azure functions, AWS Lambda, both of which support Node JS), the specifications of our project required us to use these tools. Additionally, our team's experience with Firebase makes it a good choice, in addition to its usefulness and powerful support for listeners. For these reasons, despite both team members being experienced with Dynamo DB and SQL/SQLite, we had no doubt that firebase is the ideal solution for us.
In regards to a server-less computing utility, Firebase Functions has the advantage of being on the same platform as Firebase DB. While AWS Lambda might be a strong candidate in a professional setting, its lack of free options ruled it out. Microsoft Azure, though, has enough free features to support completion of this assignment, but not for the end of the semester, meaning that if future assignments rely on this one, we will be required to eventually refactor our work to deal with a new server-less computing option. Firebase Functions is free and supports all of our needs, while also not requiring any new accounts as it is integrated into the same platform that provides our database.

Appendix A

Spike plans

Spike 1 plan

Name:

On-promise motion sensor email trigger, using Firebase

Context:

In this spike, we will implement a local program that reads data from a motion sensor and saves it to Firebase, and then another local program that listens to Firebase, and sends emails containing motion sensor data.

Gap:

Testing the viability of an on-promise implementation for this app

Goals/Deliverables:

The deliverables of this spike are:

- A program that writes to firebase based on arduino data
- A program that sends emails based on firebase data

Some of the risks that could be involved in this spike include:

- Underestimation of the time required to complete the task. To mitigate this risk, we have allocated enough time for the rest of the spikes, so that even if this one is slightly off, we can still be on track.
- A miscommunication causes us to use a different standard for the firebase communication. If this happens, we hope to be able to catch that during the merging process (or earlier) and fix it. Fixing it should not be too difficult if we employ good coding conventions.

Planned start date: 07/September/2017
Deadline: 8/September/2017

Planning notes:

First, we will define a firebase format that we will use to write and read. Then, we will divide the work (email section and data reading section), finish them separately and finally merge them and test our work.

Spike 2 plan

Name:

Server-less motion sensor email trigger, using Firebase

Context:

In this spike, we will implement a program running on Firebase's servers that listens to Firebase database, and sends emails containing motion sensor data.

Gap:
Testing the viability of a server-less implementation for this app

Goals/Deliverables:
Server-less functions for the following:

- Sends email to a pre-defined address if the motion is long
- Sends another email to a pre-defined address shows how many long and short motions have been detected.

This includes basic testing of functionality

Planned start date: 08/September/2017
Deadline: 10/September/2017

Planning notes:
Before implementing the server-less programs, we will need to agree that our work from Spike 1 was satisfactory.