KV6002 - Team Project and Professionalism

Evaluation Report

Liam Brand

0.1 The System Produced

This section aims to explore the implemented features with regards to how well they were implemented according to various criteria.

0.1.1 Fitness for Purpose

Subsystems were successfully implemented to meet the overall project goal outlined earlier. Two sensors were developed that posted data to a database, and a front end containing appropriate functionality for data visualization and analysis was created as well. The visualization and analysis subsystems were also made capable of communication with both devices, allowing the system to communicate with multiple environmental sensors. The physical demo was proof of the product's functionality.

As well, these elements have been implemented to be near real-time, as stored and processed data needs to be up to date for the system to be accurate. Analysing an hour old reading for example might not provide useful information as it's uncertain whether or not this reading represent the environment's current state. The sensors post their readings to the database every 15 seconds, allowing for the database to receive sensory data at a relatively quick pace. As well, XML http requests have been configured to fire every few seconds for relevant data visualization and data analysis methods, ensuring what is displayed or analysed is the most up to date information available and provides the most value to the user.

Adherence to industry standards was also achieved where possible. For example, the model features an ability to inform users of problematic temperature differences between sensors. When discussing temperature differences between rooms, IBACOS - a strategic partner of the US Department of Energy - notes that "Room-to-room temperature differences or floor-to-floor temperature differences should be no greater than 4 degrees Fahrenheit in the heating season"[4]. The model therefore used this as a guide for its temperature difference threshold, but unknowns such as the temperature location, potential for sensory noise and use of only one sensor rather than one for each room, so the threshold was increased to four degrees celsius to accommodate for these aspects.

0.1.2 Robustness

AWS Lambda is used, which features functionality to allow scaling[2] allowing the database to cope with additional device implementations. The devices themselves have features to cope with failures. It features two temperatures sensors with the second one being used if the first one fails, features in code to prevent erroneous measurements. As well, the devices will automatically attempt to reestablish internet connection upon being disconnected, and they also feature a 32GB Micro-SD card as temporary storage for readings data should connection be lost. Web elements such as the model's code contains try/catch statements to handle exceptions such as empty readings from the database as well, and will print appropriate messages to the console to assist in troubleshooting should these issues occur.

0.1.3 Look and Feel

Petrie and Fraser[7] describe different issues that significantly impede users' ability to properly make use of a website. The primary offenders are complex pages, unclear navigation and poor colour contrast. The website's pages are simple, with large block elements for listed devices and clear headings denoting sections such as the model's device comparison results. Navigation is achieved via the use of a navbar, something seen often in websites and so it shouldn't be unfamiliar to users. The website's colour scheme is mainly dark blue and white, offering a good contrast between the background and elements like text. These aspects allow the system's interactive element to be familiar, easy to use and visually pleasing.

0.1.4 Consistency

Subsystems were developed with other subsystems in mind. For example, the website was properly formatted to make space for the data visualization and intelligent model. These elements have clear areas of the web page reserved for them that allows them to blend into the rest of the website, and they were included in the media queries to ensure they scaled to different screen sizes. The database endpoints allow the developed sensors to post data to the database, and endpoints were also created to provide the visualization and model subsystems with the data that they needed to perform their functions. These things allowed subsystems to work together without anything feeling like it wasn't accommodated for.

0.1.5 Technical Evaluation

Employed technology was appropriate for each subsystem. The devices used sensors that gathered relevant data, the database allowed the creation of needed endpoints and included inbuilt scaling technology, and the website subsystem's usage of HTML and CSS allowed for a pleasant looking website. The use of JavaScript for the visualization and model subsystems allowed for the creation of automated HTTP requests that ran every few seconds. This allowed for the regular retrieval of relevant data from the database which helped these systems be accurate as they always use the latest data available in the database, and as such the latest data from the sensor.

Code featured appropriate comments, indentations and method/variable names to maximize code readability. Proper functional decomposition was done where appropriate too. For example in the model, part of determining an ideal temperature was to look at temperature readings from the same time for previous days. For this, timestamps needed to be created so readings made around these times could be searched for and retrieved. Creation of these timestamps was initially done in the *checkTemperature* method, but it was later turned into a *getPreviousTimestamps* helper function to maximize readability and group together relevant code.

0.1.6 Non-Functional Requirements

Primary non-functional requirements concerned the website's usability and the rate at which data was retrieved from the sensors and processed. These aspects have already been covered in previous sections. There is also a requirement for the development of an economic model to help cost the intelligent model, which can be viewed in Appendix B.

0.2 Project Management, Process and Personal Achievement

0.2.1 Terms of Reference

The description of subsystem requirements aided development by providing an understanding of the scope, complexity and methods/technologies that were used for their implementation. For example, the model's specification helped bring up some early questions regarding its possible behavior (e.g. what to do if it's too hot) and this allowed the developer to know what needed to be understood before development could begin. Discussion of the project's testing could have used more depth however, and things like unit testing (and perhaps the respective testing frameworks for the different technologies being used) could have been mentioned.

0.2.2 Requirements and Design Documentation

Wireframes for the website were created, these helped in understanding how the website would look which helped development, and also allowed the data visualization and model subsystems to understand how their respective information should be output so that it can fit into the created website (e.g. how do we tell the user it's too hot in a way that displays well on the designed page?).

0.2.3 Time Management

Most time objectives were met, shown by the meeting minutes' satisfaction of a live end to end product by March. Development primarily finished mid April however rather than the start of April, so a more thorough understanding of task complexity and needed development time is required.

0.2.4 Configuration Management and Integration Strategies

Configuration and integration was primarily managed through the use of version control. A global repository was created with each group member added as a collaborator, and changes were committed and pushed to their own branches where appropriate. These branches were reviewed to determine their compatibility with existing code and merged into the master branch after. This ensured that all written code fit together without breaking other elements, and also provided a global repository accessible by each group member for the project's source code. Appendix A shows the repository's commit log. Integration of subsystems was also discussed during meetings, for example the meeting on the 25th of March features an end to end demo utilizing all subsystems together as an objective.

0.2.5 Testing Strategies

Use of the project during code merges on git ensured that code updates didn't break the project, and a live end to end demo was conducted prior to the demonstration to ensure all features worked. A more thorough style of integration testing with a sophisticated testing plan would have been better however, and unit tests for specific code elements would have helped ensure system stability.

0.2.6 Group Leadership

No leader was elected among the group, this had issues in determining solid internal deadlines on different pieces of functionality. Likely this is one of the reasons that the project's time plan couldn't be properly adhered to.

0.2.7 Quality Planning

Group members inspected each others developed code to look for faults overlooked during initial development. These code reviews helped prevent tunnel vision where issues are repeatedly overlooked. To increase efficiency, notes could be made from these code reviews and published in a group folder for others to see, so they can check their own code during development for similar mistakes.

0.2.8 Potential Additional Functionality

As of right now the visualization and analysis subsystems are hard coded to only retrieve data from the two created devices, moving forward this should be made more dynamic to accommodate for additional created devices to allow these subsystems to function properly as more devices are added. In addition, the data visualization could be scaled up to display data from longer periods of time (e.g. months or years). Also, the data analysis is limited in what it does with the information it learns. For example if gas or an extremely high temperature is detected in a reading this manifests through website alerts, but a fully realised product could notify the appropriate services automatically to help the user. Important notifications could also be harder to miss, sending them to the users phone as a regular mobile notification. These aspects may have been achievable with better time management.

0.2.9 Problems Encountered, Their Solutions, and Lessons Learned

Better time management could have been achieved from sticking to the project plan, in future more attention will be paid to these time plans to prevent falling behind too much. Also, a better idea of whether or not to have a leader will allow more discipline in sticking to self imposed deadlined, so in future it would be better to explicitly discuss whether or not the group will have a leader and if so, who it is.

0.3 Professional Issues

The Code of Conduct was adhered to with good collaboration between teammates, as each member showed up to the group meetings and gave fair warning if other commitments would cause them to be late. Members were polite whilst still offering fair criticism. Moving forward these standards would need to be strictly adhered to, as insufficient communication between team members could cause project issues that directly affect clients causing a loss of faith in the company and as such the developed product.

0.4 Legal Issues

Data breaches against stored data were considered during the TOR. The usage of AWS helped mitigate this, with their databases featuring implemented security measured[1] such as encryption and private keys required for endpoint access helping ensure the stored data is private. The repository containing the project's source code was also private with read and write privileges only given to group members, to prevent source code leaks. Terms of use such as copyright conditions were also adhered to, for example the AWS service terms[3] were reviewed against proposed project functionality to ensure what we were using them for was acceptable by Amazon. These terms included things like ensuring the only data stored on the database was lawfully obtained by us, and these terms would need to be kept in mind once other users come into the picture should the product be deployed industrially.

As well, the Data Protection Act[6] will need to be considered should the project be deployed. In particular, the Act states that a person has the right to find out what data relating to a person is stored by an organisation. Facilities to provide customers with this data if they request it should be set up. It also states data shouldn't be kept for longer than is needed, so it should be determined how long sensory data is required for the system's ideal function and then functionality should be in place to ensure data that is no longer needed is properly disposed of.

0.5 Social Issues

The major social concern with this product noted in the TOr is the perception of it from the public. Devices inside of a person's house sending information about its status can make many uncomfortable. During their study into the reasoning of users behind using or not using smart sensors and their opinions on them, Lau et al[5] have a participant who says "with smart speakers...I can avoid having it, so I'd really not, I don't want another thing that could possibly violate my privacy". This was difficult to mitigate during development as industrial deployment of the product never happened, so it was impossible to determine how users felt about it being in their home.

Should the product be deployed, the company needs to have complete transparency in how the sensors behave, what they store about people's homes, how this data is stored and how it is processed to provide system functionality such as the short term decision making. This could be in the form of an FAQ on the website, as well as a complete willingness to answer more specific question should users contact the company. Being open will help dissuade any negative perceptions of the sensors as customers will fully understand what the product is doing and why.

0.6 Ethical Issues

The main ethical concern identified for the project was misuse of sensor date. This was mitigated during development as to retrieve sensory data special keys are required, and these keys were only given to developers who needed certain endpoints for database access. This meant everyone only accessed the endpoints (and as such, the data) that they needed. With regards to the database itself, access was mainly restricted to the subsystem's developer.

Should the product be deployed this policy will need to continue to be adhered to so that only those who actually need data will have access to it. If a larger company was established, data privilege levels should be assigned to employees that dictate what they have access to. These levels will be based on what data the employee will need access to in order to perform their job.

Bibliography

[1] Amazon. Aws database documentation, . Available at: `https://aws.amazon.com/products/databases/` (Accessed 02/05/19).

[2] Amazon. Aws lambda documentation, . URL: `https://docs.aws.amazon.com/lambda/latest/dg/scaling.html`.

[3] Amazon. Aws service terms, . Available at: `https://aws.amazon.com/service-terms/` (Accessed 02/05/19).

[4] Arlan Burdick. Advanced strategy guideline. air distribution basics and duct design. Technical report, IBACOS, 2011.

[5] Josephine Lau, Benjamin Zimmerman, and Florian Schaub. Alexa, are you listening?: Privacy perceptions, concerns and privacy-seeking behaviors with smart speakers. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):102, 2018.

[6] Government of the United Kingdom. Data protection act 2018, 2018. Viewable at: `https://www.gov.uk/data-protection` (Accessed 02/05/19).

[7] Helen Petrie, Fraser Hamilton, and Neil King. Tension, what tension?: Website accessibility and visual design. In *Proceedings of the 2004 international cross-disciplinary workshop on Web accessibility (W4A)*, pages 13–18. ACM, 2004.

Appendix A

Git Log

```
commit d722c703c63d1635d68bd3c9464539894c7d72c3
Author: liambrand <liam.brand@northumbria.ac.uk>
Date:   Thu Apr 25 13:24:08 2019 +0100

    Tweaked gas check

commit c2100a8d493d98e65188924935fd8d04fe981aad
Author: Jamie <jg.clarke@hotmai.co.uk>
Date:   Thu Apr 25 13:02:37 2019 +0100

    hardware work final

commit 4e88fbad654ee8a4901b47a9de6f6526d27fd0b4
Author: Jamie <jg.clarke@hotmai.co.uk>
Date:   Thu Apr 25 12:56:57 2019 +0100

    final commit!

commit e9c9e37bedc9fcea9f30d7def13166bf45028927
Merge: 8275392 97545f9
Author: SnowTurtle96 <jg.clarke@hotmail.co.uk>
Date:   Thu Apr 25 12:47:07 2019 +0100

    Merge pull request #9 from SnowTurtle96/pass-encryption

    Pass encryption

commit 827539232911f3a873078a8efe60ddb89b73877f
Author: liambrand <liam.brand@northumbria.ac.uk>
Date:   Thu Apr 25 10:47:42 2019 +0100
```

```
    Fixed gas detection threshold

commit c0dbf5c532cff17205ae22687ed8e4f014fb450f
Author: liambrand <liam.brand@northumbria.ac.uk>
Date:   Thu Apr 25 10:07:52 2019 +0100

    Added dumb gas method with associated alert

commit 06f762ba3b0f77f3105da2ca33f8bae0e5f9e9ff
Author: liambrand <liam.brand@northumbria.ac.uk>
Date:   Thu Apr 25 09:48:39 2019 +0100

    Some cleanup

commit 597e531ddeab7d568fad4a1b6baf17017f9a20e8
Author: liambrand <liam.brand@northumbria.ac.uk>
Date:   Thu Apr 25 09:45:17 2019 +0100

    Added relevant sections for device comparison

commit 7d281cad7ca6c8c3a9cf8cbc950d0906b21500a7
Author: liambrand <liam.brand@northumbria.ac.uk>
Date:   Thu Apr 25 09:21:13 2019 +0100

    Buffer for device comparison

commit 97545f94580f23e51a0424ae89a761f5729f88d6
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
Date:   Wed Apr 24 21:07:01 2019 +0100

    Comments

commit 96fe6d0983d07975799b013bae0d86ebad5adbcb
Author: liambrand <liam.brand@northumbria.ac.uk>
Date:   Wed Apr 24 12:09:46 2019 +0100

    Removed another method stub

commit 80d7b8e6cfcd8d8ea4dc9c719a276106bfe5dcaa
Merge: 5aa0d7d 69ae270
Author: liambrand <liam.brand@northumbria.ac.uk>
Date:   Wed Apr 24 12:05:25 2019 +0100

    Merge branch 'master' of https://www.github.com/SnowTurtle96/Group-Project

commit 5aa0d7dade55b69f0a5c1462e83784f1b13be2f8
```

```
Author: liambrand <liam.brand@northumbria.ac.uk>
Date:   Wed Apr 24 12:05:02 2019 +0100

    Removed old method stub

commit 0952571b3f1812b16745b6e7e98a92b11e199517
Author: liambrand <liam.brand@northumbria.ac.uk>
Date:   Wed Apr 24 11:32:18 2019 +0100

    Added dateFns library

commit 826ae89d7301a16dcf03106f2ba22631b02bba72
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
Date:   Wed Apr 24 10:59:33 2019 +0100

    Changes - comments required

commit cd589f4765d9673c5c3a78265602220cec79117f
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
Date:   Wed Apr 24 01:24:23 2019 +0100

    Summary page updates

commit 69ae2702f6a244a422bcce9c5c90a7ebb32e110f
Merge: db2f8e5 9353b52
Author: SnowTurtle96 <jg.clarke@hotmail.co.uk>
Date:   Wed Apr 24 00:57:55 2019 +0100

    Merge pull request #8 from SnowTurtle96/pass-encryption

    Pass encryption

commit 9353b5263e26dd3d2eeced6caa6ae53fa161c8a5
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
Date:   Tue Apr 23 23:53:34 2019 +0100

    Updates

commit 31b620a4a4da00b37ee6a9ad74d87d3111b0aab1
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
Date:   Tue Apr 23 23:43:25 2019 +0100

    CSS updates

commit 1516d424824b8ec2ab9a2f352252071e48d4e450
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
```

```
Date:   Tue Apr 23 23:20:24 2019 +0100

    Delete device added

commit 164e1eb0dc6da3c2c1efc86a71bdffc2211f8bf7
Merge: 65aa0e0 db2f8e5
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
Date:   Tue Apr 23 22:50:31 2019 +0100

    Merge branch 'master' into pass-encryption

commit 65aa0e051354d7e6000ac2fbbce55921342e9ff1
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
Date:   Tue Apr 23 22:37:44 2019 +0100

    Update

commit db2f8e56792db7ace88b856ff3088cdbb24b2c17
Author: liambrand <liam.brand@northumbria.ac.uk>
Date:   Tue Apr 23 21:35:56 2019 +0100

    Added device comparison

commit 53320775540b6efd39b5647418164938e9210498
Author: liambrand <liam.brand@northumbria.ac.uk>
Date:   Tue Apr 23 20:04:12 2019 +0100

    Model implemented into respective pages

commit 5525e536f964d8232b2f4fef2feb58cffed66dce
Merge: ea03845 515a577
Author: SnowTurtle96 <jg.clarke@hotmail.co.uk>
Date:   Tue Apr 23 18:42:10 2019 +0100

    Merge pull request #7 from SnowTurtle96/Charts

    adding all changes to charts, weekly left

commit 515a5772c2a6be1ddd4a640f41dbf71db4229dbc
Author: ANewby <adam_newby1@hotmail.co.uk>
Date:   Tue Apr 23 18:38:30 2019 +0100

    adding all changes to charts, weekly left

commit ea0384534b330c57f67b17086faeadde1062bd86
Author: Student <student@C17775497.offcampusnetwork.co.uk>
```

```
Date:   Tue Apr 23 15:34:12 2019 +0100

    files deleted

commit 11da59995d028142dbebcd9598397618a48464f6
Merge: 2ea683c 6e37c57
Author: Student <student@C17775497.offcampusnetwork.co.uk>
Date:   Tue Apr 23 15:30:13 2019 +0100

    Merge branch 'master' of https://github.com/SnowTurtle96/Group-Project

commit 2ea683cd7b820f3268a6ae0b111beb9394d3b36e
Author: Student <student@C17775497.offcampusnetwork.co.uk>
Date:   Tue Apr 23 15:29:38 2019 +0100

    code commented and documentation included

commit 6e37c5763d8a1fdd6cb8966844842166f79aea24
Merge: fa6480b 2dc4b8c
Author: SnowTurtle96 <jg.clarke@hotmail.co.uk>
Date:   Tue Apr 23 00:56:06 2019 +0100

    Merge pull request #6 from SnowTurtle96/pass-encryption

    Pass encryption

commit 2dc4b8c15e75f3fcba16dfc153e847ec4e27db1e
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
Date:   Mon Apr 22 22:45:30 2019 +0100

    Tabindex update

commit fa6480be966069a2fe78eb5a9f7145c7677ae4a0
Author: liambrand <liam.brand@northumbria.ac.uk>
Date:   Mon Apr 22 22:39:14 2019 +0100

    Changed hardcoded user id to 97

commit f2ba6703d4b0c7a91bfd32d39993f6840dacffc2
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
Date:   Mon Apr 22 22:35:25 2019 +0100

    Encryption complete

commit cfc02ad6d4dcf8fa546ede7d9c78ff898e5a5334
Merge: 775a714 cdbe33a
```

```
Author: SnowTurtle96 <jg.clarke@hotmail.co.uk>
Date:   Mon Apr 22 22:21:22 2019 +0100

    Merge pull request #5 from SnowTurtle96/media-queries

    Media queries

commit cdbe33acfe63fc87fadb21b6d21b559fb19e4b4d
Merge: 4c70475 775a714
Author: SnowTurtle96 <jg.clarke@hotmail.co.uk>
Date:   Mon Apr 22 22:21:07 2019 +0100

    Merge branch 'master' into media-queries

commit 4c70475574e6d07e2ab5b9b3ad94a898fcd57bc5
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
Date:   Mon Apr 22 11:25:28 2019 +0100

    Update device-entry.php

commit 775a714c4aba00db4228c054c1b03a30713f3d78
Author: liambrand <liam.brand@northumbria.ac.uk>
Date:   Mon Apr 22 09:43:57 2019 +0100

    Removed example.js file from model

commit 5ef66432bbf19da900ba51899d00236c83988b4a
Author: liambrand <liam.brand@northumbria.ac.uk>
Date:   Sun Apr 21 15:05:43 2019 +0100

    Incorporated AlertifyJS, prototype device comparison implemented

commit 38831d87b65898dca2296fe217ceeec26c152592
Author: liambrand <liam.brand@northumbria.ac.uk>
Date:   Sun Apr 21 14:07:42 2019 +0100

    Newer model, changed imports to load jQuery first

commit 888ac6580f45dbe21b30ddb9d0e25bcb424cfc87
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
Date:   Fri Apr 19 12:16:36 2019 +0100

    Index media query

commit 21a845d0b11aba6a179d5bed441156afd7f9a7c7
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
```

```
Date:   Thu Apr 18 00:05:36 2019 +0100

    Media query prep, model data added (bugs)

commit 66be4147291cb68041db521b4e9998e3ffe1928b
Merge: 5043823 03fe8ff
Author: SnowTurtle96 <jg.clarke@hotmail.co.uk>
Date:   Mon Apr 15 23:51:12 2019 +0100

    Merge pull request #4 from SnowTurtle96/branch-apicall

    Branch apicall

commit 03fe8ffe231b498d4e8acfe52dacc7db5a37f5d9
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
Date:   Mon Apr 15 23:47:57 2019 +0100

    Device summary pages

commit 75283e7cab5b6d9389d4ac8829427b266caac99e
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
Date:   Wed Apr 10 22:10:00 2019 +0100

    Test

commit 97efb58274b2c78bc208bea50acffad4c1657185
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
Date:   Wed Apr 10 21:43:55 2019 +0100

    Register fix

commit 3666397c002bb328b8cd39a7b7e3ca105b0de92e
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
Date:   Wed Apr 10 20:49:58 2019 +0100

    Small fixes

commit 525a61c599e1b504a5960910087d730b04e54c35
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
Date:   Wed Apr 10 16:53:13 2019 +0100

    Added no-devices scenario

commit 6f4eab68c8b35ba284d1709b0b06458315703c4a
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
Date:   Wed Apr 10 16:43:22 2019 +0100
```

```
    apicall fix

commit 5043823dd50f80c7d9fae0940121880f289ab523
Author: liambrand <liam.brand@northumbria.ac.uk>
Date:   Wed Apr 10 13:49:59 2019 +0100

    Basic model working

commit c56a98297993f9fac9de92603913f2297fae98ba
Merge: 764706f 998f3a8
Author: SnowTurtle96 <jg.clarke@hotmail.co.uk>
Date:   Wed Apr 10 11:47:13 2019 +0100

    Merge pull request #3 from SnowTurtle96/branch-apicall

    Layout change, api-call issue

commit 998f3a861d61bdf0673fb79dfe88b60de9aba922
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
Date:   Wed Apr 10 11:44:01 2019 +0100

    Layout change, api-call issue

commit 764706f3c5270b6e65170fef13e52c1e7ef763cc
Author: liambrand <liam.brand@northumbria.ac.uk>
Date:   Wed Apr 10 11:14:55 2019 +0100

    Moved script includes to header.php

commit 92d46488d5d63eb4eabcee60a25bfc7a70f5b99c
Author: SARABIRDS <under863@gmail.com>
Date:   Wed Apr 10 02:11:27 2019 +0100

    initial commit

commit 0c81167490a0035269786dc66ed1e99e419476a1
Merge: 34bc207 419ab61
Author: SnowTurtle96 <jg.clarke@hotmail.co.uk>
Date:   Wed Apr 10 01:00:43 2019 +0100

    Merge pull request #2 from SnowTurtle96/branch-apicall

    Branch apicall

commit 34bc2076cd62293fc3cf750ff1549d793d300cf5
```

```
Merge: 3945797 6d37c6d
Author: SnowTurtle96 <jg.clarke@hotmail.co.uk>
Date:   Wed Apr 10 01:00:23 2019 +0100

    Merge pull request #1 from SnowTurtle96/model

    Model

commit 3945797b0f5b89bcec6f22fdc6583e70e81a8e4f
Author: Jamie <jg.clarke@hotmai.co.uk>
Date:   Wed Apr 10 00:54:36 2019 +0100

    lambdas

commit 419ab619b006aa7cdc4f28f93a820933239e8722
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
Date:   Wed Apr 10 00:45:11 2019 +0100

    Start of user device calling - broken

commit 9e45e746014855001083f4980e4e755c9fc9a594
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
Date:   Mon Apr 8 23:11:35 2019 +0100

    Device entry fix

commit 60be8dbbde2454c9fadb8a122b8a7d6d0df18abf
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
Date:   Mon Apr 8 22:52:44 2019 +0100

    Device creation fix

commit 22bf7c646c9953e108ee8325a36155b95dcd0fbe
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
Date:   Mon Apr 8 22:21:17 2019 +0100

    Login functionality complete

commit d2b285a4ff59f965f7c57ff3b9190d2a461e52b1
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
Date:   Mon Apr 8 01:28:28 2019 +0100

    Login validation check

    To fix
```

```
commit c3017e18eb7d6c34b03f8bcd5abd9120969e1676
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
Date:   Sun Apr 7 23:52:26 2019 +0100

    API for new device, plus form added

commit 5ac1c169450f8bba7e68e3e18cab66737b355d79
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
Date:   Sun Apr 7 18:44:06 2019 +0100

    API fix

commit a4e14c146c1efb4b8464c6196feec3b3f6ad4d2d
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
Date:   Sun Apr 7 15:01:08 2019 +0100

    apiCall function updates, addition of login GET

    To fix

commit 0a73ec96aca445665d81f01cd49b336a97813bdf
Author: chris-mordue-nu <chris.mordue@northumbria.ac.uk>
Date:   Fri Apr 5 03:24:21 2019 +0100

    API-Call helper file

    Almost complete helper file for API call, data passing other than login/return.

    issue:
    fname=test&lname=test&email=tst%40test.c&pass=ttt&login=&return=

commit d71ea224d9d4af0cfe0cecd879a990bd9e85f832
Author: ANewby <adam_newby1@hotmail.co.uk>
Date:   Thu Apr 4 23:46:24 2019 +0100

    Charts added with two dropdowns

commit 6d37c6d5f8d3655aad54eee13da8ecd118e41cfb
Author: liambrand <liam.brand@northumbria.ac.uk>
Date:   Thu Apr 4 20:30:35 2019 +0100

    Merged model.js into device-model for now

commit 25798897952dd58808026fabb2bc4fefb46db44e
Author: liambrand <liam.brand@northumbria.ac.uk>
Date:   Thu Apr 4 20:16:08 2019 +0100
```

```
    Added model files

commit 093b2270a54113ea4514dd034c4c7f5c165e7ddf
Author: Jamie <jg.clarke@hotmai.co.uk>
Date:   Thu Apr 4 13:29:10 2019 +0100

    new folder structure, Chris's web work

commit 6d4b051a8121a5e77554fbf1ce1710b5cdcb58ca
Author: Jamie <jg.clarke@hotmai.co.uk>
Date:   Tue Apr 2 15:14:19 2019 +0100

    Initial Commit
```

Appendix B

Economic Model

Economic Breakdown for the Short-Term Model

As part of the model's development, the system's implementation needs to be financially evaluated to determine the cost of its deployment on an industrial level. The Terms of Reference outlines the project's main finances, but here will be a more in-depth exploration of the financial aspects behind the model specifically. There will also be a brief look at the potential finances behind developing a more sophisticated model.

Current Model

If the model was to continue being used in its current form, this would involve it continuing to retrieve device data from the system's database via the use of an endpoint. This endpoint takes the user's id then returns all associated device information, allowing it to be searched for relevant information that can be used for the short-term decision-making process.

The primary cost consideration here would be the continued and increasing usage of AWS Lambda, an Amazon service that automatically runs associated code for various events. In the case of our system, the event is when an http endpoint is triggered. Usage of Lambda remained free during the prototype stage thanks to the relatively low amount of requests and data use. Moving forward however, possible costs will need to be considered. Lambda's usage is based on two things, requests and compute time. Requests pertains to simple requests that call Lambda code, and compute time is the time it takes for the code to execute and terminate. The costs (shown below in dollars) can be viewed on AWS' Lambda page.

Requests

Lambda offers the first million requests each month for free. First how many users the system in its current implementation can support whilst remaining in this free tier will be calculated, then a rough cost per user in request fees will be determined.

The model sends a request to the endpoint that returns device data approximately every four seconds. To be completely prepared for financial ramifications, the most intensive scenario will be assumed where the model runs constantly and is at no point deactivated for maintenance or updates.

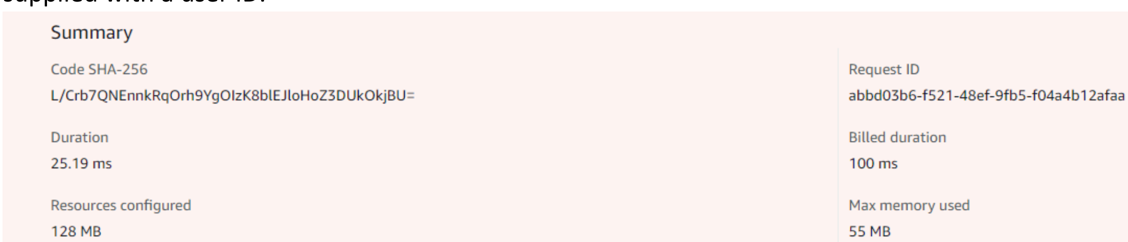
With 86400 seconds being in a day, scheduled retrieval of a single user's data will use 21600 (86400/4) requests each day. That means over the course of 30 days, a single user's model will use 648000 requests. Dividing a million by this figure yields a little over 1.5, meaning in its current state the system can only support one user full time without incurring fees. Beyond this point, requests would need to be paid for.

Lambda offers 1 million requests for $0.20 (AWS Lambda, 2018), which is equivalent to £0.15 at the time of writing. We can work out the cost of a single request by diving this cost by 1 million, then by multiplying this by the number of requests needed for a single user's model we can determine how much in requests a single user will cost. By doing this, we arrive at the figure of £0.0972 per user. So, for each user that uses the product that the current model needs to accommodate for, an additional £0.0972 per month will need to be spent on requests from AWS Lambda.

Compute Time

Lambda offers 400,000 gb-seconds per month free per month. First it will be determined how many users can be supporter by the system in its current implementation without incurring costs. Then a cost per user in compute time fees will be determined.

The screenshot below shows the Lambda details for the endpoint that returns user device data once supplied with a user ID.



The memory used by the endpoint doesn't always equal 55mb, but to ensure that the worst-case scenario has been prepared for it will be assumed that every request uses this much data. This amount of data can increase if more device data is returned but this increase is ultimately negligible, nearly all the computing time used is from processing the respective scripts and code that is used for the endpoint's function rather than the actual bandwidth used returning the data.

400,000 GB-seconds per month equates to around 400000000 megabytes. This is enough to process around 7272727 requests. It was earlier established that a single user will (in a worst-case scenario) require 648000 requests to retrieve up to date model information. This means that from a free perspective, the current system's usage of Lambda can support around 11 users (7272727/648000 is about 11) before costs are incurred from the usage of the endpoints. After this point, a single user's model requests will use around 35640000 megabytes of compute-time, equating to around $0.59 in monthly costs. At the time of writing, this translates into around £0.45.

In summary, the free tier will be exceeded by just two users. Even though Lambda's memory allowance offers support for many more, the sheer number of requests required to keep the model up to date burns through the free allowance quite quickly. Only the respective free allowances have been exceeded, each user will cost £0.0972 in request fees and £0.45 in compute-time fees, giving a total per-user cost of roughly £0.54.

Improved Model

Rather than continuing with the current model's implementation, a more sophisticated model could be created that takes advantage of true deep learning technologies. This will allow its behaviour to be more accurate, as it will be capable of picking up on small nuances in user behaviour and distinguishing them from real problems. For example if a user frequently has parties at certain times of the year, the current implementation might flag this activity as a high temperature whereas a true machine learned model might recognise it and understand that it isn't a problem.

AWS

AWS offers technologies capable of deep learning. Whilst its lack of prior use in the project makes it difficult to estimate the cost per user as was done with the current model's implementation, instance pricing can still be reviewed.

The three main rentable instances in order of power are P2, P3 and G3 (AWS Deep Learning Developer Guide, 2018). To begin with, it would be best to start with the cheapest option. This would prevent money being wasted should this foray into deep learning prove to not be useful, but should more power be required it will be easy to scale up compared to wasting money and realizing what was purchased was overkill.

P2 instances are categorised by their specifications, with more expensive instances containing additional GPUs, vCPUs and more RAM, among other things. The closest P2 instances that can be obtained are based in Ireland, below is a pricing breakdown taken from the EC2 instance documentation of how much these instances would cost.

p2.xlarge	4	12	61 GiB	EBS Only	$0.972 per Hour
p2.8xlarge	32	94	488 GiB	EBS Only	$7.776 per Hour
p2.16xlarge	64	188	768 GiB	EBS Only	$15.552 per Hour

How much this will cost is something that can only be properly determined once they have been used, as how many computing hours are needed to train the model is unknown. As well, the model may not be usable during its training, but frequent training will likely be needed in order to keep up with user habits as they change and nuances occur. Two different instances could be obtained as a work around for this. As one model trains, the other performs the necessary short term decision making. Once this first model is done training it takes over and the second model begins training. This could be repeated frequently to allow for the models to both stay up to date and current, but alternating deep learning instances would mean this could happen with any service interruption.

Sources

- Amazon. AWS Lambda Documentation. Available at: https://aws.amazon.com/lambda/ (Accessed (03/05/19).
- Amazon. Recommended Deep Learning Instances. Available at: https://docs.aws.amazon.com/dlami/latest/devguide/gpu.html (Accessed (03/05/19).
- Amazon. EC2 Pricing. Available at: https://aws.amazon.com/ec2/pricing/ (Accessed 03/05/19).
- Amazon. P2 Instance Details. Available at: https://aws.amazon.com/ec2/instance-types/p2/ (Accessed 03/05/19).

Appendix C

Meeting Minutes

KV6002 MEETING MINUTES

Location: CIS

Date: 2nd March 2019

Time: 11:15

Agenda Items

1. It was agreed that the design and development of sub-systems should begin from next week, and designs for each sub-system are to be agreed by all team members before development begins.
2. The sending of device readings was discussed as a group, and it was agreed that data would not be updated in intervals and would only update if the reading value increased or decreased by a specified amount (such as a change of value by 0.5).
3. Only dummy data will be used whilst testing each sub-system for the overall product to ensure that it functions as expected, until readings can be received accurately.
4. It is important that the web application developed is mobile friendly, as this is replacing a mobile application.
5. The questionnaire should include a question for determining the purpose of the website, whether it has a corporate front, with abilities to log in and manage devices, or, is solely for the management of a device.

Action Items	Owner(s)	Deadline	Status
Develop and agree product design diagrams, including; (1) **product wireframes**, (2) **ERD**	JC, AN, CM	04/03/19	New
Create file structure for web application	CM	04/03/19	New
List a minimum of three questions based on literature and product reviews for each sub-system to gather requirements.	All	06/03/19	New
Completion of a single device build	ZA	06/03/19	New

KV6002 MEETING MINUTES

Location: CIS

Date: 11th March 2019

Time: 10:00

Agenda Items

1. Continue with the development of the individual subsystems.
2. Gave brief overview of each other's statuses and how far along everyone is.

Action Items	Owner(s)	Deadline	Status
Finish an MVP product of each subsystem	All	18/03/19	New
Implement the answers from questionnaire with client	All	18/03/19	New

KV6002 MEETING MINUTES

Location: CIS

Date: 18th March 2019

Time: 10:00

Agenda Items

1. MVP product progress is to be determined.
2. Completed MVP subsystems should be explained so that the rest of the group can understand them, which will aid in joining subsystems together to form the whole product.
3. For subsystems not at the MVP stage, blockers will be discussed to overcome issues.

Action Items	Owner(s)	Deadline	Status
Get end to end demo	All	30/03/19	New
Implement live data for subsystems that require it	All	25/03/19	New

KV6002 MEETING MINUTES

Location: CIS

Date: 25th March 2019

Time: 10:00

Agenda Items

1. Product progress is to be determined.
2. Individual subsystem updates from each group member
3. Comparison between what each subsystem is and to the objectives that were outlined in the TOR

Action Items	Owner(s)	Deadline	Status
Get end to end demo with live data	All	03/04/19	Ongoing

KV6002 MEETING MINUTES

Location: CIS Ground Floor

Date: 5th April 2019

Time: 12:00

Agenda Items

1. Continue progressing with sub-systems, taking note that we will need time to test.
2. Update all group members on current progress, and what each hope to achieve.

Action Items	Owner(s)	Deadline	Status
Progress with individual sub-system	All	12/03/19	New
Keep group members updated on status and progress	All	12/03/19	New