8

Project closure report

Project name: IABee

Team number: 66

Project partner: BHCorps - Ismaïl Ben Hariz

Prepared By

Document Owner(s)	**Project Role (if defined)**
Tobias OHANA	
Victor MOURADIAN	
Paul PICHLAK	
Wassim SERRADJ	
Virgile PROCUREUR	

TABLE DES MATIERES

PROJECT CLOSURE REPORT PURPOSE

The purpose of this project report will be to synthesize the various information important to the understanding of the realization of this project. In this report we will discuss the methods of organization of the team, the technical reports explained as well as the difficulties encountered by the team. This report will aim to give the maximum information for the client as well as for the team that will take over the project so that they are put in the best conditions for its success.

1 Background

This report has been made according to all the tasks carried out by the team and taking into account all the data of the productions made during the year. All the links, PRM report, schedules and documents helping to understand will be discussed.

We will also give additional information on the tasks to be carried out and the procedure to follow in order to carry them out in the best possible way, also specifying what was expected by the client and what we were able to provide him with.

2 Project Descriptive Reminders

The aim of the project was to realize a mobile application to detect queen bees in a hive. Another task was to realize a functional connected hive with different options depending on what already exists on the market.
The team consists of :
Victor MOURADIAN: student in IBO.
Tobias OHANA: student in NE.
Paul PICHLAK : student in MNM.
Virgile PROCUREUR: student in IBO.
Wassim SERRADJ : student in NE.
The client of the project is Ismaïl Ben Hariz, the PRM is Antinea Salerno-Penciocelli.
The aim of the project was the realization of an application having as main function to detect a queen bee as well as the realization of a connected hive. These projects are part of a set of projects that the client wishes to realize for a start-up company of products dedicated to beekeepers.

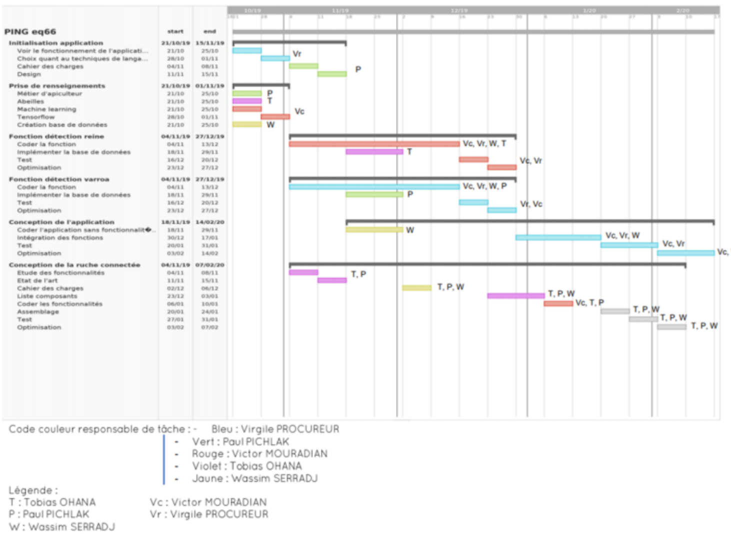
3 Objectives Set versus Results Achieved

3.1 Project Initial Objectives

The initial objectives of the project that we were given were as follows: To make a mobile application, running on Android and IOS, to detect the queen bee in a hive and to detect whether a bee is infected with varroa or not.

We had set ourselves an initial schedule, validated by the business partner. In this planning is specified that we have designed a connected hive. This was not foreseen in the initial subject but our pi² team proposed this extension of the subject to the partner. The business partner has accepted that we work on the connected hive on condition that the other deliverables are made available on time.

<u>First planning :</u>



In terms of technical specifications, we were free to work on the software we wanted. This freedom led us to a research phase to find out in which language we were going to code the application.

3.2 Results Achieved

Here are the main results we achieved at the end of this project:

The mobile application is functional on Android with queen bee recognition. We have a prototype of a connected hive that allows us to collect various data such as temperature or weight and can upload them at regular intervals on TheThingsNetwork only if a LoraWan network is avalaible.

<u>Second planning :</u>

02-déc					
Codage fonction reine	10	0	0	10	0
Alimentation BDD	2	8	8	0	8
Ruche connectée	0	2	2	0	4
Contact apiculteur	0	2	2	0	2
Total	12	12	12	10	14
09-déc					
Codage fonction reine	5	0	0	5	0
Alimentation BDD	3	5	5	0	4
Ruche connectée	0	0	0	0	4
Total	8	5	5	5	8
16-déc					
Codage fonction reine	5	0	0	5	0
Alimentation BDD	3	5	2	0	4
Ruche connectée	0	0	0	0	4
Total	8	5	2	5	8
06-janv					
Codage fonction reine	8	0	0	8	0
Ruche connectée	0	0	0	0	5
Codage application	0	7	7	0	7
Total	8	7	7	8	12
13-janv					
Codage fonction reine	8	0	0	8	0
Ruche connectée	0	0	0	0	5
Codage application	0	7	7	0	7
MBTI	1	1	1	1	1
Réunion	1	1	1	1	1
Total	9	8	8	9	13
20-janv					
RDV PRM	2	2	2	2	2
Préparation rdv	2	2	3	4	3
Codage fonction reine	8	0	0	4	0
Ruche connectée	0	0	0	2	5
Codage application	0	7	7	0	4
CR rdv	0	0	0	2	0
Total	12	11	12	14	14
27-janv					
Codage fonction reine	8	0	0	6	0
Ruche connectée	0	0	0	2	5
Codage application	0	7	7	0	4
Total	8	7	7	8	9
03-févr					
Codage fonction reine	8	0	0	6	0
Ruche connectée	0	0	0	2	5
Codage application	0	7	7	0	4
Total	8	7	7	8	9
10-févr					
Codage fonction reine	10	0	0	8	0
Ruche connectée	0	0	0	2	8
Codage application	0	10	10	0	4
Total	10	10	10	10	12
17-févr					
Codage fonction reine	6	0	0	6	0
Ruche connectée	0	0	0	2	6
Codage application	4	5	5	0	4
Total	10	5	5	8	10

As you can see, we decided to focus on the queen bee detection function and we dropped the varroa detection function because it was going to take too much time. We preferred qualitative work to quantitative work, as the partner company informed us that the work would be continued by another team next year. Another problem we encountered was the change of programming language for the mobile application. Indeed, at the beginning we were using React Native. But we realized that it was very complicated (if not impossible) to access the phone components with this language (our application needed to access the smartphone's camera). So we changed our programming language, we switched to Native. The application is functional, however we wanted to set up a menu ("Navigation bar") so that we don't just have the function of detecting the queen bee when opening the application. Our application being already coded, to have a navigation bar we have to convert the "activity" into "fragments". In other words, we would have had to rethink the code structure of our application, which we didn't have time to do since this problem appeared at the end of the project.

The corporate partner and PRM have indicated that they are satisfied with the progress of our work and our objectives during the year.

3.3 List of Deliverables

- Distribution of individual work time (30/03/2020)
- PRM Meeting & Progress Report (19/03/2020)
- Connected beehive documentation (14/02/2020)
- Project report 1st semester (01/12/2019)

4 Technical Review

4.1 The app

As mentioned in the previous point, for the coding of the mobile application we decided to choose Native language on Android Studio in order to have a multitude of possibilities and not to be blocked by the limits of the programming language. The language we started with initially, React Native, doesn't allow us to access the phone components (we needed to access the camera).

4.2 The queen bee detection function

The queen bee function is a TENSORFLOW model of real-time object recognition. In order to realize the model we decided not to start from scratch. We used the technique called "transfer learning", which allows us to start from a model that has already been trained on images and therefore knows its "work" (i.e. recognize objects big and small).
This saved us a lot of time on the choice of the architecture, the different parameters etc... Moreover the fact of starting from an already trained model allows us to have satisfying results quickly.

At the beginning of the project we decided to start with a YOLO (You only see once) architecture with a training of the model on the COCO dataset (70 000 images), so we had a model able to recognize queen bees quickly and with a good precision.

Détection réalisé sur l'ordinateur.

But, when we chose YOLO we had forgotten an important aspect of our problem, the use on mobile phones. There are two ways to use it on a mobile phone:

- Online, with a permanent sending of images taken by the smartphone to online servers on AWS or Google Cloud, thus requiring skills that we did not have. But also our application being intended for beekeepers, a permanent connection to the internet could be a problem.
- Offline, with Tensorflow lite. The smartphone takes care of the detection itself using its internal components and the Tensorflow lite model supplied.

Due to our problem we decided to go offline, but the architecture of the model is not yet compatible with Tensorflow Lite. So we had to revise our plans and start on another solution.
The SSD MobileNet v3 architecture, the model is more powerful, lighter and of course compatible with Tensorflow Lite. After starting from what we had done for the YOLO model, we managed to get a model being able to detect queen bees on smartphone.

Recognition realised on a smartphone

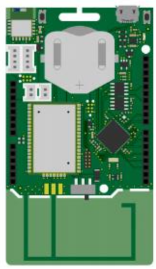
If you want more technical information about the realization I invite you to visit the project repository on github.

4.3 Connected Hive:

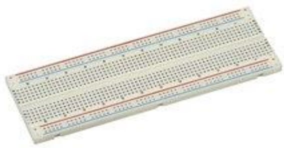
For the creation of the connected hive, the company BHCorp gave us a free hand to define its characteristics. So we started by meeting amateur and professional beekeepers, trade unions and people interested in beekeeping in order to define the needs of our potential customers. We also made a state of the art of the current products.
After studying the needs and problems encountered, we defined that access to free distance learning to become a beekeeper and the climate emergency are the 2 main problems we want to address.
For this, we want to develop a connected hive with temperature, humidity and weight sensors. These data will be analyzed by AI and Big Data in order to propose on our application to the user adapted actions to be carried out accompanied by video tutorial.
To develop this hive we had several constraints:
- Bees do not support continuous high frequencies.
- We have to develop a low-cost product to help beekeepers in the best way possible
- We won't be able to connect our hive to a power outlet, so battery life is important.
- Finally, we must also use a long-range communication protocol adapted to the campaign.

So we decided to develop a hive connected under LoraWan. Indeed, this technology meets all the constraints mentioned above. This hive will allow you to know in real time the temperature, the humidity with the sensors connected in the hives. The collected data will be sent to a server so that they can be read remotely by the beekeeper on a mobile application. The first feature of our system will therefore be a real-time monitoring of the hive. Then, this data will be extracted from a database and analyzed and processed through Artificial Intelligence in order to combat the problem of climate change.
We will first examine the different materials and components used for our system :

Computer SODAQ Explorer 10k ohm resistor DHT22

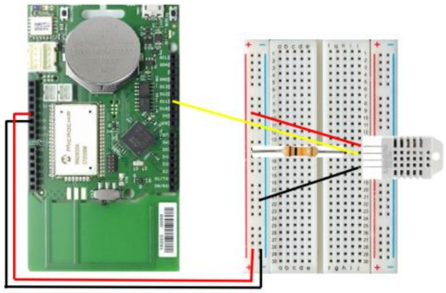





Feeds 3V flats rounds battery Breadboard

For the sensor, we wanted to capture the temperature and humidity. The DHT22 can do both with great accuracy. This sensor picks up the temperature a bit slowly but this is not a constraint in our case. Finally, this sensor is not expensive, so we chose to use it.
Here is the circuit :



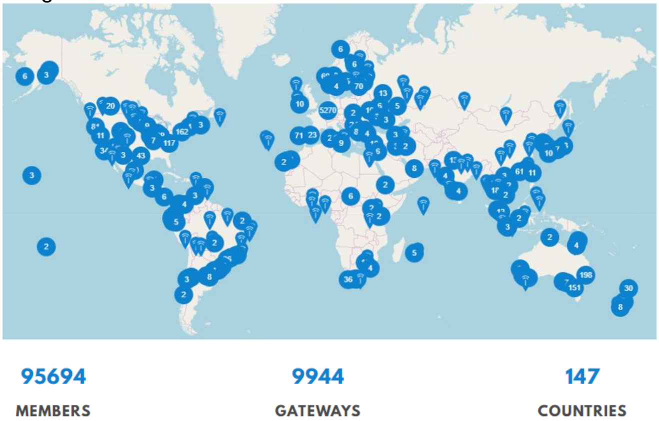
Lora Wan is a communication protocol like WIFI or Bluetooth. It was developed by Cycleo of Grenoble, France, and acquired by Semtech, a founding member of the LoRa Alliance. This communication protocol works with Low Radio Rate. This is why we have chosen to use this technology to preserve the health of bees.

There is another communication protocol called SigFox, very similar to LoraWan. We chose LoraWan because the flow and the amount of data sent is much more important than SigFox. Moreover, this technology allows to secure the connections between the connected objects and the server contrary to SigFox which has less possibilities to secure its connections. LoraWan is thus a technology in full expansion; it is for all these reasons that we preferred LoraWan to SigFox.

Finally, LoraWan is also a long range radio communication and it is a cheap technology because it is open source and there is a lot of free Cloud to recover all the data. We see that LoraWan corresponds to all our constraints. Now, we are going to see how it works:

To begin, the microcontroller reads the temperature of the DHT22. Then, the Lora module of the SODAQ explorer sends a signal to a LoraWan Gateway via Low Radio Rate. This signal can use 2 activation modes. In our case, we used the OTAA activation which works with an address called DevEUI and a password called AppKey. These 2 informations will be sent with the gateway with a secure IP connection to a LoraWan Cloud. The cloud will verify that the DevEUI and AppKey are correct and registered on their site. For our case, we chose to use The Thing Network as a Cloud because it is free. Now we can get the value on the Internet via a computer by going to the site or sending a mail for example. Then we have to use a cloud to retrieve the data.

TheThingNetwork is a website, it provides a set of open tools and a global and open network to build low cost IoT applications, with maximum security and ready to use. The Things Network is a proud contributing member of the LoRa alliance and that's why we used their services to send data through the LoraWan network. Lorwan is used because of its particularity of emitting electromagnetic waves at lower frequencies than known technologies.



Above, here is a map showing the extent of the network in the world (147 countries), they have 9944 gateways, which guarantees us to have a way to communicate our data. We need to set up a connection between the device and a gateway. We follow a tutorial to program the SODAQ Explorer card and connect it to The Things Network (TTN). After collecting the data on TTN, we used Cayenne to make a dashboard and allow a simpler implementation to the application.
Regarding the battery life, we decided to use the DeepSleep available under LoraWan to maximize its autonomy. We had some technical incidents due to timer libraries that were not up to date that we solved afterwards.
Finally, for retrieving data from The Things Network to a database, we have developed an API that writes our data to a JSON file that we can display on a Dashboard on a website. We also used MongoDB and Python with Pandas to analyze these data.

Github : https://github.com/WassimSERRADJ/Connected-Hive-LoraWan

5 Planned Resources vs Used Resources

For the beginning of the project, we had planned to use different resources. First, the involvement of all team members (5 in this case). No financial resources were required for this project. Concerning computer resources, we planned to use Teams, the Office suite, different coding tools (Visual Studio Code, VoTT, GitHub, Node.js...), in physics, we needed hardware such as a beehive, sensors and a SODAQ explorer microcontroller.
In the end, the vast majority of the resources we wanted to use were used. The only difference to note is for the use of Node.js. Node.js was necessary for the design of the application in React Native, so we changed tool (to use Android Studio, among others) because we finally coded the application in Native.

6 Expected end date vs actual end date

As far as deadlines are concerned, we managed to meet about half of the dates we had set ourselves. Especially for the deadlines concerning the "research" part of the project (where we informed ourselves about the beekeeping environment, potential targets, where we took care of the state of the art, etc...) we were able to meet the deadlines and deliver our various deliverables on time.

Concerning all the technical deliverables, the situation was a little more complicated. Indeed, regarding the connected hive, the deadlines were met for the hive itself (connection of sensors, functional hive among others), however, we have not really found a solution to date to process the information as we wanted. So the problem for this deliverable is more a problem of the chosen method than a real time problem, since the solutions we had initially chosen did not really correspond to what we were looking for as a result.

Finally, it is the application that presented the most difficulties from this point of view. So the first step was to find a viable solution for the model, which is not a particular concern. Then we had to feed the database to train our model to recognize the queens with the camera of a phone. This task took longer than expected because of the requirements that everyone had in their respective majors but also on other tasks of the project. Secondly, this task also took longer than expected due to the structure of the application, for two main reasons. The first one is that it was necessary to appropriate the notions in order to code on ReactNative first and then in Native then, which was more difficult than expected. Finally, the fact of changing language and method during the project created additional difficulties that caused a delay in the structure of the application. Thus, as far as the "final" modifications on the application (the menu and the rebrand) are concerned, we only managed to ensure the rebrand, implementing a menu involving big structural modifications that we unfortunately could not handle in the context of the end of the second semester.

7 Methodological review

First of all, the communication between our project team and the other parties (ESILV, Mrs. Antinéa Penciolelli, who is our PRM, and the company manager Ismaïl Ben Hariz), the communication was assured by the group's referent, Virgile Procureur (appointed at the beginning of the school year when the subject of the Pi² was chosen).

The majority of the team members had already had the opportunity to work together before, so we already had affinities between us to work as a group.

This allowed us to have dynamism and flexibility in the team, and it is mainly for this reason that there was no team leader or supervisor, as we already knew how to work together effectively.

Since the project was mainly divided into 2 main parts (design and development of the connected hive and the mobile real-time recognition application), we initially split into two teams that changed and evolved over the course of the project.

Thus, our team adopted an agile method with rush periods when there were renderings to be produced or when a task was nearing completion.

In terms of organisation and internal communication, we used classic tools for collaboration: Teams for common hosting of documents and files, facebook messenger for internal communication, github for version control and collaborative application development, and google colab to use the GPU provided by Google (necessary for building the TensorFlow model) and to create shared notebooks.

8 Risk Management

- Negligible risks in grey
- Manageable risks in yellow
- Dangerous risks to be transferred to orange
- Risks to be avoided completely in red

9 "Post-Project" tasks

Concerning the post-project tasks :

For the application, we initially had to finish the menu before handing it over, but the difficulties involved and the lack of time unfortunately forced us to hand over the task to a future team. For the rest, the application being functional, everything we wanted to do was done on time.

For the hive, it would now be necessary to establish data pipelines to be able to exploit the data, and to establish analysis algorithms. We have the data on TheThingsNetwork and we can transfer it at any given time to a third-party database, and now we would need to make the data collection more dynamic. That is, create some sort of event that is triggered when a new set of data is available on TheThingsNetwork that would transfer that new set of data to our database.

10 Satisfaction of customer and users

Mr.Ben Hariz (client) followed the project throughout and was mostly satisfied with the work we provided, and especially with the research we carried out into the technologies involved, their implementation and the results obtained.

As far as the connected hive is concerned, we interviewed several amateur and professional beekeepers and presidents of beekeepers' unions. The main problems and needs were identified such as temperature changes with the climatic urgency as well as the costs of a connected hive and access to free online courses.

However, even if the mobile application as well as the connected hive work rather well during the first tests (various videos of queens online), we cannot really consider that it already meets the specifications. Indeed, given that temperatures are still low and the current crisis (we write this document at the end of March 2020) linked to the Covid-19, we have not yet had the opportunity to test the application in real conditions (looking for the queen in a hive), which will be the real test for our project.

11 Project Completion Recommendations

Concerning good practice in project management :

We believe it is imperative to have a good organization from the beginning in order not to get lost in the deliverables at critical moments. Thus using Teams and other task management tools is fundamental to optimize its organization. Concerning the work with several people on the technical part, we used two main tools: Google Collab and GitHub. Both tools are fundamental to be able to progress on the technical tasks. The learning curve may seem long, but using them will save precious time in the project work.

For the actions to be carried out, we can list several of them:

- On the hive:

Find a method to be able to process information dynamically. Currently our model allows "only" to return the information in raw form, but we would like this information to be processed in order to guide the users in the choices to be made for the maintenance of their hives.

On the application :

To be able to make a menu, it was one of our secondary objectives that we did not succeed in fulfilling unfortunately. The purpose of this menu will be to serve as a basis for the rest of the functions of the application, so it will be very important to do it before tackling any other function. If the menu is realized, the main functions that we thought to implement are: the detection of the varroa mite (on the same model as the detection of the queen), the access to the data of the connected hive (if the work on the connected hive has progressed correctly) and finally a part advices/tutorials for the amateur beekeepers which aims to help them in the maintenance of their hives.

12 Annexes to the closure report (french)

Please list the annexes to the report

Annexe 1 – Questionnaire retour apiculteurs (Novembre 2019)

Annexe 2 - Spécificités application (Octobre 2019)

Annexe 3 – Recherches abeilles (Octobre 2019)

Annexe 4 – Recherches Machine Learning (Octobre 2019)

Annexe 5 – Recherches apiculteurs (Octobre 2019)

Annexe 6 – Etat de l'art (Octobre 2019)

Annexe 7 – Lien GitHub pour récupérer les fichiers de l'application :

https://github.com/Ltothe3/BeeIA-master

Annexe 8 – Fichier APK de l'application (Mars 2020)

Annexe 9 – Connected Hive

https://github.com/WassimSERRADJ/Connected-Hive-LoraWan

Annexe 10 – Final application

https://github.com/finaldzn/BeeIA/releases?fbclid=IwAR19qIkWHGi3E3xsXqlbx993ib4LZbRb_dL2TPubBNYqGF-cg8Shw5nmZI8