

SPIKE PLAN (SPIKE-1)

Name: Getting data from motion sensor and processing it on local server.

Context:

1. Every second it is checked if the motion sensor has sent any motion.
2. If motion is detected, motion information is sent to the database. Motion information consists of motion start time, motion end time, length of the motion and the current time.
3. As soon as data is added to firebase, the listener on the local server is called with a snapshot of the database sent to the listener.
4. Then first it is checked if the motion is in the range $>T1$ AND $<T2$ (where $T1$ and $T2$ are predefined thresholds). If yes, then an email is sent to a predefined address with the motion information using the nodemailer library.
5. Then it is checked if the last 5 motions are available from the snapshot. If yes, then information about those last 5 motions is sent using the nodemailer library.
6. At last it is checked if motion length is $> T2$. If yes, then all the records in the database is deleted.

Gap:

Risks the spike is being conducted to mitigate

- General risks associated with both spikes
 1. The motion sensor is not detecting motion due to it being damaged or faulty.
 2. Motion information is not being added to the firebase instantly.
 3. Firebase is not being able to send snapshot of the database to the listeners in the local server.
 4. Bonescript versions is deprecating and not being supported by beaglebone black.
- Specific risks associated with SPIKE-1
 1. Scaling local server becomes too costly and expertise support needed.
 2. Lack of separation of concerns as back-end code and front-end code getting mixed up.
 3. Maintenance becoming time-consuming and costly.

Goals/Deliverables:

What you want to come out of this spike

1. Motion sensor is detecting motion correctly and is not faulty.
2. Motion information is being added to the firebase instantly.
3. Firebase is correctly send the required snapshot of the database to the listeners in the local server.
4. Bonescript version are not deprecating and are being supported by beaglebone black.
5. Scaling local server can be managed under budget without any expertise support.
6. Good libraries can help to separate backend and front-end code to achieve separation of concerns.
7. Maintenance can be done in a timely manner with costing under budget.

Planned Start Date: 14 April 2018

Deadline: 22 April 2018

Planning Notes:

1. One team member installs the library needed for hosting the local server.
2. The other team member installs library for the beaglebone black and for using the firebase.
3. Both team members start working on the code.
4. One team member focuses on processing the data from the motion sensor and pushing to the database.
5. Other team member focuses on processing the data from the database and implementing functionality for sending email and database reset.
6. Both team members test the code to see if everything is working properly.

SPIKE OUTCOMES (SPIKE-1)

Name: Getting data from motion sensor and processing it on local server.

Goals:

1. Motion sensor is detecting motion correctly and is not faulty.
2. Motion information is being added to the firebase instantly.
3. Firebase is correctly send the required snapshot of the database to the listeners in the local server.
4. Bonescript version are not deprecating and are being supported by beaglebone black.
5. Scaling local server can be managed under budget without any expertise support.
6. Good libraries can help to separate backend and front-end code to achieve separation of concerns.
7. Maintenance can be done in a timely manner with costing under budget.

Personnel:

- Team Member 1

Name: Hussain Sadiq Abuwala

- Team Member 2

Name: Tasin Aryan

Technologies, Tools, and Resources used:

- Firebase

Firebase is a full suite for mobile and web application development developed in 2011. It is a platform that helps you quickly develop high-quality apps and grow your user base with no server-side programming. Firebase is made up of complementary features and components such as Analytics, Real-time Database, Cloud Storage and Functions and Hosting to name a few (Firebase, 2017). It helps us to save development time and makes it easier to maintain and support our application.

- Bonescript (Server PIR communication)

BoneScript is a Node.js library specifically optimized for the Beaglebone Black (Beaglebone, 2017). It contains several functions specially created for the Beaglebone. These several functions from the Bonescript library are useful for interacting with the Beaglebone Black. This library simplifies the process of accessing and configuring the pins of the Beaglebone Black. The several functions from Bonescript allow you to control an output, read data from a button/sensor, blink an LED, or even control a motor.

- Node.js and Express

Node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code on the server side. It lets developers use JavaScript for server-side scripting to produce dynamic web page content before the page is sent to the user's web browser.

Express – used as the server for the website. Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. It is used to provide small, robust tooling for HTTP servers, making it a great solution for single page applications, web sites, hybrids, or public HTTP APIs.

- Beaglebone

Beaglebone Black is a low-cost, community-supported development platform for developers and hobbyists

- PIR Motion Sensor

PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range. They are small, inexpensive, low-power, easy to use and don't wear out.

Tasks undertaken:

1. Collect data from motion sensor.
2. Send motion information to database which includes current time, motion start time and motion end time.
3. Send email if motion length is $>T1$ and $<T2$ (where $T1$ and $T2$ are predefined thresholds).
4. Send email about summary of last five motions if available from the database.
5. Reset the database if length of motion is $>T2$ where $T2$ is a predefined threshold.

What we found out:

1. The motion sensor is not very accurate. Sometimes it works properly but sometimes it gives wrong readings.
2. Regarding scaling of the server, we found out that when traffic increases, we need help from experts to scale our server accordingly as it is complicated to do on our own.
3. All other risks that associated with this spike can be mitigated except the ones mentioned above.

Open issues/risks:

We could achieve all the spike goals except we found faults in the motion sensor and the scaling of the servers is not possible to do on our own.

Recommendations:

There could be one more spike to finding out different alternatives for finding out an accurate motion sensor.

SPIKE PLAN (SPIKE-2)

Name: Getting data from motion sensor and processing it on cloud servers.

Context:

1. Every second it is checked if the motion sensor sends any motion.
2. If motion is detected, motion information is sent to the database. Motion information consist of motion start time, motion end time, length of the motion and the current time.
3. As soon as data is added to firebase, firebase server calls the functions stored in their server with snapshots of the database at that time.
4. Then using the first function, it is checked If the motion is in the range $>T1$ AND $<T2$ (where T1 and T2 are predefined thresholds). If yes, then an email is sent to a predefined address with the motion information using the nodemailer library.
5. Then using the second function, it is checked if there are last 5 motions available from the snapshot. If yes, then information about the last 5 motions is sent using the nodemailer library.
6. At last using the third function, it is checked if motion length is $> T2$. If yes, then all the records in the database is deleted.

Gap:

Risks the spike is being conducted to mitigate

- General risks associated with both spikes
 1. The motion sensor is not detecting motion due to it being damaged or faulty.
 2. Motion information is not being added to the firebase instantly.
 3. Firebase is not being able to send snapshot of the database to the listeners in the local server.
 4. Bonescript versions is deprecating and not being supported by beaglebone black.
- Specific risks associated with SPIKE-2
 1. Firebase is not scaling up the app automatically.
 2. Firebase servers is not able to call the functions stored in their serves instantly. There is some latency.
 3. Firebase functions is difficult to learn for the team members due to lack of clear documentation.
 4. Command line tools to manage firebase functions are not being installed.

Goals:**What you want to come out of this spike**

1. Motion sensor is detecting motion correctly and is not faulty.
2. Motion information is being added to the firebase instantly.
3. Firebase is correctly sending the required snapshot of the database to the listeners in the local server.
4. Bonescript version are not deprecating and are being supported by beaglebone black.
5. Firebase is scaling its servers up and down automatically.
6. Firebase functions are called instantly.
7. Good documentation is available to easily learn firebase functions.
8. Command line tools are installed correctly.

Planned Start Date: 14 April 2018

Deadline: 22 April 2018

Planning Notes:

1. One team member installs the library needed for installing the firebase project and the command line tools for deploying functions.
2. The other team member installs library for the beaglebone black and for using the firebase.
3. Both team members start working on the code.
4. One team member focuses on processing the data from the motion sensor and pushing to the database.
5. The other team member focuses on processing the data from the database and implementing functionality for sending email and database reset by implementing firebase functions and deploying on the cloud
6. Both team members test the code to see if everything is working properly.

SPIKE OUTCOMES (SPIKE-2)

Name: Getting data from motion sensor and processing it on cloud servers.

Goals:

1. Motion sensor is detecting motion correctly and is not faulty.
2. Motion information is being added to the firebase instantly.
3. Firebase is correctly send the required snapshot of the database to the listeners in the local server.
4. Bonescript version are not deprecating and are being supported by beaglebone black.
5. Firebase is scaling its servers up and down automatically.
6. Firebase functions are called instantly.
7. Good documentation is available to easily learn firebase functions.
8. Command line tools are installed correctly.

Personnel:

- Team Member 1

Name: Hussain Sadiq Abuwala

- Team Member 2

Name: Tasin Aryan

Technologies, Tools, and Resources used:

- Firebase

Firebase is a full suite for mobile and web application development developed in 2011. It is a platform that helps you quickly develop high-quality apps and grow your user base with no server-side programming. Firebase is made up of complementary features and components such as Analytics, Real-time Database, Cloud Storage and Functions and Hosting to name a few (Firebase, 2017). It helps us to save development time and makes it easier to maintain and support our application.

- Bonescript (Server PIR communication)

BoneScript is a Node.js library specifically optimized for the Beaglebone Black (Beaglebone, 2017). It contains several functions specially created for the Beaglebone. These several functions from the Bonescript library are useful for interacting with the Beaglebone Black. This library simplifies the process of accessing and configuring the pins of the Beaglebone Black. The several functions from Bonescript allow you to control an output, read data from a button/sensor, blink an LED, or even control a motor.

- Node.js and Express

Node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code on the server side. It lets developers use JavaScript for server-side scripting to produce dynamic web page content before the page is sent to the user's web browser.

Express – used as the server for the website. Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. It is used to provide small, robust tooling for HTTP servers, making it a great solution for single page applications, web sites, hybrids, or public HTTP APIs.

- Beaglebone

Beaglebone Black is a low-cost, community-supported development platform for developers and hobbyists

- PIR Motion Sensor

PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range. They are small, inexpensive, low-power, easy to use and don't wear out.

Tasks undertaken:

1. Collect data from motion sensor.
2. Send motion information to database which includes current time, motion start time and motion end time.
3. Implement three firebase functions:
 - First function sends email if motion length is $>T_1$ and $<T_2$ (where T_1 and T_2 are predefined thresholds.)
 - Second function sends email about summary of last five motions if available from the database.
 - Third function resets the database if length of motion is $>T_2$ where T_2 is a predefined threshold.

What we found out:

4. The motion sensor is not very accurate. Sometimes it works properly but sometimes it gives wrong readings.
5. All other risks that associated with this spike can be mitigated except the ones mentioned above.

Open issues/risks:

We could achieve all the spike goals except that the motion sensor is faulty.

Recommendations:

There could be one more spike to finding out different alternatives for finding out an accurate motion sensor.