Project Report

AniMap: Animal Tracker And Health Monitoring Application

Name: Gearoid Lacey

Student Number: C00183380

Due date: 5th April 2017

Table of Contents

Abstract

The aim of this document is to provide an overview of the project. This overview will include a description of the project, a section documenting the end products conformance to the original design document and functional specification, a section on the learning outcomes and a section documenting an overall review of the project.

1 Introduction

The purpose of this document is to provide an outline to my project AniMap. AniMap is a proof of concept project that aims to try and determine the source of Tuberculosis (hereby referred to as TB) in a herd of cattle by recording an animal's location in a field using a GPS device called NanoTracker.

The overview of this project will begin in the project description component of this document. This section will provide a description of each use case within the mobile application, which will be accompanied by various annotated screenshots.

Following this a section documenting the mobile applications conformance to the original functional and design specification will be supplied. This section will detail the differences, if any, in the use cases described in the functional specification. This section will also detail any differences, if any, in the database design and how the application functions overall.

In the next section all learning outcomes gained over the course of the project will be documented. The learning outcomes will be divided into two sections, technical and personal.

A section documenting problems I encountered over the course of the project will also be created. As well as documenting the errors encountered during the project this section will also include the resolution for each error. This section of the document will also include any features that were not implemented and the reason they were not implemented.

There will also be an appendix that will include emails showing my correspondence with Round Solutions, the providers of the NanoTracker.

2 Project Description

2.1 What is AniMap

AniMap is a proof of concept project that utilizes a multi-platform mobile application and a remote gps-tracking device to determine the source of TB in a herd of cattle. The process starts when the tracking device that is placed around the animals neck, sends its gps coordinates to a MySQL database that is hosted on pythonanywhere.

The application user, in this case a farmer, can then use the application to view each animal's latest location to within 2-meter accuracy. They will also be able to view the path the animal travelled around the field. Each coordinate pair is displayed using an animal icon on a map which, if clicked, will display the time and date the location was taken at.

A core feature of AniMap is its ability to provide the farmer with valuable information regarding where each animal is spending the most time in the field. From this the farmer will may be able to determine the source of TB in a herd of cattle if it has already been indicated to the farmer that there are infected animals in the herd by a vet.

It must be noted that, although the main aim of this project is to try and prove that the source of TB in a herd of cattle can be located using their coordinates, this device can be used on any animal, and therefore the users of this application are not limited to farmers.

2.2 Core components of AniMap

AniMap consists of three core components. These components are the remote tracking device (hereby referred to as the NanoTracker), the mobile application

and the flask API. Each of these components will be documented under individual headings.

2.3 NanoTracker

2.3.1 Brief Description

The NanoTracker is a gps-tracking device made by a German company called Round Solutions who specialize in creating and distributing device relating to the IoT and M2M markets. The device comes in several variants, a 2g module, a 3g module specifically for European countries and another 3g module for the American markets. The 2g module was selected for this project as it was marginally cheaper than its 3g counterpart by €15. The 2g module cost €184 at the time of purchase although it currently costs €169. The main factor in choosing the NanoTracker was based on the device be programmed in python 2.7.2 and also having an accompanying software development kit.

In this project, the device functions by gathering its current gps coordinates and time which it then sends via http post to the Flask API at user defined intervals. The NanoTracker is powered by a 380mAh battery. There is no exact battery life as such as it is dependent on what the user sets the time between http posts to be. The NanoTracker uses a nano-sim to send data to the Flask API that is hosted on pythonanywhere. It is also important to note that to interface with the device on your computer you must install Virtual Com Port drivers which are linked in the NanoTracker manual.

To upload and remove python scripts to the device a software package called RS-Term-Plus is used which is available on the Round Solutions website. The RS-Term-Plus package is used for a wide range of products so not all functions available in the package are relevant to the NanoTracker. The functions from the rsterm package that were relevant will be documented along with annotated screenshots in the following section.

2.3.1 Rsterm

The following screen displayed in figure 1 is the initial screen of the RS-Term-Plus package. There are three elements of this screen that must be noted. They are the Com Port value that is specific to the machine, the bit rate for data transfer. As displayed in figure 1 the required bit rate as per the Nano Tracker manual is 115200. Another important section of the screen is the send command section which contains a button labeled AT. If the user has installed the VCP drivers and selected the correct Com Port, pressing the AT button prints “OK” in the message box which is displayed in figure 2. This is used to determine if the NanoTracker is responsive. Underneath the attention command button is an input field and a send button. This allows the use manual input command to send to the device. It is currently populated with the sequence “aaa”. This command is used to terminate execution of the currently executing script.

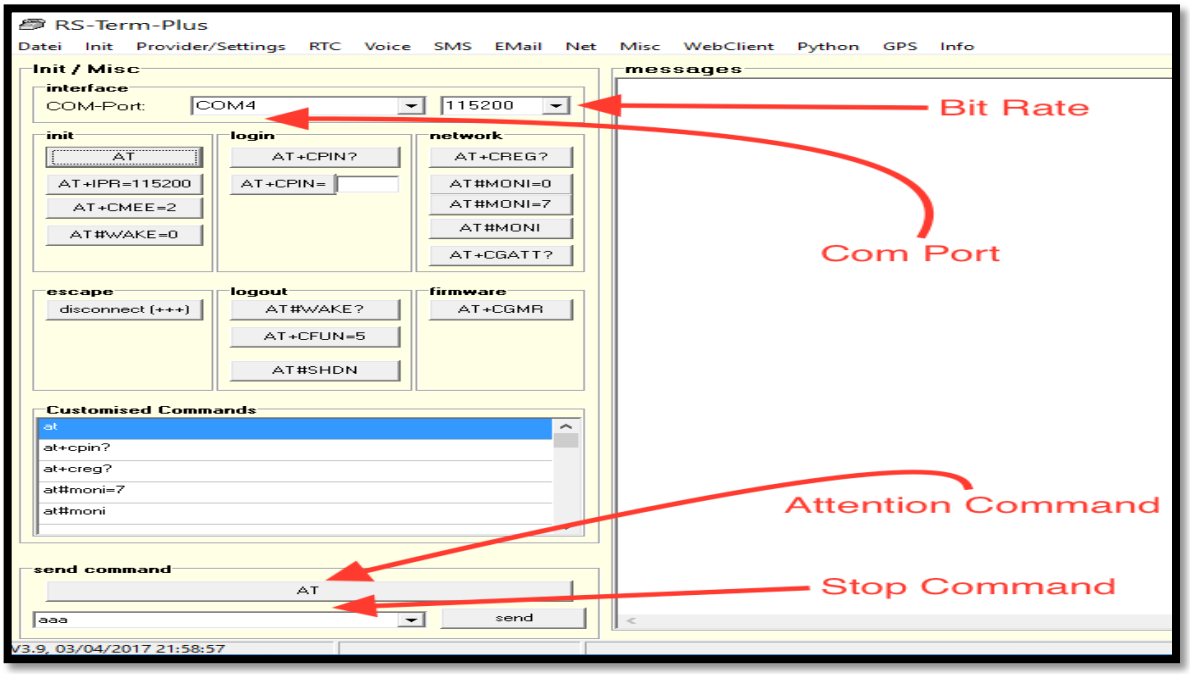


Figure 1. RS-Term-Plus Opening screen

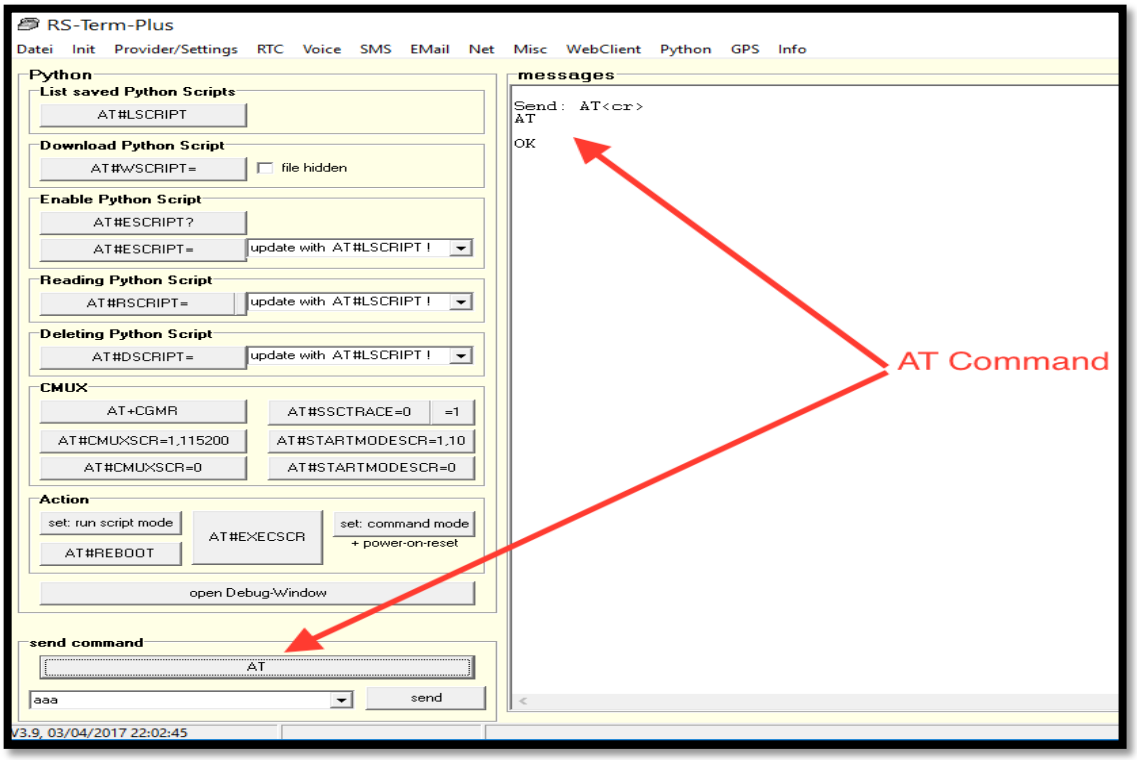


Figure 2. AT Command In The Python screen

There are five buttons on the python screen that were the most commonly used throughout the project. They are the list scripts button which list the scripts currently on the NanoTracker, the download script button which downloads a script to the NanoTracker from the computer, the enable script button which sets a script as the executing script, the read python script button and the delete python script. The buttons are all indicated in Figure 3.

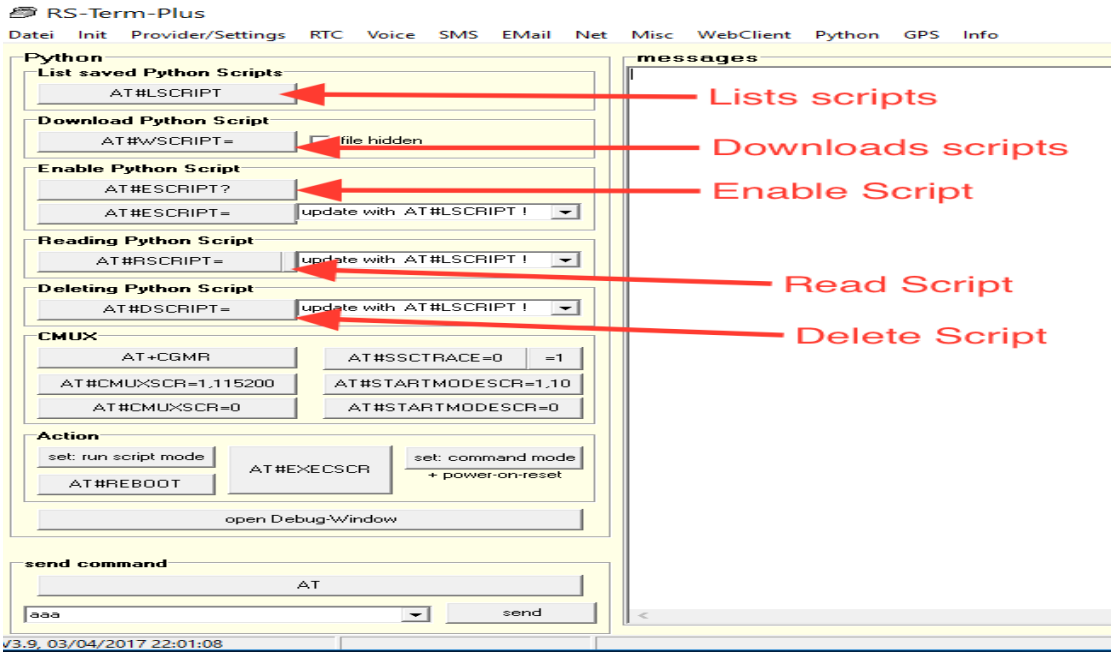


Figure 3. Button Functions

There are three drop down fields in Figure 3 associated with the enable, read and delete script buttons. There are currently populated with the text “update with AT#LSCRIPT”. This requires the user to press the list scripts button which will then populate the drop down fields with all the scripts currently on the device.

2.4 Android Application

2.4.1 Brief Description

The AniMap mobile application was built using Adobe PhoneGap. Adobe PhoneGap is hybrid mobile application development framework that allows developers to create applications using HTML, CSS and JavaScript that can then be built for multiple mobile platforms as opposed to creating individual applications for each mobile platform. The mobile application then sends data using jQuery to a python Flask API that is hosted on pythonanywhere.

2.4.2 Home screen

On opening the application the user is presented with a screen displaying three options. They are Login, Register or Change Password which can be seen in Figure 4. This page requires no backend functionality, as the user will only be moving to another page.

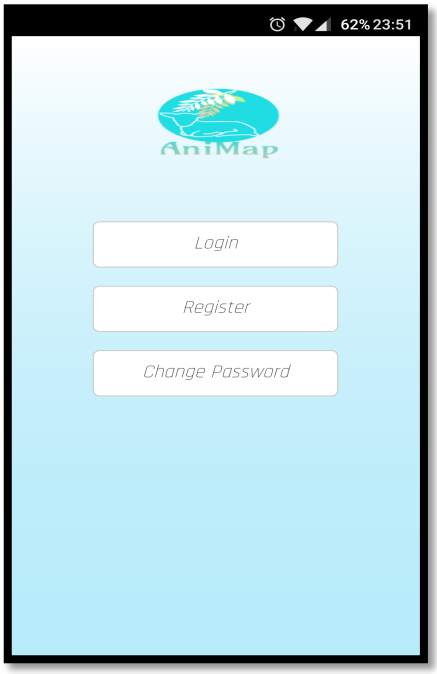


Figure 4. Home screen

2.4.3 Register screen

If the user clicks the Register button they are redirected to the Register screen. They are then required to enter a username and password. They are also required to confirm their password to eradicate any potential errors in the password creation process. When the user clicks the register button the form data is sent using jQuery to the Flask API. For this project all validation is performed in the API. On the Register screen in particular there are checks in place to determine if the user has filled in all the fields, if the username is not already taken and that both password fields match. If any of this validation fails the user is presented with an appropriate error message describing what was wrong. On successful registration the user is automatically logged in and redirected to the Menu screen. Note all passwords are hashed before they are stored in the database.

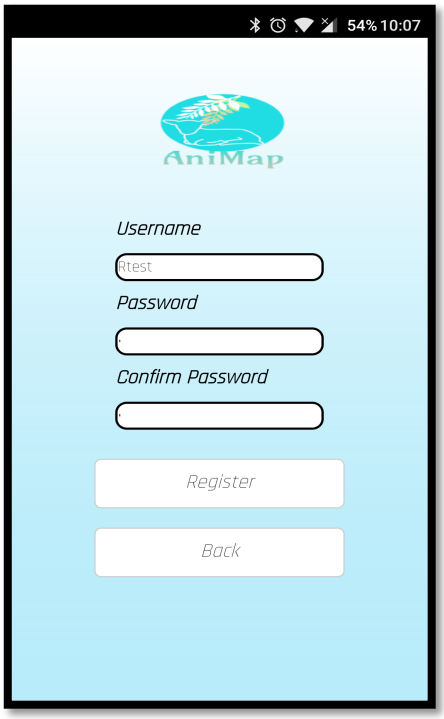


Figure 5. Register screen

2.4.4 Change Password

If the user clicks the Change Password button they are redirected to the Change Password screen. They are then required to enter their username and password. Following this the user may enter a new password which they must match in the confirm password field. The validation on this screen consists of checking the user has not left any fields blank, their username and current password are correct and that the new password and confirmation of the new password match. In this case a successful scenario would result in the users password being updated in the register table. The user would also be automatically logged in and redirected to the Menu screen. Note all passwords are hashed before they are stored in the database.

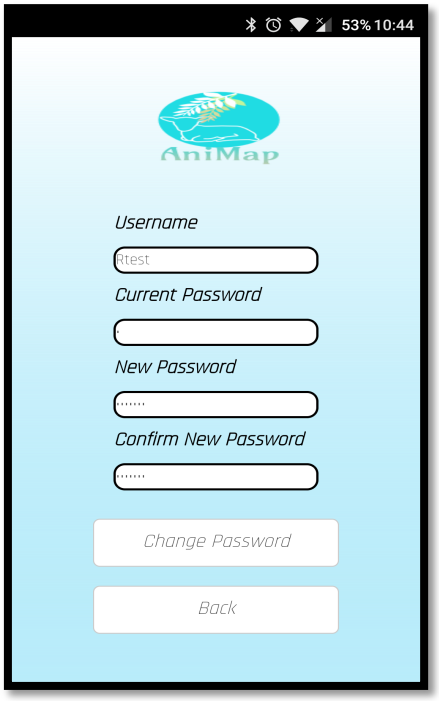


Figure 6. Change Password screen

2.4.5 Login screen

Assuming the user already has an account and has selected the login option the user then advances to the Login screen. The login in form is quite small which makes for easy access for an already registered user. The validation undertaken on this screen consists of checking that no fields are left blank, checking that the user exists in the database and that the entered password is correct. On successful login the user is redirected to the Menu screen.

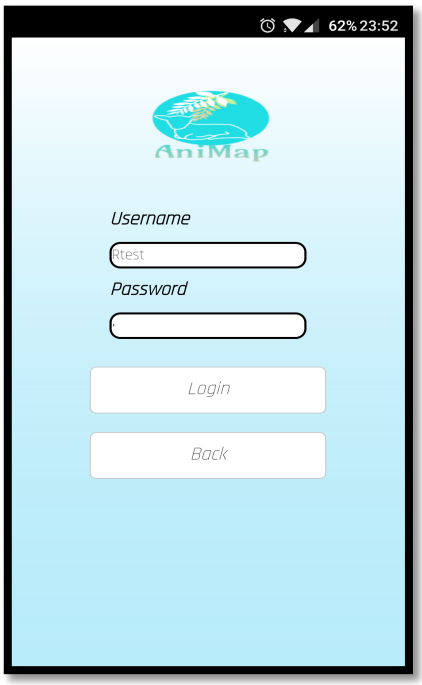


Figure 7. Login screen

2.4.6 Menu screen

As mentioned in the previous use cases the success scenario of each results in a redirect to the Menu screen which will be displayed over two images as the user is required to scroll through the menu to see all the available options.

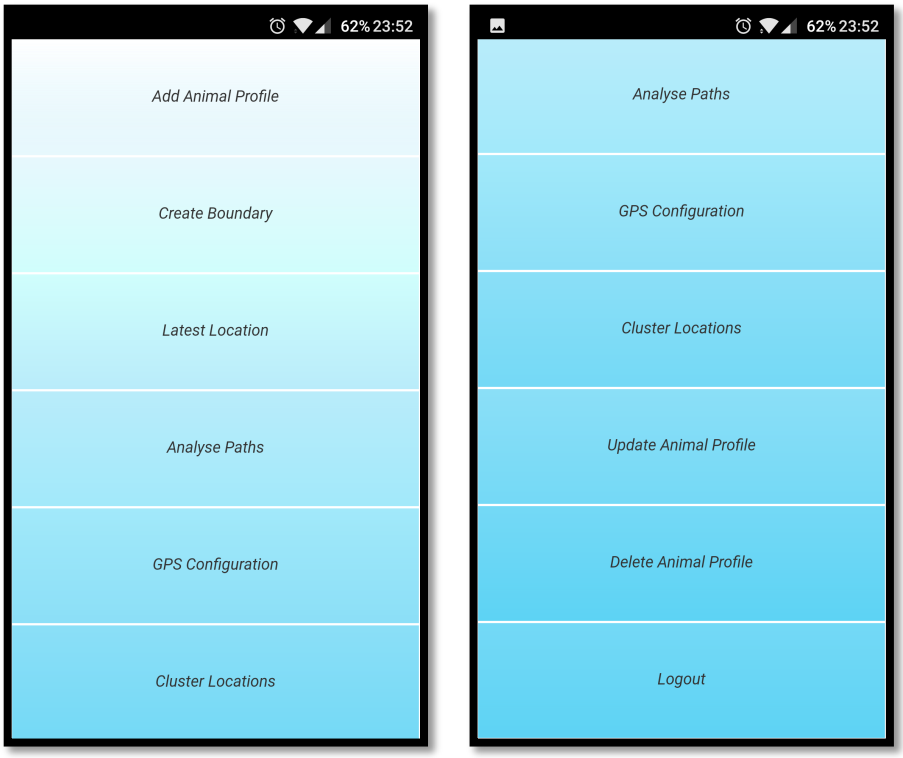


Figure 7. Menu screen

2.4.6.1 Add Animal Profile screen

On selection of the Add Animal Profile option the user is redirected to the Add Animal screen. This particular screen is one of the most densely populated screens in the application in terms of user input. As shown in Figure 8, there are multiple fields for the user to fill in. Firstly the user is required to enter a unique name for the animal, in the case of a herd of cattle, a cows tag number would be an appropriate option. Following this the user is required to select the type of animal from a dropdown list, enter the breed of the animal, enter the animals weight, their gender and the number of the sim card used in the NanoTracker.

Each input field is validated in the backend. As both the weight and tracking number are input fields of type number when the user clicks on this field a numeric keypad is displayed by default to prevent users entering letters.

Similarly to previous functionalities, on submission of the form it is validated to ensure no fields are empty. In a success scenario the animal details are added to the Animal table in the database.

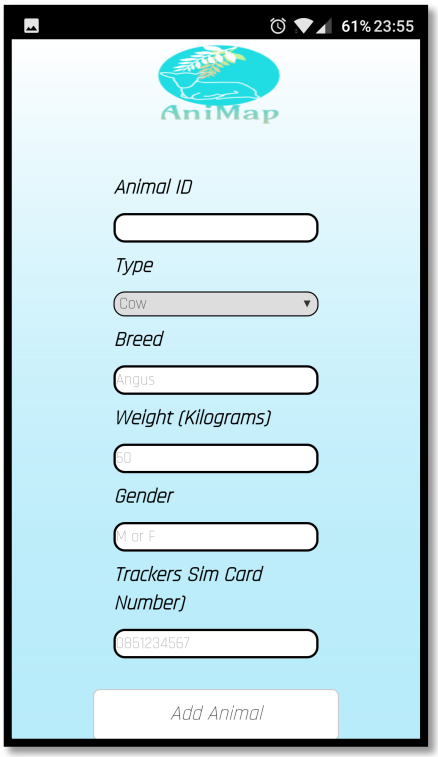


Figure 8. Add Animal screen

2.4.6.2 Create Boundary screen

On selection of the Create Boundary menu option the user is redirected to a screen that is split to show a map and some input fields as shown in Figure 9. If there are no animals associated with their account then they are prompted to add an animal to their account and redirected to the Add Animal screen.

On the map there is a marker which the user can drag to any location they desire. This point on the map will then be used as the center of a boundary that will be defined by the user.

After selecting a point on the map the user is then required to select an animal from the drop down list of animal names associated with their account. Following this they then enter the radius of the boundary they wish to create. When the user clicks the Create Boundary button a text is sent to the NanoTracker containing the coordinates of the boundaries center point and the size of the radius.

Note, although the Create Boundary use case functions as documented above, the entire functionality could not be completed due to incorrect documentation provided by Round Solutions for the NanoTracker.

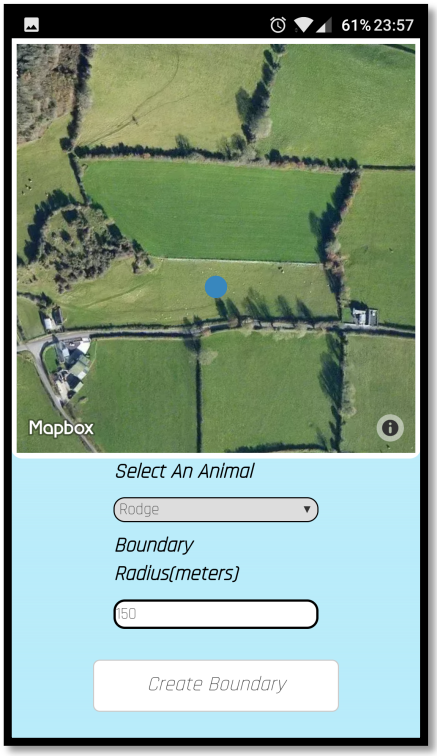


Figure 9. Create Boundary screen

2.4.6.3 Latest Location screen

On choosing the Latest Location menu option, a request is sent to the Flask API that retrieves the latest location of all the animals associated with the user. If no animals are associated with the user they are prompted to add an animal and redirected to the Add Animal screen.

Assuming there are animals associated with the user, a satellite map that is implemented using Mapbox is displayed with each of the animal(s) latest locations as shown in Figure 10. If the user clicks on the animal icon a popup displaying the date and time the location was recorded is displayed.

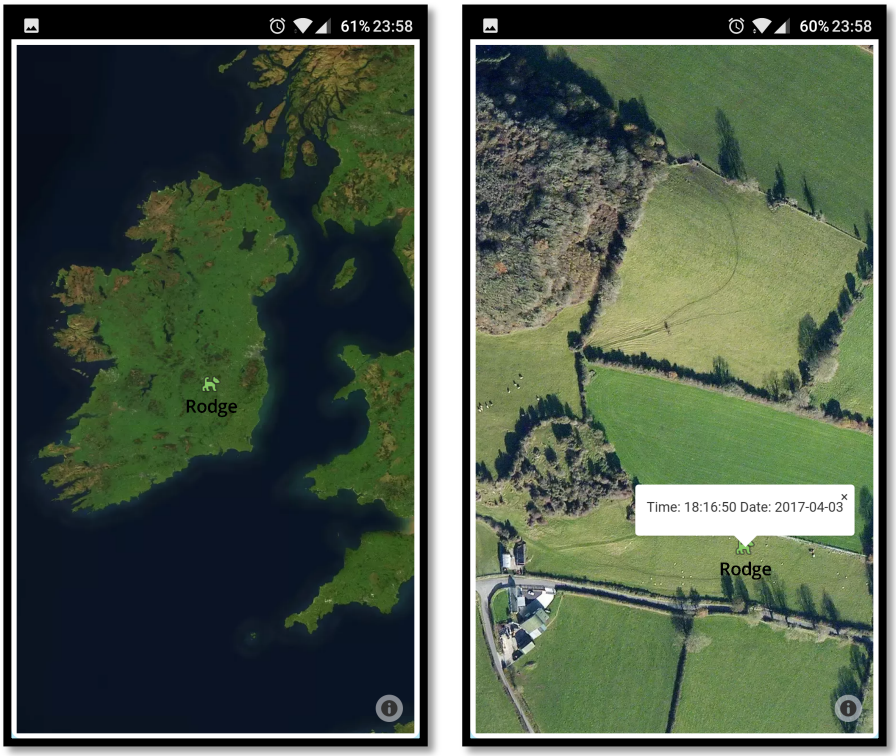


Figure 10. Latest Location screens

2.4.6.4 Analyse Paths screen

If the user clicks the Analyse Paths option, they are then redirected to a page containing four drop down boxes. Like previous use cases if there are no animals associated with the users account then they are prompted accordingly.

In a successful scenario where the user has one or more animals associated with their account they are required to select two animals, a start date and an end date.

If for example the user only wants to view one animals path, they must then select the same animal in both animal selection drop downs as shown in Figure 11. The user must also select a start and end date that will be used to fetch the data. If the user sets an end date that occurs before a start date then they are prompted to re-enter the dates. Similarly to previous use cases a check for empty fields is also carried out.

Once the data has been successfully retrieved a satellite map is displayed which shows the specified animals paths. If the user clicks on an animal icon a popup displaying the date and time the location was recorded is displayed.

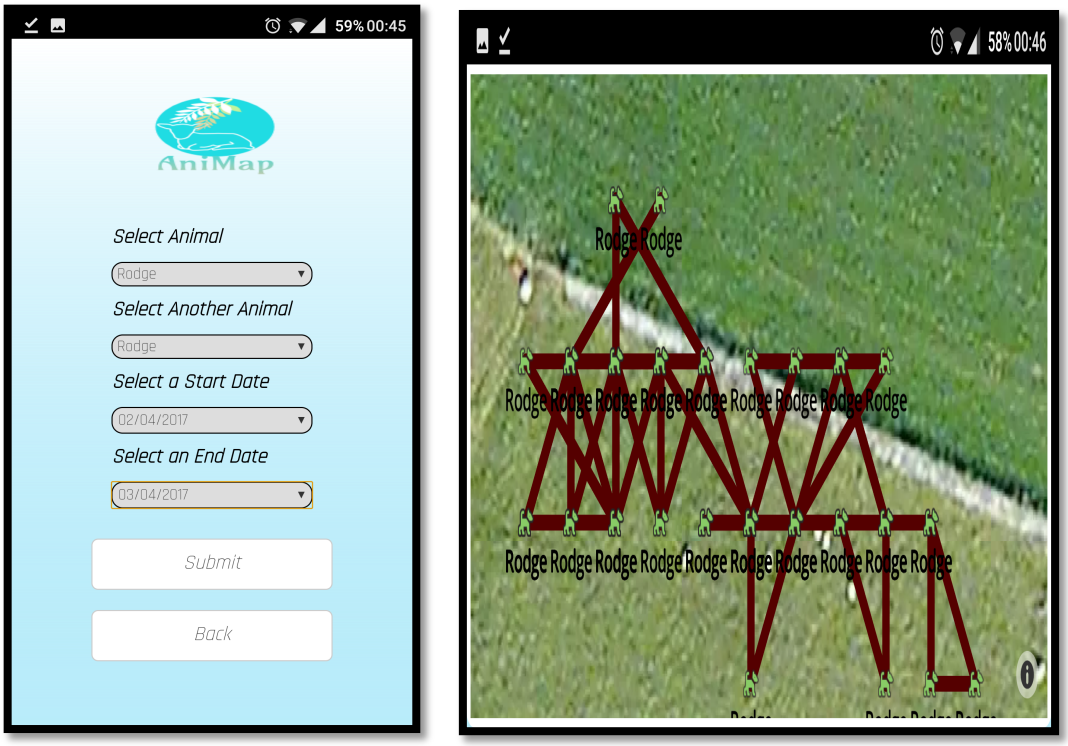


Figure 11. Analyse Paths screens

2.4.6.5 GPS Configuration screen

The purpose of this screen is to remotely configure the NanoTracker via text using a PhoneGap SMS plugin. When the user clicks this menu option they are redirected to a screen that requires them to select an animal, whose tracker they wish to configure, select a time metric and the amount of time they want between GPS requests. An example of user input is displayed in Figure 12.

In a successful scenario the application sends an SMS to the NanoTracker in the form **interval:120,** for example. The NanoTracker then receives this SMS and parses the message to extract everything after the colon and before comma. This value is then used to force the device to sleep, in this case for 120 seconds, between each GPS request.

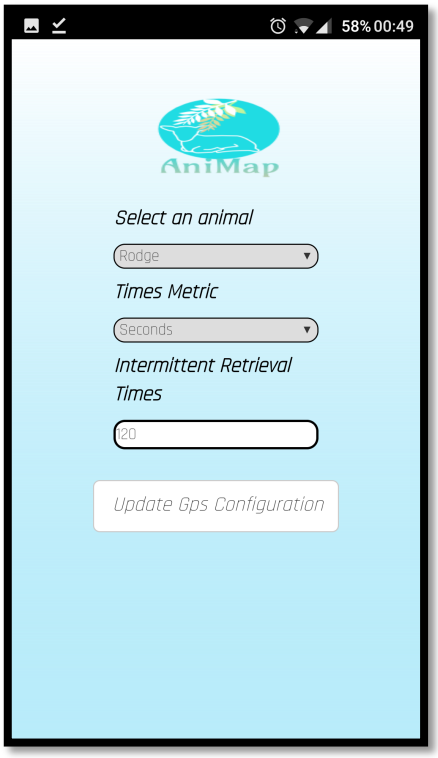


Figure 12. GPS Configuration screen

2.4.6.6 Cluster Locations screen

This functionality could potentially be incredibly useful in determining the source of TB in a herd of cattle, in conjunction with some of the previously mentioned use cases. When the user selects the Cluster Locations option in the Menu, if there is not a sufficient amount of data for the clustering algorithm to take place the user will be alerted to this.

Assuming the user has a sufficient amount of data to perform the clustering of locations they will be redirected to a screen where they will be required to select from two drop down lists. They will also be required to select a start date for the data retrieval. On submission of these inputs the user will be presented with a satellite map which will display the location of each cluster. If the user clicks on a cluster, information about each cluster will be displayed in a pop-up as shown in Figure 13.

In terms of the clustering algorithm, that will be discussed in more detail when documenting the API.

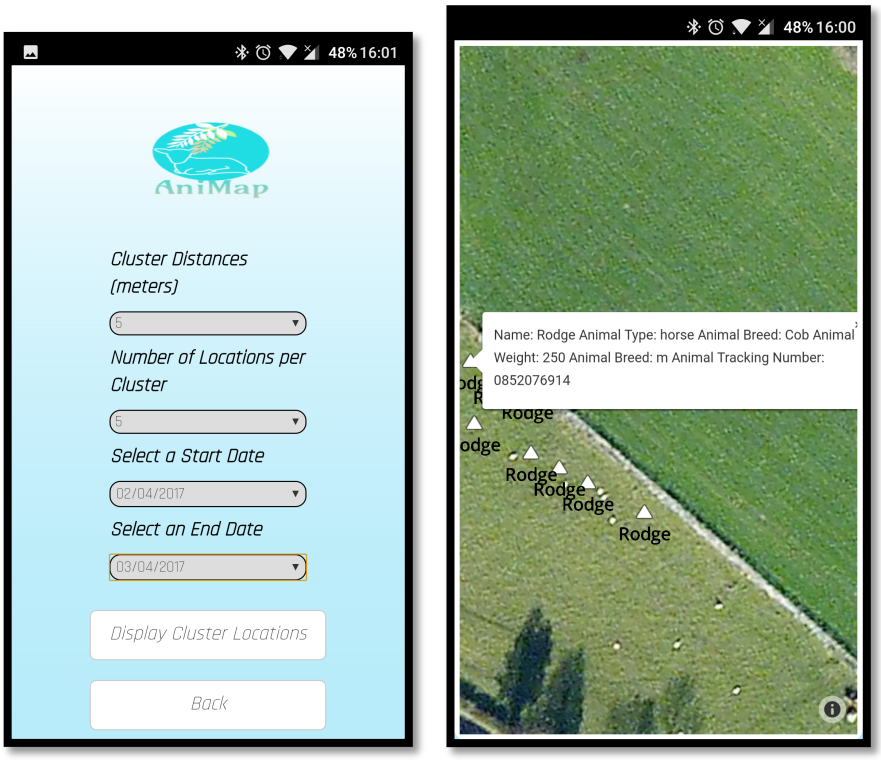


Figure 13. Cluster Locations screen

2.4.6.7 Update Animal Profile screen

As information relating to the animal my change over time, the user has the ability to update the animals profile by selecting the Update Animal Profile option from the menu. The user selects one of the animals associated with their account from the dropdown list which then populates all the fields with the animals information. The information in each of these fields can be changed as deemed appropriate by the application user as shown in Figure 14.

Similarly to previous use cases the user will be alerted appropriately if they attempt to update the animals profile with blank fields.

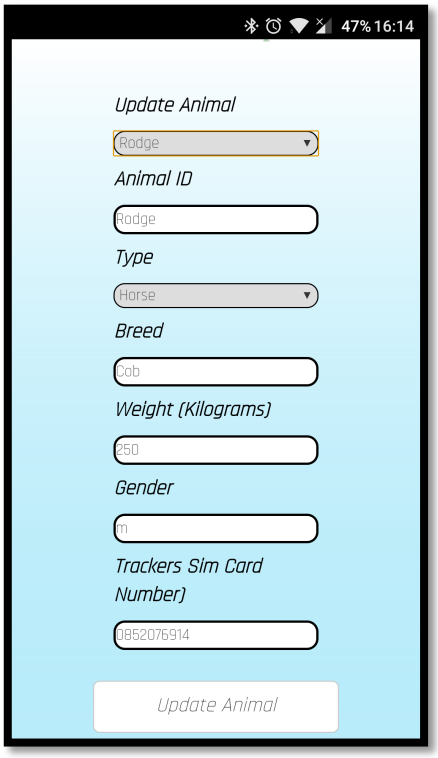


Figure 14. Update Animal screen

2.4.6.8 Delete Animal Profile screen

The Delete Animal Profile use case is very simple in its approach. Once the user selects the Delete Animal Profile option from the menu they are redirected to a screen which requires them to select an animal associated with their account from a drop down list. Once the user clicks the delete button all data relating to this animal from all database tables is removed. This use case is displayed in Figure 15.

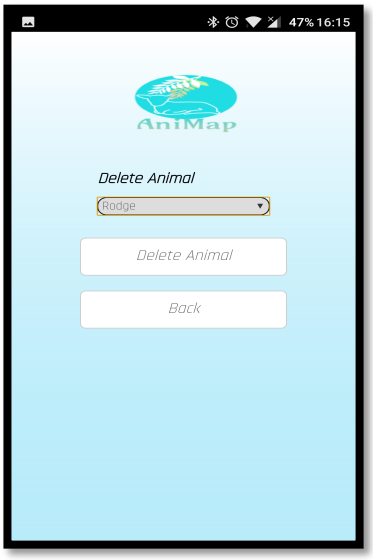


Figure 15. Delete Animal Profile screen

2.4.6.9 Logout Menu Option

Like the Delete Animal Profile use case the Logout use case quite simple in its approach also. Once the user click this button a logged in flag in the Register database table is set to 0. The user is then redirected to the Home screen.

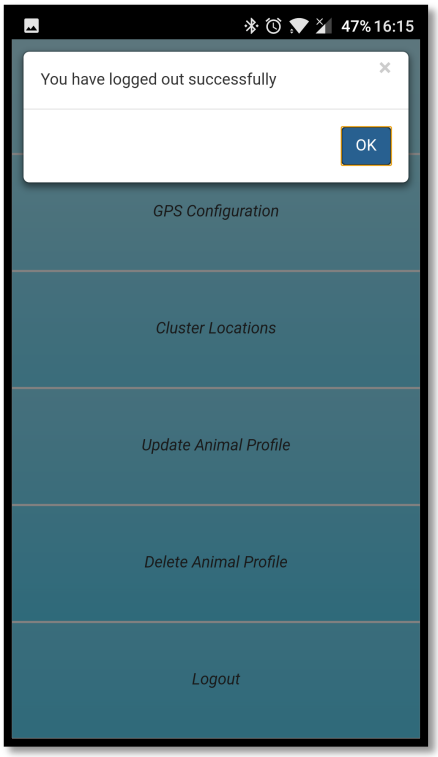


Figure 16. Logout Menu Option

2.5 Flask API

The Python Flask API is a substantial component of the entire system. Not only is the mobile application communicating with the API but the NanoTracker also sends data to the API that is then inserted into the relevant tables in the database. For each of the functionalities mentioned in the previous section, the API checks and maintains an "isLoggenIn" attribute in the Register table which functions as a session value. This column is of type Tiny Int. If the "isLoggenIn" value is 0 it means the user is either logged out or their session has timed out. Therefore before attempting to perform any functions the API checks that the "isLoggenIn" value associated with that user is 1 otherwise they are redirected to the Home screen.

2.5.1 MySQL Database

The types of databases can generally be divided into two categories, SQL and NoSQL databases. SQL databases are traditionally more useful for data that is tabular in nature whereas NOSQL are generally used where data is more unstructured.

For this project a MySQL database was selected and hosted on pythonanywhere. The decision of using a MySQL database was heavily influenced by the tabular structure of the original database design. Although the database design changed over the course of the project, a MySQL database still remained the most suitable overall.

3 Conformance To Specification and Design

As the project progressed it became apparent that there would need to be adjustments made to the functional specification use cases as well as the database design. A description of the changes and the reasoning behind them will be documented in the upcoming sections of this document.

3.1 Use Case Changes

3.1.1 Omitted Use Cases

There was a small selection of use cases that were removed over the duration of the project for various reasons which I will now explain.

3.1.1.1 Isolate Animal Use Case

The Isolate Animal functionality was initially specified so that the application user could add animals to a specific table if the were suffering from some ailment whether it was an illness or injury. This table would then have been used for other functionalities such as the Analyse Paths use case where the user would only be able to look at the paths if the animals in this table.

This use case was omitted as it was an unneccesary step and added unneccesary complications. By removing this functionality it made implementing the Analyse Paths functionality easier and more effective. Instead of only allowing the application view the paths of sick or injured animals, they can now view the path of any animal they wish.

3.1.1.2 Store User Credentials Use Case

This use case was not included in the project as it would currently serve no purpose. The only user credentials stored in the database are the application users login details and animals associated with their account.

3.1.1.3 Retrieve GPS Data Use Case

This functionality does exist in the project but it would technically be a sub function of other use cases. Therefore it should not be included as its own use case.

3.1.2 Amended Use Cases

Again, as the project progressed it became apparent that some use cases needed to be amended. These amendments ranged from a change in use case name to changes in the steps involved in the use cases.

3.1.2.1 Login Use Case

The change to this use case was relatively small. Instead of the application opening to a Login screen it now opens on the Home screen which gives the user three options. They are, Login, Register and Change Password buttons. On clicking any of these buttons the user is then redirected to the appropriate screen as documented in the project description.

3.1.2.2 Register Use Case

The Register use case required two changes. The first change follows the changes made to the login screen where the user is presented with the Home screen and can then navigate to the Register screen by pressing the Register button. The second change was made to the login form which added a confirm password field. The reasoning behind this field was to remove user input error where, in the original use case, the user would create a username and password but wasn't required to confirm that password by entering it again and ensuring the two passwords matched.

3.1.2.3 Logout Use Case

As the application interface did not conform to the description given in the original Logout use case it was amended so that the Logout use case was a button on the main menu. Not only did it make implementing the use case simpler, it more importantly made it easier for the user to logout. Instead of having to click a menu button in the top corner of the screen it is now easily accessible as an option in the Main Menu screen.

3.1.2.4 Change Password Use Case

Like the Register and Login use cases, the navigation to this use case changed. On opening the application the user is presented with the Home screen where they press the Change Password button which redirects them to the Change Password screen. This screen was amended so that it now asks the user to confirm their new password, again to try reduce user input error.

3.1.2.5 Analyse Path Use Case

The Analyse Path use case changed so that the user could select to animal paths to view and also a start date and an end date. By selecting two animals the application user can directly see any common areas traversed by two animals, for example, two cows with TB. By limiting the animal selection to two, it partially prevents them map from being cluttered with animal paths. The inclusion of start and end dates provides the same purpose.

3.1.2.6 Display Current Location Use Case

The first change to this use case is it has been renamed as Latest Location. This use case has also been made simpler and more effective by automatically displaying the latest location of all the animals associated with the user in comparison to the original use case where user input was required to select one animal to view. The final change to this use case allows the user to click the animal icon on the map which displays a popup informing the user what date and time the location was recorded.

3.1.2.7 Set GPS Interval Use Case

The first change to this use case is it has been renamed as GPS Configuration.
One change to the use case is that the user is required to select a time metric from the drop down list. Also this use case now sends the GPS Interval to the NanoTracker device.

3.1.2.8 Store GPS Configuration Use Case

The main change with this functionality is that the Interval is now stored on the NanoTracker device instead of within the application itself. When the GPS Interval is sent to the NanoTracker, the interval is read and stored in a text file on the device. After the NanoTracker sends its coordinates to the Flask API, it opens the text file, reads the value and the device sleeps for that amount of time before sending its coordinates again.

3.1.2.9 Create Boundary Use Case

This use case differs from the original as the user is also required to pick the center of the boundary on the map. The final change to this use case that is not mentioned in the original is, when the user clicks the create boundary button the application then sends the coordinates of the center of the boundary and the radius of the boundary to the NanoTracker.

3.1.3 Additional Use Cases

During the course of the project it became clear that some use cases were required or would useful for application users.

3.1.3.1 Cluster Locations Use Case

This use case was an important addition to the application. The use case functions by requiring the user to select two values from the drop down lists provided. The first value represents the maximum distance a location can be from the center of a cluster. The second value specifies how many of the coordinates that fall within the distance of the center are required to make a cluster. For example if the cluster distance is set to 5 meters and the number of locations per cluster is set to 5 then a satellite map will be displayed show every

cluster that has at least 5 recorded locations within 5 meters of the cluster centroid.

This is use case was implemented using a python package called sklearn. Within this package there are numerous clustering algorithms available. The cluster algorithm used in this project is called DBSCAN (Density Based Spatial Clustering of Applications with Noise).

The algorithm takes the two previously mentioned inputs along with an algorithm name and a distance metric as parameter. The algorithm used in DBSCAN for this project was the ball tree algorithm uses a hyper-sphere data structure. This is a computationally expensive process if the data is now well structured. The ball tree algorithm can be very efficient when there is high dimensionality in the data set. In this application there is low dimensionality as the algorithm is only working with longitude and latitude coordinates. The algorithm works by picking two arbitrary points and making them clusters it then analyses the distance of all neighboring points and assigns them to one of the clusters if the meet the distance criteria.

The distance of the points to the centroid is determined by the Haversine formula. This formula uses the great circle technique to determine the shortest distance between two points on the surface of a sphere, in this case the Earth. The ball tree algorithm repeats the previous steps recursively until all clusters have been formed.

The advantages of using DBSCAN over other clustering algorithms such as k-means, is it doesn't require the number of clusters as a parameter. It determines the number of clusters automatically. Also it allows for data sets with outliers. As the algorithm assigns a label to each coordinate pair to determine which cluster it belongs to, it marks outliers as -1.

In this application the DBSCAN algorithm is used to return an array of cluster coordinates which are then displayed on a satellite map. If the user clicks on a cluster, information regarding that animal is displayed in a popup.

3.1.3.2 Delete Animal Profile Use Case

This use case was introduced to enable a user remove all data relating to an animal or animals from their profile. This would be particularly useful for farmers where animals may only be on the farm for a short period of time before being sold for example. The use case is quite simple in functionality the user simply selects the Delete Animal Profile option from the menu which then redirect them a screen where they select an animal from a dropdown list. The user then clicks Delete Animal and all information regarding that animal is removed from the database tables.

3.1.3.3 Speed Alert Use Case

This use case was introduced to alert a user via text when an animal surpasses a certain speed. This would be particularly useful for sheep farmers where stray dogs may enter a field and begin to chase the sheep. This causes particular stress on the sheep especially with ewes that are in lamb.

3.1.4 Summary of Use Case Adjustments

In summary there were a lot of use case changes mainly due to the authors lack of experience in requirements gathering and application development. Noticeably the changes that were made over the duration of the project often reduced or removed unnecessary complexity and also improved the user interface.

3.1 Database Changes

The changes to the database also removed unnecessary complexity by limiting the number of database tables to three. Those tables are, the Register table, the Animal table and the currentCoordinates table as shown in Figure 17.

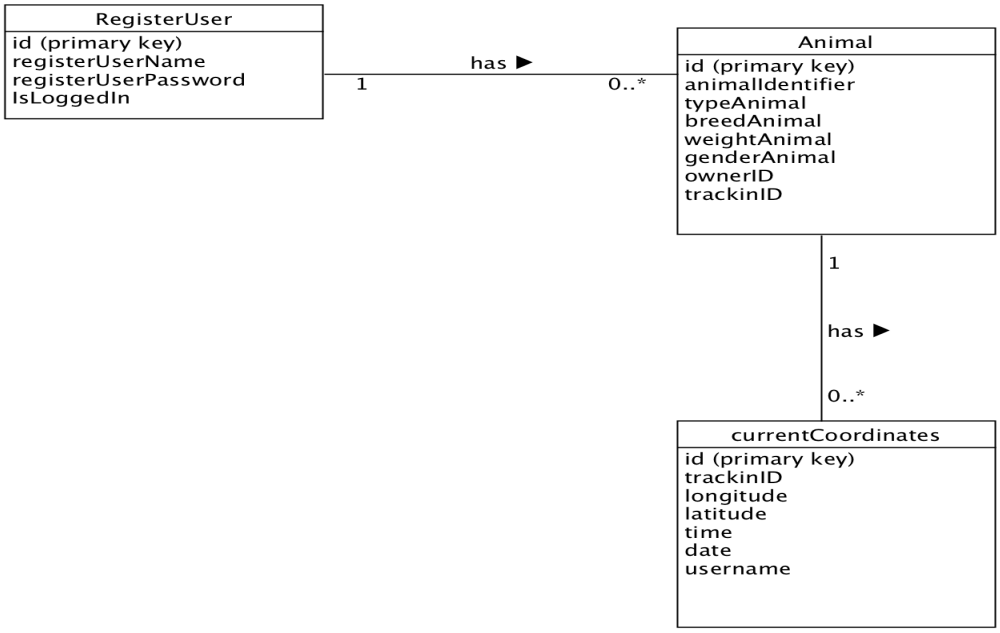


Figure 17. Database Design Diagram

4 Learning Outcomes

4.1 Technical Outcomes

Over the duration of the project I was required to learn and implement various technologies that were new to me. This was initially quite daunting but it was also a source of motivation for me to undertake a project of this scale which I had no prior experience of.

As we only began learning Python from the middle of September I was nervous about using python and more specifically Flask which is a python web framework for my backend. Although I was apprehensive about using Python as my backend technology it soon became apparent that it was quite an easy language to pick up in a relatively short space of time. This was due to Pythons simple syntax and from the experience of learning other languages throughout the course. If the backend had been programmed in Java or C++ there would have been a sizeable increase in complexity in my opinion.

Python wasn't the only technology that was new to me in this project. Despite having used JavaScript before I had never used jQuery. Initially I found it difficult to work with jQuery but as the project progressed it become more apparent how useful a library it is. Adobe PhoneGap was also a new technology to me but due to the vast quantity of information available through their documentation and other online sources, PhoneGap provided very few problems throughout.

Working with the NanoTracker was also completely new to me. This proved to be the most difficult aspect of the project due to the lack of documentation and also some crucial errors in the NanoTrackers documentation provided by Round Solutions.

Overall, I have gained a lot of knowledge in several different technologies that I would otherwise not have had the chance to work with. Undertaking the project

has also proven to myself that I can complete projects of this scale. If I was to undertake the Project again I feel I could get a better end product in a shorter time frame which is a successful outcome in my opinion.

4.1 Personal Outcomes

At the beginning of this project I was very apprehensive about my ability to complete a project of this scale based on my lack of experience in application development. Although programming modules are undertaken in each of the four year throughout the Software Engineering course they cannot prepare you for a project of this scale.

By completing a project of this scale, I have proven to myself and others that I am capable of undertaking a large and complex piece of work and producing the required deliverables. The project provided me with multiple challenges throughout which made the whole process more interesting and therefore making me more productive.

Aside from gaining knowledge in new technologies I have also learned to organise my work more efficiently. Time management is crucial when dealing with the project and the fourth year course in general. I have also learned to work more independently during this project. Its often the case with students that they go directly to a lecturer for example, in search of an answer. While there is nothing wrong with this it is also beneficial to learn to conduct research on your own. Personally I found that I learned a lot more about the technologies by adopting this approach.

In hindsight I should have started programming even before iteration one started. Personally I found a lot of time was spent at the beginning of iteration two trying to implement some basic backend functionalities whereas had I begun programming earlier i.e. before iteration one, I could have been more productive in iteration two and three.

I would also advise any students undertaking this course next year to start working on project from day one and give it the time that it needs. The further you progress in the year the more deadlines there are for each subject, so having a good start at project will prove beneficial in the latter half of the year.

5 Project Review

5.1 What Went Right

In general I think the project was successful. As it was a proof of concept project and the concept I was trying to prove was that the source of TB in herd of cattle could be determined by analyzing their locations, it was always going to be difficult to gauge the success. To adequately determine if the project was successful would require having each cow in a herd of cattle being tracked for at least a year as cattle are tested annually for TB. Hence it is quite difficult the gauge the success of this project.

Pythonanywhere was a great addition to the project. Being able to host your API and MySQL database for free was an excellent aspect of pythonanywhere. It must also be noted that setting up a project to run on pythonanywhere probably couldn't be made any simpler. This was a huge advantage during the project as it meant less time was required to configure pythonanywhere which meant more time could be dedicated to overall project.

Working with the NanoTracker was not a straightforward process but it does function as intended. There were some distinct advantages to using the NanoTracker. One of those advantages was it was programmed in Python and came with a software development kit. Another advantage was the NanoTrackers compact size. When putting a device on an animal regardless of its purpose it should also be taken into consideration how it fits the animal. Again as the device was so small this never proved to be a problem.

5.2 What Went Wrong

Over the duration of the project there were numerous problems encountered most of which were in relation to the NanoTracker. The first of these problems occurred in the very early stages when I received the NanoTracker. The device is interfaced via USB. When I plugged the device into the USB ports located on the top of my desktop case and tried to upload scripts to it on RS-Term I was presented with an error "Module not connected…please check". This error lasted for approximately a week. After emailing Round Solutions about they were unable to tell me the cause of the error. The solution to this error happened by chance when I plugged the device into a USB port at the back of the desktop which then allowed me to interface with the device correctly. Although this was not documented in the NanoTracker manual I was told that this occurred because the USB ports at the back of the computer are connected directly to the motherboard.

One of the most substantial and consequential errors encountered relating to the NanoTracker was caused by incorrect documentation for the NanoTracker. The NanoTracker datasheet clearly states that the 2g Module is implemented using python version 2.7.2. One questioning the Round Solutions staff regarding using external Python packages on the NanoTracker, I was told that the package must be compatible with Python version 1.5.2 and uploaded in .pyc format.

Following this I questioned why the device required external Python packages to be compatible with Python version 1.5.2 despite it being documented that the NanoTracker was running Python version 2.7.2. To my surprise I was told that I had spotted an error in the documentation and that the device was actually running Python version 1.5.2.

This error proved very costly for application. It meant that my Create Boundary functionality now serves no purpose as I cannot implement the Haversine formula on the NanoTracker device due to the math module in Python version 1.5.2 not supporting radians. Had Round Solutions not made this error the

Create Boundary functionality would have been implemented where the NanoTracker would send a text to the user when an animal went out of bounds.

Another problem was encountered when implementing the Speed Alert use case. This problem was again a direct consequence of inadequate documentation provided by Round Solutions. The problem on this occasion was that nowhere in the NanoTracker manual did it say that the device did not support floating point numbers.

When getting the speed, the speed value is returned in knots but in string format. To use this value I could only take everything up to the decimal point meaning a loss of precision. Also as 1 knot is equal to 1.85 kph it meant I had to let 1 knot equal 2 kph to avoid using floating point numbers. Again this meant a loss of precision. As this error and the previous error occurred very late in the project where time was an issue I was left incredibly frustrated by Round Solutions standard of documentation.

On several occasions throughout the course of this project I noticed that Round Solutions had been updating their documentation in accordance with questions I had been asking them. Although this is a positive, they did in turn make another error in the process of updating the documents to reflect my questions by leaving out an entire section on how to update the micro-controller code. Again I noticed this error several weeks later when I attempted to update the micro-controller code to prevent the accelerometer from causing the device to sleep after 10 minutes of inactivity. When I questioned them about how to update the micro-controller code they informed me that they had left out this section in the previous update. This is an important section of the document as it informs the user that they need to by a special cable and accompanying device for debugging the micro-controller. Fortunately I have not needed to update the micro-controller code as the movement of the animals head when they are grazing is enough to keep the device awake.

Finally another problem with this project has been the lack of unit testing. This was due to time constraints. Had I undertaken unit testing, which I have never done before, the time spent learning unit testing would have taken from the functionalities in the end product.

Although I encountered a substantial number of errors during the project I have still gained a lot of knowledge in each of the technologies. A lot of this knowledge was gained by encountering errors, determining what caused them and devising a solution to solve them.

5.3 Future Feature

There is one future feature that is not implemented in my project that I feel could be incredibly useful. In my opinion I think it could be possible to preemptively diagnose TB in a herd of cattle by analysing accelerometer data to determine when a cow has coughed and the frequency between coughs. This functionality could be very effective as it would alert the farmer to the animal showing signs of TB. The farmer could then use this information to isolate the animal to prevent the TB spreading throughout his/her herd which provides the farmer with more financial security as losing an cow to TB is quite costly.

6 Acknowledgements

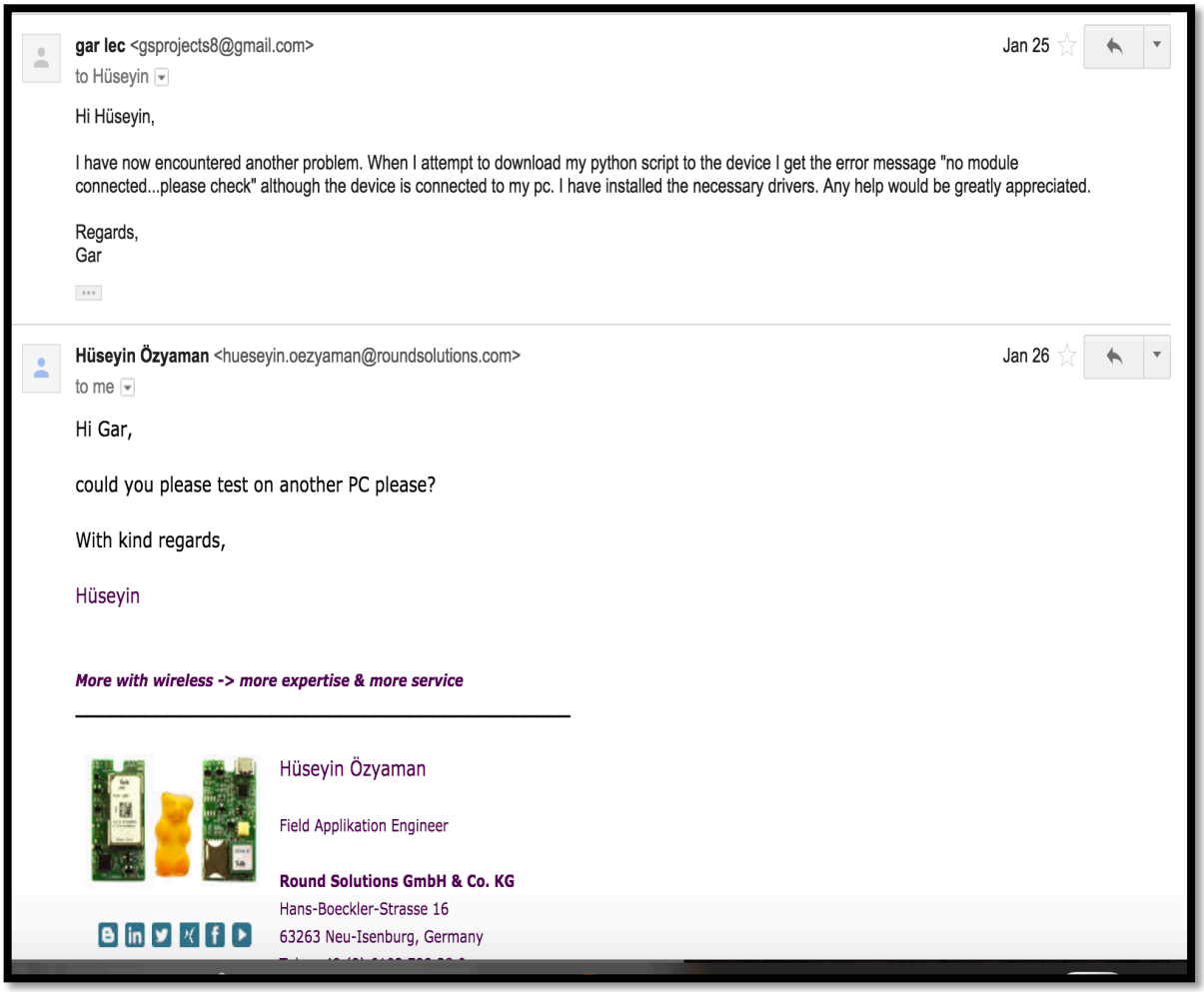
Firstly I would like to thank my project supervisor, Nigel Whyte for all his help and guidance over the duration of the project.

I would also like to thank my classmates for offering their opinion regarding the best course of action to take when issues arose in my project.

I would like to thank the Round Solutions staff for answering my numerous questions and providing me with vital information over the duration of the project

Finally I would like to thank the lecturing staff for fielding any question I may have asked throughout the year regarding my project.

Appendix A. USB Port Complication Correspondence



gar lec <gsprojects8@gmail.com> Jan 25

to Hüseyin

Hi Hüseyin,

I have now encountered another problem. When I attempt to download my python script to the device I get the error message "no module connected...please check" although the device is connected to my pc. I have installed the necessary drivers. Any help would be greatly appreciated.

Regards,
Gar

Hüseyin Özyaman <hueseyin.oezyaman@roundsolutions.com> Jan 26

to me

Hi Gar,

could you please test on another PC please?

With kind regards,

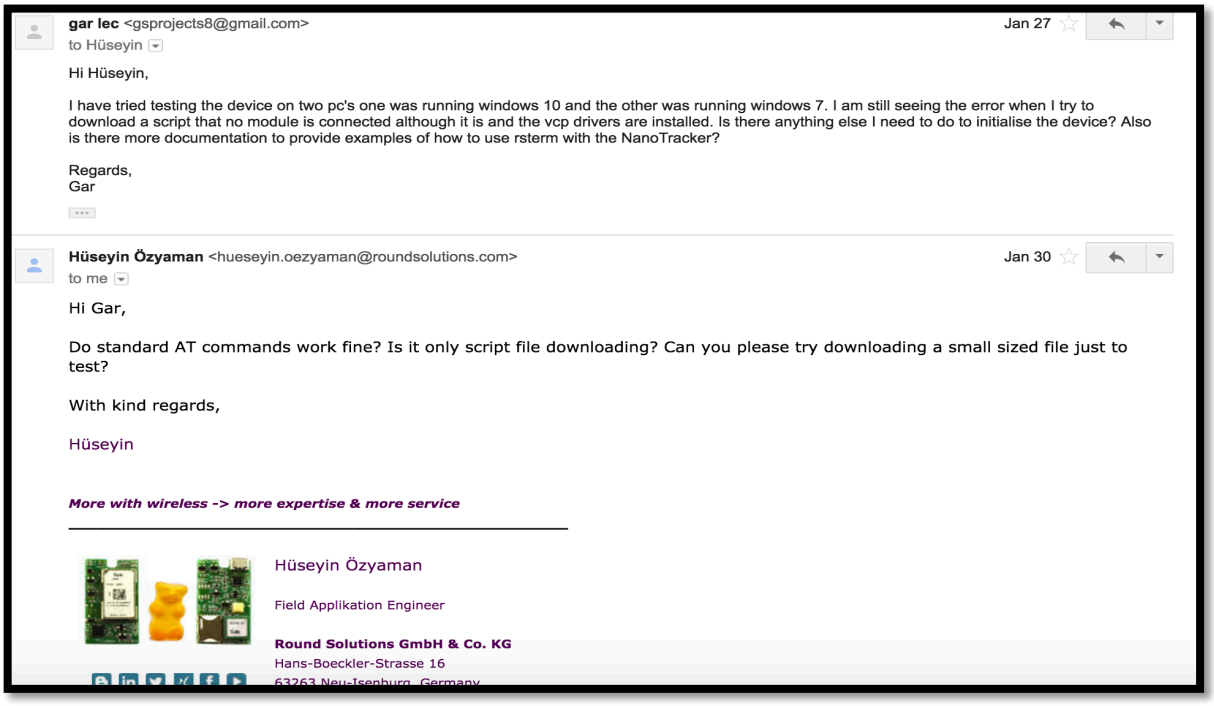
Hüseyin

More with wireless -> more expertise & more service

Hüseyin Özyaman

Field Applikation Engineer

Round Solutions GmbH & Co. KG
Hans-Boeckler-Strasse 16
63263 Neu-Isenburg, Germany



gar lec <gsprojects8@gmail.com> Jan 27

to Hüseyin

Hi Hüseyin,

I have tried testing the device on two pc's one was running windows 10 and the other was running windows 7. I am still seeing the error when I try to download a script that no module is connected although it is and the vcp drivers are installed. Is there anything else I need to do to initialise the device? Also is there more documentation to provide examples of how to use rsterm with the NanoTracker?

Regards,
Gar

Hüseyin Özyaman <hueseyin.oezyaman@roundsolutions.com> Jan 30

to me

Hi Gar,

Do standard AT commands work fine? Is it only script file downloading? Can you please try downloading a small sized file just to test?

With kind regards,

Hüseyin

More with wireless -> more expertise & more service

Hüseyin Özyaman

Field Applikation Engineer

Round Solutions GmbH & Co. KG
Hans-Boeckler-Strasse 16
63263 Neu-Isenburg, Germany

gar lec <gsprojects8@gmail.com> Jan 30

to Hüseyin

Hi Huseyin,

When I have the python tab open in rsterm and I press the AT button a message appears in the dialog box saying "Send: AT<cr>" which i believe is supposed to produce the message "ok". This message never appears. I also tried downloading a smaller file which also didnt work, the message "no module connected ... please check!" still appears.

When downloading the setup executable for the vcp driver, I installed it following the on screen instructions. At the end of this process I was notified that I had installed the two ftdi drivers and that they were ready to use. Was I supposed to install anything else or is installing that setup executable enough?

Regards,
Gar

gar lec <gsprojects8@gmail.com> Jan 31

to Hüseyin

Hi Hüseyin,

Just to let you know I seem to have got the device working by plugging the device directly onto the motherboard instead of a usb port thats built into the case and wired to the motherboard. Is there any reason for this?

Regards,
Gar

Hüseyin Özyaman <hueseyin.oezyaman@roundsolutions.com> Jan 31

to me

Hi Gar,

I'm glad that it works now.

The ports on motherboard are connected directly to the USB controller.
Connectors on the front are extended with cables, where you did not know the quality of the cables and they must support signal frequencies of up to 480 MHz for high-speed connection. You could ask your computer supplier why they used low quality cables.

With kind regards,

Hüseyin

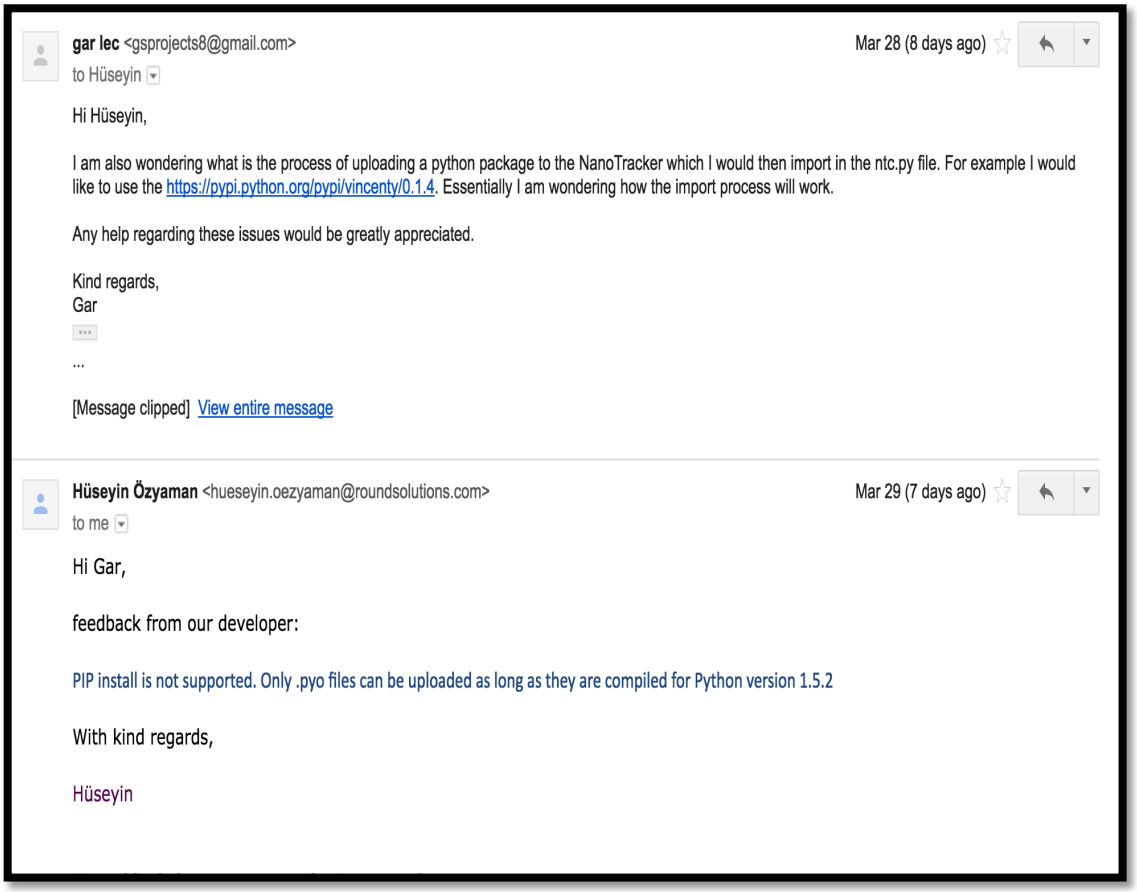
More with wireless -> more expertise & more service

Hüseyin Özyaman

Field Applikation Engineer

Round Solutions GmbH & Co. KG
Hans-Boeckler-Strasse 16
63263 Neu-Isenburg, Germany
Tel.: +49 (0) 6102 799 28 0

Appendix B. Incorrect Python Version Correspondence



gar lec <gsprojects8@gmail.com> Mar 28 (8 days ago)

to Hüseyin

Hi Hüseyin,

I am also wondering what is the process of uploading a python package to the NanoTracker which I would then import in the ntc.py file. For example I would like to use the https://pypi.python.org/pypi/vincenty/0.1.4. Essentially I am wondering how the import process will work.

Any help regarding these issues would be greatly appreciated.

Kind regards,
Gar

...

[Message clipped] View entire message

Hüseyin Özyaman <hueseyin.oezyaman@roundsolutions.com> Mar 29 (7 days ago)

to me

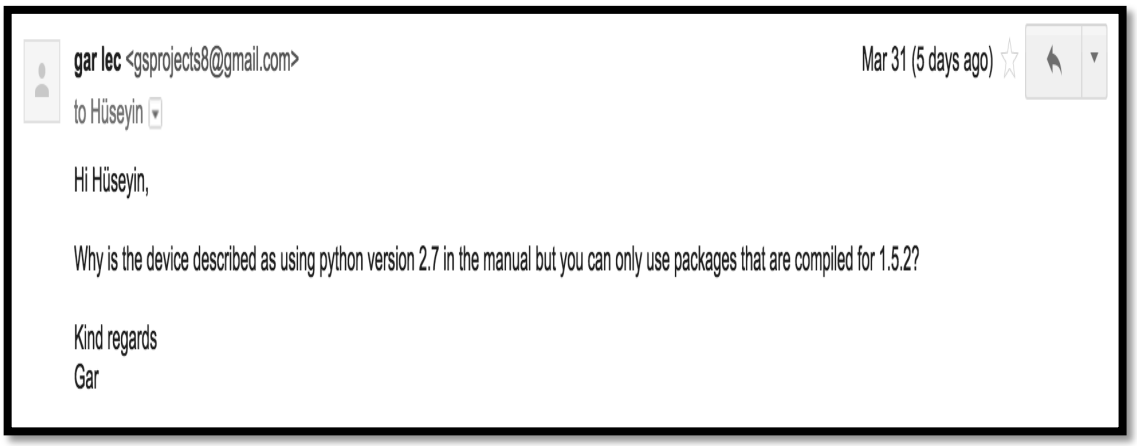
Hi Gar,

feedback from our developer:

PIP install is not supported. Only .pyo files can be uploaded as long as they are compiled for Python version 1.5.2

With kind regards,

Hüseyin



gar lec <gsprojects8@gmail.com> Mar 31 (5 days ago)

to Hüseyin

Hi Hüseyin,

Why is the device described as using python version 2.7 in the manual but you can only use packages that are compiled for 1.5.2?

Kind regards
Gar

Hüseyin Özyaman <hueseyin.oezyaman@roundsolutions.com> Mar 31 (5 days ago)

to Mazen, me

Hi Gar,

yes, you discover an mistake ☺. The 2G variant has only 1.5.2 we will improve with next update.

Have a nice weekend and kind regards,

Hüseyin

More with wireless -> more expertise & more service

Hüseyin Özyaman

Field Application Engineer

Round Solutions GmbH & Co. KG

Hans-Boeckler-Strasse 16

63263 Neu-Isenburg, Germany

Tel.: +49 (0) 6102 799 28 0

www.roundsolutions.com

Appendix C. Missing Micro-Controller Documentation

Von: gar lec [mailto:gsprojects8@gmail.com]
Gesendet: Samstag, 18. März 2017 16:19
An: Hüseyin Özyaman
Betreff: Re: Python API

`Hi Hüseyin,

I have read the instructions that are provided in the source code for the micro controller and implemented the changes I wanted to make but I don't know how to update the micro controller with the new source code. I have looked at http://www.roundsolutions.com/media/pdf/GE866Q_Telit_Modules_Software_User_Guide_r16(10).pdf and more specifically at the firmware update section of the document. Is this what I should be looking at or where exactly is the documentation that describes the process of updating the micro controller with new code.

Kind regards,
Gar

Hüseyin Özyaman <hueseyin.oezyaman@roundsolutions.com> Mar 20

to me

Hi Gar,

For some reason we had a mix up in the updated manual and data was missing. The information requested is found in the attached manual.

With kind regards,

Hüseyin

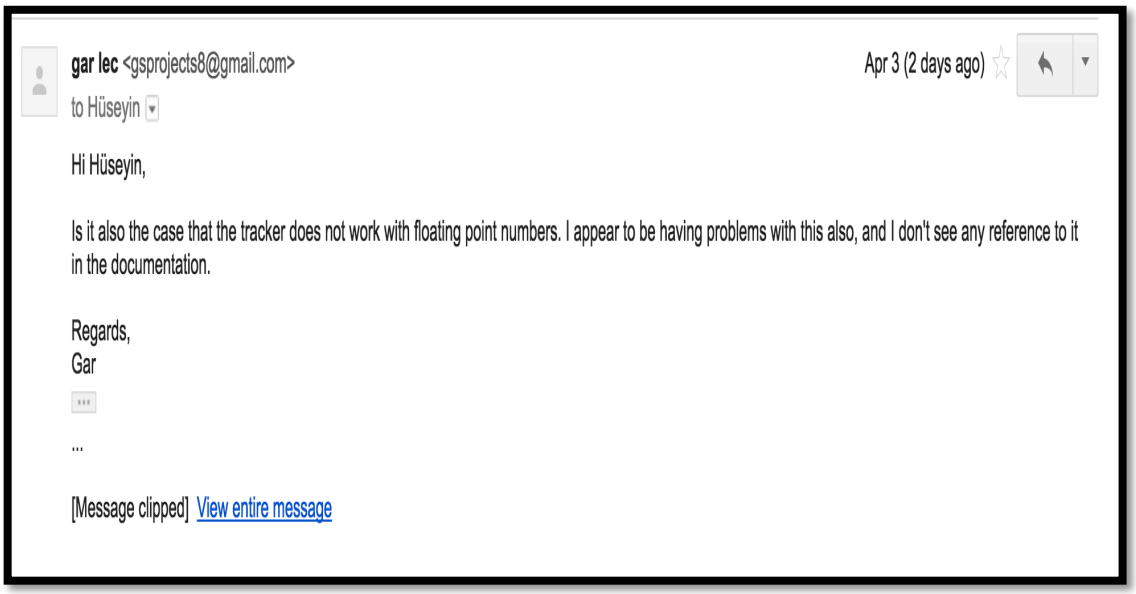
More with wireless -> more expertise & more service

Hüseyin Özyaman

Field Applikation Engineer

Round Solutions GmbH & Co. KG
Hans-Boeckler-Strasse 16
63263 Neu-Isenburg, Germany
Tel.: +49 (0) 6102 799 28 0
www.roundsolutions.com

Appendix D. Floating Numbers Support Correspondence



gar lec <gsprojects8@gmail.com> Apr 3 (2 days ago)

to Hüseyin

Hi Hüseyin,

Is it also the case that the tracker does not work with floating point numbers. I appear to be having problems with this also, and I don't see any reference to it in the documentation.

Regards,
Gar

...

[Message clipped] View entire message

Hüseyin Özyaman <hueseyin.oezyaman@roundsolutions.com> 8:49 AM (16 hours ago)

to me

Hi Gar,

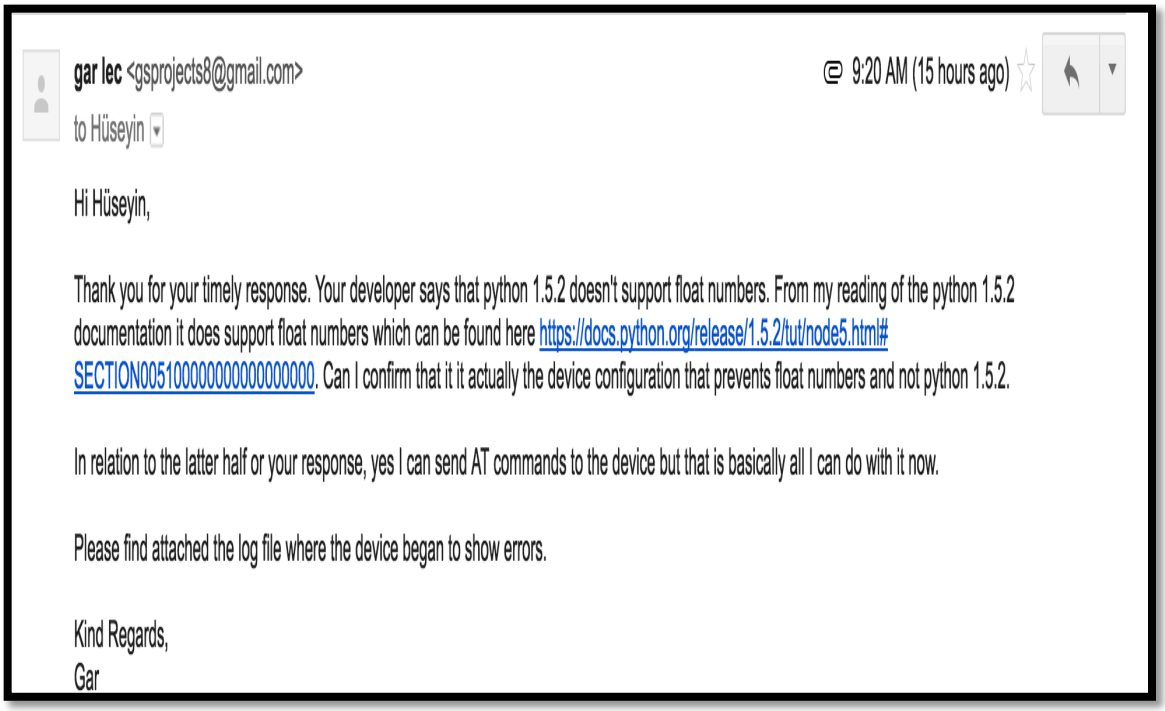
answer from our developer:

Yes it is correct that version 1.5.2 doesn't support floating point numbers. The easiest way to work with this is to multiply the number by 100 or a 1000 and work with it to take into account the numbers after the digital point.

Can we have the log file from RSTerm to see what is going on? Is the client able to send AT commands to the device?

With kind regards,

Hüseyin



gar lec <gsprojects8@gmail.com> 9:20 AM (15 hours ago)

to Hüseyin

Hi Hüseyin,

Thank you for your timely response. Your developer says that python 1.5.2 doesn't support float numbers. From my reading of the python 1.5.2 documentation it does support float numbers which can be found here https://docs.python.org/release/1.5.2/tut/node5.html#SECTION005100000000000000000. Can I confirm that it it actually the device configuration that prevents float numbers and not python 1.5.2.

In relation to the latter half or your response, yes I can send AT commands to the device but that is basically all I can do with it now.

Please find attached the log file where the device began to show errors.

Kind Regards,
Gar