AGRITECH AND PRECISION FARMING

Capstone Design

April 2020

Mohamed Abdul Kahar
Supervised by: Dr. Kevin Smith
AGRITECH AND PRECISION FARMING

Capstone Report

Student Statement:
I have used the available knowledge about the company we are working with in order to have a decent final report, showcasing a form of project that would prove to be beneficial and interesting for all parties involved in it. In addition to that, I have made sure that the design and implementation of this project will not violate people’s privacy, safety, security, or go against societal and environmental aspects.

_____________________________________________________
Capstone Student:      Mohamed Abdul Kahar

_____________________________________________________
Approved by the Supervisor

_____________________________________________________
SUPERVISOR:            Dr. Kevin Smith
ACKNOWLEDGEMENTS

This work could not have been possible if it weren’t for its great contributors. I would like to thank Dr. Smith for providing helpful and insightful guidance on how to advance in the project efficiently. I would like to thank Miss Mac Arthur for helping my team and I in connecting us with different companies to know which one was best for us, as well as giving us constructive feedback on our work. I would like to thank company officials for providing us with as much information as they could to help us build a solution. I would like to thank my teammates for being professional and serious this semester, which helped all of us a lot, especially during these dire circumstances. Finally, I would like to thank Dr. Alj for always clarifying points we asked about, reminding us of deadlines, and giving us updates.
# CONTENTS

List of Figures 5  
Abstract 6  
1 Introduction 7  
  1.1 Overview 7  
  1.2 Methodology 8  
2 STEEPLE Analysis 9  
  2.1 Societal 9  
  2.2 Technological 9  
  2.3 Economical 9  
  2.4 Environmental 9  
  2.5 Political 10  
  2.6 Legal 10  
  2.7 Ethical 10  
3 Requirement Specification 11  
  3.1 Functional Requirements 11  
  3.2 Non-Functional Requirements 12  
4 Feasibility Study 13  
5 Design 14  
6 Implementation Details of the Solution 25  
7 Technological Enablers 29  
8 Conclusion 30  
9 Future Perspective 31  
10 References 32
LIST OF FIGURES

Fig 1. The first page of the app.
Fig 2. Water subpages.
Fig 3. Average rainfall per month in 2019.
Fig 4. Real-time weather predictions.
Fig 5. Weather report of Berkane for three days [3].
Fig 6. “Crops” subpages.
Fig 7. Clementine data in 2019.
Fig. 8 Future water needs per day.
Fig 9. Foot Crow’s notation of the database.
Fig 10. Chunk of database code.
Fig 11. Table population with data.
Fig 12. Tools used.
Fig 13. Main page.
Fig 14. Water page and subpages.
Fig 15. Connecting to database
ABSTRACT

This capstone project is part of the bigger picture, which is the newly created concept of joint capstone, where students from different schools work together to achieve a solution for a real life company. The computer science aspect of the project deals with the creation of a piece of software that acts as a data receiver and processor. The company we worked with is owned by Mr. Ali Belhaj, who owns multiple citrus farms in the country. His farms have sensors that are used to accurately water each plant individually, rather than considering the whole ground to be watered as one element. This helps in better use of resources, which is environmentally responsible. Our software would have been an app that demonstrates data about each farm, but since we didn’t receive information about each farm, the app displays information about crops and water for now. The sensors would send the data recorded to the app in order for the manager to see if resources are indeed being used efficiently. Therefore, the app needs to have a friendly GUI that is easily navigable and usable. For demonstration purposes, the app has data already in it.

Keywords: Farm, database, GUI, app, sensors, resources, water, environment, irrigation systems…
1 INTRODUCTION

1.1 Overview

The app is a dashboard containing different buttons. Those buttons would normally be a group of data about different farm elements classified based on our customer’s desire, which in this case, is Mr. Belhaj. It will give the manager an overall view of how resources are used. Are things being done efficiently? Are they not? The app would receive data from sensors to process and turn into information and results that are easy to be read by the manager. However, as mentioned earlier, the app doesn’t display information about farms. Instead, it currently displays information about crops, meaning how much water they will need in the future and at what time, and information about water, meaning how much it will rain on what day.

In today’s age, environmental problems are on the rise. There are serious attempts made by different parties to lower long-term damage to the environment. Problems causing such damage include, but are not limited to, high carbon dioxide emissions, water waste, and little to no recycling. Mr. Belhaj intends on upgrading his arsenal by being technologically more advanced to help the environment and be more sustainable. This does not only help the environment, but it also helps his business since there will be less resources used to achieve the same, if not better, results. The objective of this app is to help achieve that goal. Although it will not be more than a working prototype, it will still serve as a foundation for a great cause.

In this report, I will be discussing the building blocks of the app. This will include the methodology, functional requirements, non-functional requirements, and conduct a STEEPLE analysis. In addition to this, there will be discussions about design, solution implementation, and diagrams. Since the outbreak of COVID-19 hindered a lot of progress, denied us from meeting our customers more often, and restricted our capabilities in gathering more information, this project was adapted to fit current circumstances.
1.2 Methodology

The app’s main objective is to create a viable solution for water problems. The idea is to compare different weather data and factors in order to achieve a plausible result. Since weather is never constant, especially with global warming being noticeably present, it was challenging to come up with a solution that would make real time computations. While average monthly data can give a general picture of how much resources were needed in different circumstances, it is not entirely useful.

Water is a problem in Morocco, and in order to make efficient use of it, there is a need for software that can show exactly how much water each plant needs. For that to happen, we need to know when to water each plant, and how much water to give it. This depends on different factors, such as how much rain we got on a specific day at a specific time (if it rained at all that day).

The dashboard app was not enough to make such computations. There was a need for a weather app that would help direct the dashboard app. Meaning, in order to take different weather elements into consideration, and make the necessary computations, the weather app had to be integrated with the dashboard app. We could have just made a database with different tables that would perform calculations based on data we enter. However, it is much better to have a real-life representation of the weather.

To achieve this, I had to build the dashboard app, which is a simple design that simply displays pre-existing data. With the help of the weather app, the dashboard app was able to display information on screen. In addition to that, the app was connected to a database with weather data identical to the weather app.
2 STEEPLE ANALYSIS

2.1 Societal

The societal part of this project is that it will lead to a new idea that could open up new opportunities to people. One opportunity is job opportunities, where if the project is to go big, many computer science majors interested in such an idea could work in this field. In addition to that, it will raise awareness about the social responsibility people should have regarding environmental problems, which is crucial to working seriously towards the overall benefit of people and the earth.

2.2 Technological

The technological side of the project is simply bringing something new to the table. The app itself is the technological aspect. The project will advance the company in the technological side of things, by having it possess something other farming companies might not have. This will give them the edge, which would in turn make them a more attractive enterprise. In addition to that, it will revolutionize data processing in the farming world in Morocco, as it is a new innovation that will benefit companies.

2.3 Economical

There are no economical implications regarding the computer science aspect of this project.

2.4 Environmental

The environment will greatly benefit from this project. As mentioned earlier, reducing the use of water to achieve the same, if not better results will yield a major impact. Agriculture uses a lot of water, which causes environmental problems. However, if we had a system that reduces such inconsistencies and inconveniences, we will end up with the benefits of agriculture minus the unnecessary waste. Therefore, the system will be environmentally responsible.
2.5 **Political**

There are no political implications regarding the computer science aspect of this project.

2.6 **Legal**

There are no legal implications regarding the computer science aspect of this project.

2.7 **Ethical**

The ethical aspect involves not using pirated software to develop the app. It also implies that any references used are mentioned, and that there is no claim of other people’s work. Finally, the app shall cause no harm for anyone or anything, rather it serves as a tool to help people.
3 REQUIREMENT SPECIFICATION

3.1 Functional Requirements

After meeting with company officials, I had to make significant changes to the functional requirements of the app. In the past, based on the information I had, I thought the user would have complete control of the data. Meaning they could enter different data about weather aspects, and how much water irrigation systems should give to each plant. However, that was not the case. Based on later discussions conducted with one of the company officials, I was informed that the app should only display information that was gathered in real-time.

In this case, the only functional requirement of the app is that the user shall be able to view information displayed by the app. The user will only be able to navigate the app. They will be able to see how much it rained on some day at a specific time, how much more water plants will need, and past data to have a general idea about how the weather could look like, although global warming could prove past data to be less helpful.
3.2 Non-Functional Requirements

Non-Functional requirements represent what the system should do. There are many aspects of non-functional requirements that need to be present for the software to be reliable and appealing. This can be achieved by coding the software and testing it many times to make sure that it doesn’t fail, crash, take too long, is complicated to use, jeopardizes the user’s privacy or security, doesn’t take too much space, and is compatible with different devices. The bullet points below show what those non-functional requirements are:

- **Performance:** The system should output results of different inputs in a reasonable time.
- **Safety:** The system should be safe to use.
- **Security:** The system should protect the data and information related to the farms.
- **Space:** The system should not exceed storage space, or use an unreasonable amount of space.
- **Portability:** The system should be portable, meaning it would be functional on any operating system.
- **Reliability:** The system should be reliable and not crash when using it.
- **Usability:** The system should be easy to use.
4 FEASIBILITY STUDY

The feasibility study changed in the course of the last month. Since we were forced to work from home, conducting the feasibility study proved to be challenging, since we couldn’t go to different farms, conduct more meetings with company officials, and couldn’t get quick feedback and response. In order to present a realistic outcome, we needed to adapt new strategies in order to complete the project.

In the first half of the semester, our feasibility study consisted of trying to see whether our solution could be created. This was possible if we found a company, interviewed them about their needs, set up meetings online about updates and further questions, met them in real life, and visited their establishments. Thankfully, all of the above was possible. Although we didn’t gather as much information as we wanted, it was still enough to work out some kind of solution.

After being placed in quarantine, we didn’t receive enough responses. Although we conducted an online interview to ask more questions as we progressed with the project, we still couldn’t visit the farms. This could have been useful, since we would have seen how their equipment worked, how the farms are divided, how they look like, and so on.

Regarding the computer science aspect of the project, I relied on what the customer wanted for the app to be like, but I also conducted research about weather patterns and crop needs. For the app to simulate real life results, it needed real life data, and what we were provided with by company officials was simply not enough. With this in mind, there wasn’t much information on the internet about weather pattern details in Berkane, which was the region we worked with. Therefore, the later sections have just enough information to present a solution in its early stages.
5 DESIGN

The design of the app is extremely basic. This was done because it was requested by our customer. They didn’t want anything too complex looking, since it is only in early stages of development, there was little time, and the purpose was to simply show that the solution could work. The app looks standard, with some tweaks to make it look more appealing. In order to better demonstrate how the app looks like, screenshots were taken to show that.

![Agricize 0.0.1](image)

**Fig 1.** The first page of the app.

The figure above is a screenshot of the first page of the app. When opening it, we are welcomed to its beta version. We can see two buttons there. The first button is called “Water”, 
and in there, we find all the information related to water. This includes data about rain, and data about the irrigation systems.

The second button is called “Crops”, and it basically has information about crops that are grown on the farm. Data includes how much water was needed, future projections, and so on. When clicking on the buttons, we can have a better idea about what we can see.

![Image of water subpages]

Fig 2. Water subpages.

The figure above shows the different subpages we get when clicking on the “Water” page. The first one is “Rainfall 2019”, which has data about rainfall in each month of 2019 on average. The data was gathered from a website that displayed past weather data of Berkane in
detail. This included information about temperature, wind speed, humidity, precipitation, and so on [1]. Although it isn’t useful to have an average estimate of how much water plants will need per month, it is still useful to look at for comparison purposes with future data.

The second subpage contains information about projected rainfall, meaning how much rain we will get in the next few days based on weather data. For this to be accurate, a weather predicting app needs to be integrated with the app.

Finally, the third subpage would contain information about irrigation systems. Since this app demonstrates data that was entered beforehand for simulation purposes, this subpage doesn’t contain anything. Future work on this part of the app includes connecting the app to actual sensors that would send data to the app. For this to happen, there needs to be a software that connects the two.
Fig 3. Average rainfall per month in 2019

The figure above shows data collected from the past year. The data above is an average of data collected during a month. So, for each month, there was data about rain, humidity, and temperature. The data was collected during the morning, afternoon, evening, and night. Then, the four readings were divided by four. This was the process until the end of the month, and the total result was averaged again [1]. This information isn’t useful for real-time actions to be taken when it comes to watering plants. However, they can be useful when looking at the averaged data of 2020 when it concludes.
The figure above shows data displayed based on information collected by a weather app. It shows different weather data, but most importantly for farms, how much it will rain that day and at what time of the day. This can greatly help engineers program irrigation systems accordingly [2].
Fig 5. Weather report of Berkane for three days [3].

The figure above shows a weather report of Berkane. The report was created by an app developed using Python, where upon clicking on a link, the user is redirected to another page with weather information about the region for which they live in. The location can be easily altered by changing the location in the URL with the name of the city [2].
The figure above shows “Crops” subpages. For the sake of time, only clementine data was used in the app. Although there could have been other crops of citrus such as lemons and oranges, there wasn’t enough information provided about those. The first button displays information about Clementines in 2019. The information includes how much water they would have needed on average per month based on data about rain collected in 2019, which was documented in the “Water” page.

The second subpage gives information about future projections of clementine. Basically, it tells employees how much water clementines will need based on the weather information collected by the weather app, and displayed in the “Rainfall Projections” subpage.
Fig 7. Clementine data in 2019.

The figure above shows past data about clementines. The table shows how much water was needed from irrigation systems on average per month after calculating how average monthly rainfall was recorded that year. Again, information like this one isn’t very useful when wanting to estimate real-time calculations, but it can be used to come up with some sort of pattern for future calculations. For this to make sense, I had to conduct a research on clementines since company officials didn’t tell us enough about them. With the research, I was able to know how much water those crops needed, which was 900-1200 mm of water per year, with an average of 3-4.5 mm per day [4].
The figure above shows how much water plants will need based on how much it rained/will rain during those days. In addition to that, the tables show the preferable time to water plants. In this case, it says noon. This is because during those three days, it never rained/will rain at noon. Rather, it only rained/will rain during the morning, evening, or night.

The design could use a lot of work to make it more user-friendly. It could also use some improvements, such as a built in weather application, rather than having two separate programs and try to link the two.

Finally, the app has panels on the top right-side of each page. The purpose of those is to provide users with an explanation of the functionality of each page, rather than having text on the pages themselves.
In order to have meaningful and accurate results, it is important to divide the scope into areas. This helps us when we talk about granularity. This means that when using resources in a certain area, we will be able to conduct accurate measurements. In this case, it makes sense to divide farms into hectares. The reason behind this is that farms vary in size, so farms cannot be taken as a constant variable. In addition, provinces can be thought of the same way. Therefore, having a consistently measurable element that can accurately yield meaningful results is paramount. Meanwhile, we can say that we indeed have a measurement for one hectare, which is a standard value called INnet. INnet is equal to one litre per second per hectare, which translates to a daily water requirement of 8.6 mm [5]. Since citrus needs about 4 mm of water per day, we can make calculations to translate into INnet, which is 47 percent of one INnet [5]. To get the requirement of a farm, we can compute the area of that farm, then multiply it by the INnet. However, this value remains a proxy value that is used to make predictions and estimates [5].

The INnet value can differ during different times of the day. The amount of water supplied to plants depends on other variables, such as climate, evaporation rate, and so on. With this in mind, it is important to note that the INnet value can be three times greater at hot, dry climates compared to humid climates. This goes back to the fact that humid climates have water droplets in the air, while in hot climates, water is simply evaporating constantly [5]. Again, this INnet value is simply a proxy value, so depending on how hot and dry it is, the inflation in the value could be less than three times greater. For sensors to profit from such measurements, they need to have the ability to process this data efficiently in order to make the best out of the information provided.

Relating the previous point to the measurements conducted on April 19th for example, and if it was sunny rather than partially cloudy with light showers, there could have been more water needed rather than the 0.2 mm measurement. In this case, the system needs to have a mechanism to adapt and learn to cope and act based on weather inconsistencies. We can say that average estimates, reliance on past data, and considering future projections is simply not enough.
In order to make the app aware of miscalculated predictions, machine learning must be implemented. By using AI, the app can be customized based on its clients’ needs. This further minimizes waste. There are already implemented solutions in this field with companies such as aWhere and FarmShots leading the movement to improve resource usage in agriculture. By taking weather factors such as humidity and precipitation, the AI can predict more accurately how much resources are to be used [6].
6 IMPLEMENTATION DETAILS OF THE SOLUTION

The solution was implemented using different tools that are mentioned in the “Technology Enablers” section. It is based on an HTML page that contains many subpages, has different data for display purposes, and has a database containing information for display purposes. The figures below show more details of the implementation.

![Database Diagram]

Fig 9. Foot Crow’s notation of the database.

The figure above shows the simple database diagram. Table “Weather” is the one side, and table “Rain” is the many side. This means that Rain has Weather’s primary key as a foreign key.

Next, we have “Crop”, which is the bridge table between “Rain” and “Irrigate”. Since the relationship between Rain and Irrigate is many-to-many, we needed a bridge table that contains both of the tables’ primary keys. The reason for that is because the amount of rain water collected by crops will determine how much water irrigation systems should provide.
Fig 10. Chunk of database code.

The figure above shows the creation of the four tables. By reverse engineering the code, we were able to create the diagram in Figure 9.

Fig 11. Table population with data.
The figure above shows the table "Crop" populated with data according to the weather app predictions.

```html
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1, minimum-scale=1">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.4.4/jquery.js"></script>
<script type="text/javascript"></script>
</head>

<div data-role="page" id="menu">
  
  <div data-role="header">
    <h1>Agricize 0.0.1</h1>
  </div><!-- /header -->

  <div role="main" class="ui-content">
    <p>Welcome to the beta version of this app.</p>
    <a href="#Water" class="ui-btn">Water</a>
    <a href="#Crops" class="ui-btn">Crops</a>
  </div><!-- /content -->

</div><!-- /page -->
</html>
```

Fig 12. Tools used.

The figure above shows the tools we incorporated in our design. In order for the app to fit within phone dimensions, we had to scale it appropriately.

Fig 13. Main page.

The figure above shows the main page build.

Fig 14. Water page and subpages.

The figure above shows the build of “Water” page and its subpages.
Fig 15. Connecting to database

```php
$db = mysqli_connect('localhost', 'abdulkaharm', 'oljefl8kk', 'abdulkaharm')
    or die('Error');
```

Fig 16. Displaying database data.

```php
$query = "SELECT * FROM Crop";
mysqli_query($db, $query) or die ('Error');
$result = mysqli_query($db, $query);
$row = mysqli_fetch_array($result);
while($row = mysqli_fetch_array($result)){
    echo $row['CropID'] . ' ' . $row['CropName'] . ' ' . $row['WaterNeeded'] . $row['RainID'] . ' ' . $row['CropType'] . ' ' . $row['CropID'];
}
mysqli_close($db);
```

The two figures above show connection of html to database, then displaying data of crops from the database using php.
7 TECHNOLOGICAL ENABLERS

Technological enablers are basically the tools used to develop a piece of software. These include, but aren’t limited to, APIs, IDEs, and so on. For the app I created, I used HTML, CSS, JavaScript, and MySQL. Basically, the app was developed using scripting languages that would be phone friendly. This means that the app would be written as a website, but when zipping the files and uploading them to a cloud, the zipped file would be converted to a QR code that could be scanned.

In my case, I used Adobe PhoneGap. This platform is a cloud where developers can upload their code for the software to turn into downloadable apps. The platform creates three versions: an IOS app, a Windows app, and an Android app. Once we scan the generated QR code (using a QR scanner from our phone) a link will appear, and we will have the choice of visiting the website or not. Once we click on the website, a download link is created. Once downloaded, we can install the app, and then we are free to use it.

In addition to that, I used JQuery to help me add functionalities to the app that would make it look more like a phone app. These included a background for the app so it doesn’t look boring, panels to describe each page rather than have text everywhere, and smooth transitions while navigating the app (In my case, I used a fade, meaning when the use clicks on a button or goes back to a previous page, the page fades into a selected page.)

The purpose of using MySQL was to create a database that could hold information for simulation purposes. The real-life application of the app would be the integration of a weather app in order to have real time updates about actions to take when faced with certain weather conditions. Although the database is useful for demonstration purposes, it wouldn’t be as useful for on-the-spot applications, since it would require developers to constantly update code. The weather software was developed using Python and is basically a webpage that displays live weather information once clicked [2].
8 CONCLUSION

The application is merely in its very first stages of development. It only serves as a working prototype in its beta version that serves for demonstration purposes only. Since there were various outside factors that slowed down the progress of the app, we all had to deal with what we could get our hands on. These factors included companies being unresponsive, companies responding late, companies changing their needs, and the COVID-19 complications. This led us to constantly change topics in the first two months.

The app can still be improved in the future. It is an innovation that is much needed, since environmental crises are present currently. Technological evolution is also present, and with the help of technology, we at least have a chance at combating environmental problems. The app we have starts with water saving. It helps farmers and engineers better manage their resources, reduces waste, and maximizes profit by reducing waste. This helps everyone involved, whether it’s the workers or earth.
9 FUTURE PERSPECTIVES

The app can be immensely improved in the future. One obvious addition that can be made is integrating a weather app with it. This would help the app capture real-time data and display the necessary information to the user. One of these information is how much more water needs to be given to plants after calculating how much rain was collected by plants (if it had rained at all.) Another piece of information would be the most suitable time of day to irrigate plants.

Additionally, the app could use some further design improvements to make it look more appealing and modern. This means that the app would be more user friendly. The more the app is user friendly, the more it would sell. This includes smoother navigation between pages, animation of data such as crop growth and rainfall.

Finally, the app could be discussed in detail with the actual customers to give feedback. This would greatly help us in identifying the strengths and weaknesses of the app, since we would be in direct communication with the people that requested us to build them the solution. Once we get their constructive feedback, we would be able to improve our solution. In addition to that, the app could present data on a map rather than only having textual representation of data.
10 REFERENCES


