Rain Check:

A Voice Automated Weather Forecast

Group 1: A. Weijs (2684523), L. Hutubessy (10902392), M. IJzer (2552611)

In fulfillment of the course ICT4D (X_405101)

Vrije Universiteit Amsterdam, March-May 2020

Index

C: Changelog

In the third iteration, the prototype was extended by implementing the Open Weather Map (OWM) API. The One Call package was chosen because it functions with geographical coordinates which means we can set them to the precise coordinates of the selected cities in the Zondoma and Yatenga regions. However, it seemed that weather data was not available for all five capitals, so we decided to select three cities that are dispersed enough to have significantly different weather forecasts: Gourcy, Bassi and Bougounam.

Moreover, a data model was created to retrieve the weather forecasts, with open data like temperature, rainfall and wind speed, and users produced data like location and timescale. A core element of the prototype is to make weather forecasts accessible, so qualitative values have to be attributed to the raw data. For the temperature data it was chosen to apply the Gurene temperature scale, for the rain forecast the Met office standard, and for wind the Beaufort scale. However, it was observed that there is not a lot of variation in temperature and wind in this region of the world, and that rain was also highly season dependent. The location of London was used as a reference because it has more variations in weather forecasts. Furthermore, it was decided that ideally, the margins of the amount of rainfall should be adapted according to the season so that in the rainy season, the qualitative values for rainfall are more precise than say “a lot of rain”.

The weather forecasts were formulated based on the API data and a written Python script. Text to speech was then used to create audio files of the forecasts, which formed 9 different combinations (3 locations x 3 timeframes) in two different languages. Text to speech was chosen over voice recordings, because of the flexibility of text-to-speech with regards to the amount of possible weather forecasts. It was decided that the script would run every 24 hours, so the new audio files will replace the old ones, and hence update the weather information. The conversion from the files that are produced from the script from .mp3 to .wav was not achieved automatically and was done manually as a temporary solution instead.

To further elaborate the prototype and create more value for the end users, we decided to add the possibility to provide weather information besides just receiving it. As rain gauges are used in the Zondoma and Yatenga regions to measure actual rainfall, a question was added to insert the amount of rain which would in turn serve the weather organisation in obtaining more precise and truthful weather information and improve their service. The question was added as a choice element as it wasn’t possible to work with multi-digit input, so the user can answer with for example “there is no rain” or “a little rain”.

Feedback was moreover given to keep the crop advice collected from expert knowledge despite the foreseen complications in implementing this. So it was decided to keep it as part of the prototype in the form of a dummy, as building the web interface to acquire this knowledge was out of scope.

0. Name

Rain Check

1. Summary of Key Idea

Rain Check is a voice-automated system that offers qualitative weather information to and retrieves rain data from local farmers in the Zondoma and Yatenga regions of Burkina Faso. It is difficult for farmers in such remote areas to gain access to accurate weather information, and when they do, it is often difficult to interpret or base decisions on the retrieved information. Particularly for inexperienced farmers, more specific, qualitative and localised information is useful for sowing, harvesting and protecting their crops in case of extreme weather conditions. In a climate with stark dry and rainy seasons, information on rainfall timing and amount is necessary. However, such precise information is often not available for remote areas. In the Zondoma and Yatenga regions, some farmers are equipped with a rain gauge to measure actual rainfall. Rain Check provides farmers with weather forecasts, and retrieves actual weather data in order to improve future weather services. The system supports multiple languages and has a very simple and intuitive voice based interface that translates quantitative weather data into comprehensible information.

2. Actors and goals

Stakeholder	**Operational goal**	**Responsibility in the envisaged system**
Farmer	Retrieves weather forecasts and provides amount of actual rainfall	Retrieves weather forecasts to support decision making in farming activities, and provides data on actual rainfall to improve future weather forecasts.
Weather Organisation	Provides weather forecasts	Provides weather forecasts for a specified region and timeframe. Improves accuracy of forecasts based on retrieved data.
Hosting Organisation	Provides qualitative translation of weather forecast and actual rainfall	Provides hosting service, provides up to weather forecasts and makes qualitative translation. Improves and adapts the application based on feedback.

3. Context and scope

The idea of Rain Check is based on the information gathered from farmers in Burkina Faso during the workshops of W4RA in 2014, as well as an interview with the project expert Francis Dittoh in April 2020. In the context of the Mr Météo project, these actors have led focus group discussions with farmers and innovators in the Zondoma and Yatenga regions. A clear need for weather information was identified, for farmers to more effectively plan sowing, cropping and harvesting activities: "Daily rainfall forecasts would be useful."

In West Africa, the agricultural sector is productive only during the rainy season. This season used to be from June to October, but is becoming less predictable due to climate change. Indigenous ways to figure out the weather, for instance leaves falling from trees or animal behaviour, have also become less reliable. Therefore, local weather forecasts regarding rain, wind and temperatures are important especially during the rainy season. These needs of farmers cannot be satisfied by the radio, since radio stations only broadcast global region weather information. Moreover, through the Mr Meteo project several farmers in this region own rain gauges to manually measure the amount of rain after every downfall. Currently, the propagation of this information is done through phone calls from peer to peer. This is very valuable information as for instance the cumulative amount of fallen rain is indicative of the start of the rainy season.

Another clear need has been identified to improve the accessibility of the weather information. Although French is the official language, there are dozens of languages spoken in Burkina Faso alone, of which 60 are indigenous [4]. Weather information is often simply not available in the relevant languages. And when it is, it is often not comprehensible. A value in degrees celsius to indicate temperature may not be meaningful to a farmer that has no experience with such scales of measurement. For temperature, the Gurene scale was chosen which is a scale that has been developed specifically for West African weather conditions [1]. In a study on the linguistics of this language spoken in Ghana, it was in fact determined that there is a wide range of Gurene terms and expressions to describe temperatures, as the term "hot" may be too broad for the degrees of hot temperatures common in this region of the world [1]. For wind and rain measurements, more popular scales like the Met office standard and the Beaufort scale were used respectively [2, 3].

The aim of Rain Check is to support the accessibility of local weather forecasts in remote areas, by providing meaningful weather forecasts and collecting actual weather data to improve future forecasts. Some contextual factors had to be taken into account in the design decisions of the system. Rural farmers in West Africa rarely have access to the internet, but do possess basic mobile phones or have at least access to one. As they carry these mobile phones when working in their fields, a voice interface seems the most suitable. Moreover, due to lack of connectivity, it is also difficult to localise potential users. Therefore the scope of the project has been reduced to a few select cities in the Zondoma and Yatenga regions.

4. Use case scenario script

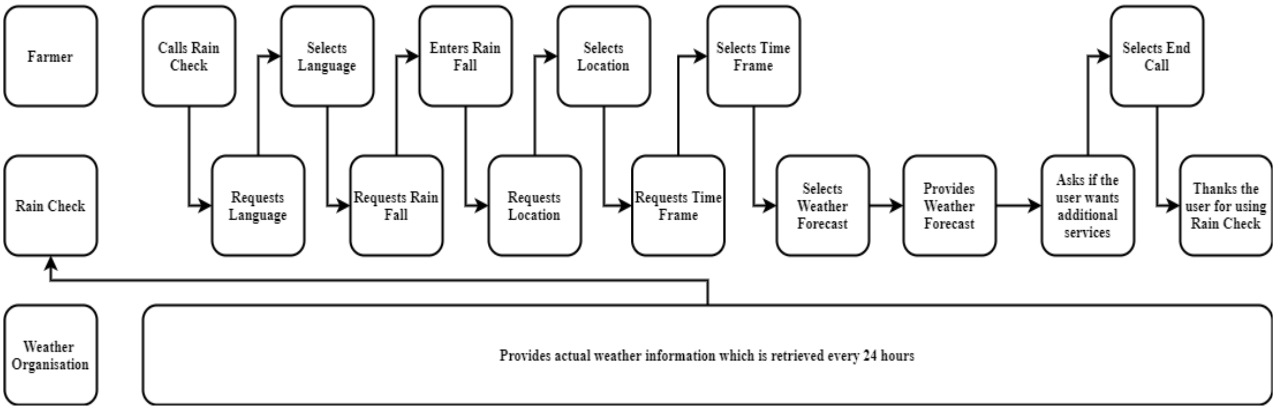
The system is designed so that the experience of the end-user, the farmer, feels natural and straightforward. Note that the selection of for example the prefered language is done by pressing numbers (DTMF) on the users mobile phone. The steps are as follows:

1. The farmer calls the number of the Rain Check service.
2. Rain Check asks to choose between proceeding the call in English or in French.
3. The farmer selects a language.
4. Rain check asks for the amount of rain collected by the rain gauge during the last rain.
5. The farmer selects the amount of rainfall.
6. Rain Check asks the farmer to choose their location: Bassi, Gourcy or Bougounam.
7. The farmer selects their location.
8. Rain Check offers to provide weather information for the coming 24 hours, the coming three days or the coming week.
9. The farmer selects his desired time frame.
10. Rain Check provides the weather forecast for the selected time frame and location.
11. Rain Check asks the user if they want to get another forecast, make use of the crop advice* service, the weather alert service*, or end the call.
12. Rain check thanks the user for using the service.

*Crop advice and weather alert services have not been implemented in the current project, but could be useful additions for further development of the application.

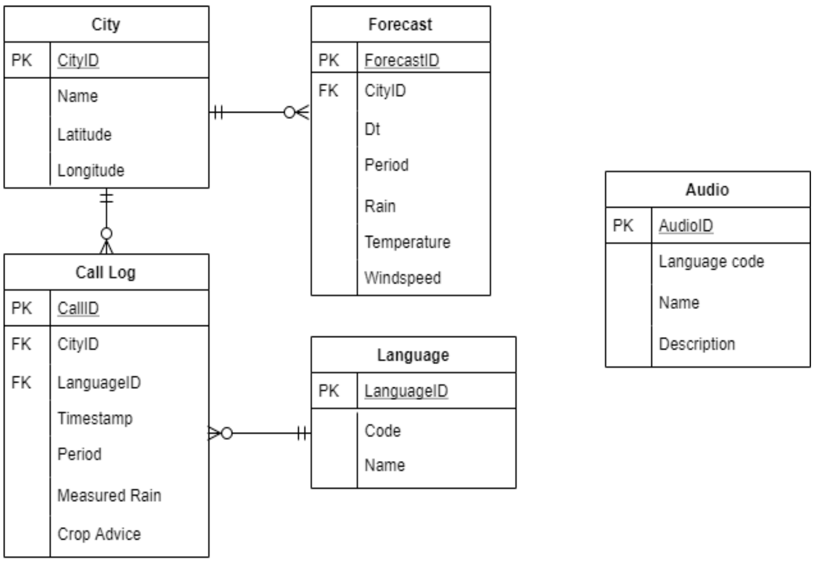
5. Interaction and communication

The figure below shows the activity diagram, with one row for each actor, of Rain Check.



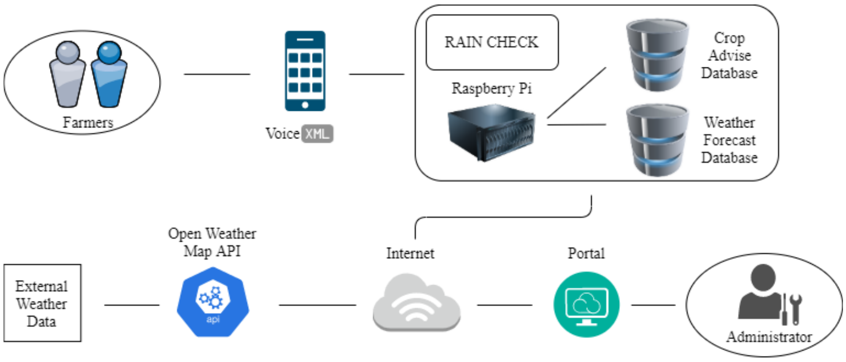
6. Information concepts

The figure below gives more information on the static data structures exchanged during the interactions. Our system supports storage of both user input and external open data. The user input is registered in a Call Log and consists of the location of the user (City, latitude, longitude), the Language, the rain gauge input (measured rain), and the desired timeframe (period). Call Log already anticipates on crop advice, an option to be added in future. The external open data from Open Weather Map are retrieved and translated into a qualitative weather forecast by a Python script forming the forecast with rain, temperature and wind speed for a certain period and location.



7. Technology infrastructure

The figure below shows the technology infrastructure needed to support Rain Check. The farmers have access to the system through their mobile phone and VXML-interface. Rain Check has been implemented on Raspberry Pi, as it's cheap and small hardware, suitable for our purpose. A central internet connection is needed to periodically load actual external weather data into the system by the OpenWeatherMap API. The system administrator can use a web portal to add expert data regarding crop advice to the system (interface not yet built in our current project scope).



8. Cost considerations

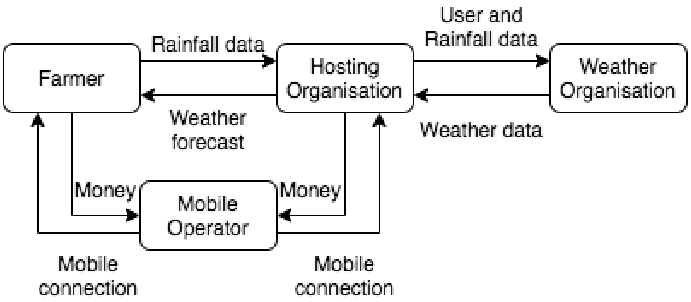
Currently Rain Check is a student project with hardware being funded by the university and no development and maintenance costs. If the project were to actually be implemented, the Total Cost of Ownership needs to be taken into consideration. There are foreseen costs related to the infrastructure, development and maintenance of the system. First, an initial investment is needed to evaluate the current prototype by users and experts. This will result in one or more developmental adaptations of the prototype. When the application is ready for practical use the costs of the operational service, maintenance, and continuous evaluation have to be covered. Therefore, it might be necessary to consider a small monetary contribution from the end-users. Based on the need-assessment of the farmers and the co-creation of the application, it's expected that the small monetary contribution won't be an issue. Overall, the table below shows a breakdown of the costs after the initial launch phase.

Costs Breakdown	**Actor Responsible**
Hardware (Raspberry Pi)	Rain Check
Mobile phone + Sim card	End-User
Telecom fees	End-User
Hosting service	Rain Check
Development and maintenance costs	Rain Check

9. Feasibility and sustainability

Although the Rain Check service is designed with farmers in the Zondoma and Yatenga regions in mind, the application can be useful for anybody interested in the weather forecasts in these areas. The system would therefore be considered successful if it is used by anybody in the cities of Bassi, Gourcy or Bougounam. This can be measured in terms of system usage, so not only how, when and where people call the Rain Check service, but also how many insert the data collected from their rain gauges. In fact, the success of the system also depends on the accuracy of the weather forecasts, so whether the predicted rainfall matches actual rainfall. By collecting this data, Rain Check provides the opportunity to improve weather forecast services offered by the weather organisation (e.g. by error analysis of the forecasts). This may in turn increase the amount of people using the service.

The latter increases the value of the system for all stakeholders. By providing weather forecasts in return for rain gauge measurements, Rain Check allows for a true value exchange, as can be seen in the figure below. By participating in this exchange, the end-users and weather organisation collaborate towards more local and precise weather broadcasts. This form of citizen data collection serves as a model for other communities to adopt rain gauges and engage in similar weather forecast improvements. A precondition for this would however be that rain gauge owners know how to correctly collect and measure rainfall data and input the corresponding values. Another important precondition is that there is a location with the necessary resources to host the system. This means a location with reliable power supply and internet connection and access to a mobile phone network. Finally, to avoid the risk that there is no capacity to maintain the system, efforts should be made to assure that there is a responsible actor, by involving for example a local NGO, farmers cooperation or government authority.



10. Key requirements

The table below shows our proposal for the key requirements of Rain Check as a starting point for further development. User evaluation is needed to determine the exact category classification of the requirements. The category 'Must have' is a summary of the most important requirements.

Must have	**Should have**	**Could have**	**Won't have**
+ Language support in English and French + Cellular access for end-user to call in + Internet access to OpenWeatherMap + Adequate and natural call flow + The use of Interactive Voice Response (IVR) + Translation of raw data to accessible information --- *Not yet implemented* + Language support for Dyula + Voice recordings instead of text-to- speech	+ Multi digit input for rainfall data + Language support for other local languages	+ Weather forecast based on end-user geolocalisation + Changing cutoff values for different seasons + Crop advice & web interface for system manager + SMS Weather alerts + Solution for parallel calls	+ Too many cities per application (see 11)

11. Prototype description

The prototype of Rain Check has been developed by using the KasaDaka software infrastructure. KasaDaka-VSDK is a python-project that uses Django to create a web-interface for voice-based applications. KasaDaka is a flexible platform. The building blocks allow the user to create a dialog-flow existing of prompts, choice elements, spoken user-input and more. Rain Check utilizes many of these blocks to achieve the interaction and communication diagram as shown in 5. Users provide Rain Check with actual rainfall measurements, and in return retrieve a weather forecast for their location and desired time frame. All functionalities of Rain check were implemented by using the standard building blocks of the KasaDaka-VSDK framework, so no custom adaptations of the framework were needed.

Several design decisions were made when creating the prototype. The first design decision that was made was to use the Open Weather Map API to obtain the weather information for the forecasts. The One Call package of OWM was chosen, because it functions with geographical coordinates which means we can set them to the precise coordinates of the selected cities in the Zondoma and Yatenga regions. The raw and quantitative data obtained from the API for each location was then translated to a qualitative weather forecast using a script. Scales such as the Gurene temperature scale, Met office standard and the Beaufort scale, were used for this translation. The qualitative result was then concatenated into a single forecast and stored as an audio file. The idea of the written script is that it reruns every 24 hours to update the weather forecasts to be actual and remain relevant.

So, it was important to ensure that the audio files that contain the forecasts could be replaced, while keeping the pointers the same and working so the flow as entered in the KasaDaka frontend does not have to be changed every time. Therefore the decision was made to create a branching structure in the call-flow, where the choices made with respect to the location and timeframe all lead to different paths for the specific forecast. One consequence of this decision is that it will be hard to extend the application with more cities, because adding one more city will increase the number of paths from 9 to 12. However, this might not necessarily be a problem since the length of the list is detrimental to the useability of the application. It would be better to host a different application with different locations for several regions instead of trying to capture all those regions into one application.

Another design decision was the use of text-to-speech instead of recorded voice labels. This decision was made because of the large number of unique forecasts that could potentially be formulated. With 5 categories for rain and temperature and 13 categories for wind speed it would result in 325 different forecasts. These then have to be multiplied by the number of cities and the number of timeframes. Therefore, using text-to-speech would be a lot easier in dynamically and flexibly loading and saving the forecasts. Although, the use of text-to-speech would be nearly impossible when extending the application with multiple local languages, that do not have a text-to-speech module. To make sure that the process is still automated, the script will have to be adapted to be able to concatenate pre-recorded audio files into one forecast.

The resources Rain Check uses are both user input, and external open data as described in 6. The user input consists of the rain gauge input, the location of the user, and the desired timeframe. The external open data is the Open Weather Map API, where the information for the weather forecast is retrieved, and Google Translate, for the text-to-speech and translation of the forecasts. These resources combine in a data model that can be used to both evaluate the application itself, and to provide the weather organisation with useful information about the actual rainfall. The application can be evaluated by looking into the number of calls received, which timeframes and locations were chosen, and the accuracy of the predictions. For example, when the longest time frame (7 days) is almost never chosen, the decision can be made to either remove that option or change the time frames options to be more suitable to the preferences of the users.

12. Pointer to the Application code

Heroku project: http://thawing-mesa-65251.herokuapp.com/vxml/start/2

Github code: https://github.com/Mick-IJ/KasaDaka-VSDK

Python script: https://github.com/Mick-IJ/KasaDaka-VSDK/blob/master/ICT4D_OWM_api.ipynb

13. Pointer to how to access the application

The application can be accessed and tested using http://ict4d.kasadaka.com. When using this website a Raspberry Pi will be used as hardware to process the application. The start-url of the application is the heroku project url http://thawing-mesa-65251.herokuapp.com/vxml/start/2. For documentation on installing this application on your own Raspberry Pi we refer you to https://www.raspberrypi.org/documentation/. A short recording/video of the application is uploaded to canvas.

14. Short usage scenario

When using the application one should be aware of what forecast they want to retrieve. This forecast can be obtained by entering which location (Bassi, Gourcy, or Bougounam) and which time frame (24 hours, 3 days, or 7 days) is desired. The first question that is asked relates to the rain gauge some of the users have. Rain Check requests for the user to enter their rain gauge information, however if you do not possess a rain gauge the option “I do not own a rain gauge” can be selected. If the user does have a rain gauge (s)he can select “no rain”, “small amount of rain”, “medium amount of rain” or “large amount of rain”. At the end of the forecast you can end the call or retrieve another forecast. The other two options, crop advice and weather alerts, have not been implemented and will thus redirect you to the end-call option.

15. Feedback questions

We were unable to ask questions to the stakeholders. The table below shows the questions that were asked during the interview with Francis Dittoh at the start of the project.

Current situation	• Which applications are currently in use? • Are there any problems/unforeseen issues that became apparent when using the current applications? • Which future applications should have priority, based on the most important goals and needs of farmers and the problems in the current situation?
Actors	• Who are the main actors in the ecosystem? • Where do most of the crops and foods are sold? Who else benefits/loses in an instance of a poor season?
Costs	• Can we suggest a subscription or other (symbolic) monetary plan for farmers in order to use the application; or does it have to be made freely available?
Performance	• How many users should use the application (total resp. simultaneously)
Technology	• Firstly, we assume that everyone has access to mobile phones, correct? • Since the VXML application is server-based, can we assume that internet connection is present at the server?

16. Discussion of Scope and Fidelity

Currently the prototype is only available in English and French, with English being used for the convenience of development and French being the primary spoken language in Burkina Faso. It is however a multilingual country with 70 spoken languages of which 66 are indigenous [4]. Ideally, the application would be available in all local languages spoken in the rural areas where people currently don’t have access to accurate weather forecasts. For the next iteration and for the purpose of testing the prototype with end-users, it would be useful to make it available in Dyula, which is the most predominant Mande language spoken in the West of Burkina Faso [4]. This would require the voice labels to be recorded individually instead of using text to speech techniques.

The prototype is also only available for three preselected cities in the Zondoma and Yatenga regions. This was chosen because of the difficulty of implementing localisation methods. Alternatives would be to deploy devices in every community as done in the RadioNet project, or the triangulation of cell towers. It would in fact be ideal if the user’s localisation could be detected using geolocalisation techniques so that the system can be used anywhere. However, the current KasaDaka framework does not allow for such a component. Also the timeframes for which the users can receive weather forecasts are restricted to one day, three days, or seven days. This was chosen to limit the amount of possible combinations of strings of weather data needed to generate a weather forecast.

A key point to improve would be the way rainfall measurements are gathered. Currently, the question is added as a choice element as it wasn’t possible to work with multi-digit input, so the user can for example answer with “there is no rain” or “a little rain”. For this functionality to be useful, the system would first ask for which time period the measurement was collected so that the weather organisations can match the collected measurements to the corresponding rainfall. The user would moreover be able to insert the amount of rainfall in millimeters, so that weather organisations can aggregate the data and determine the start of the rainy season. The rainfall measurements could for instance be sent to the Open Weather Map organisation for further error analysis.The weather forecast provided in exchange could moreover be more precise by adapting the margins of quality attributes to the season. In this way, the verbal description of predicted rainfall would be more precise during the rainy season when the amount of rain is highly valuable information. It would however be less precise during the dry season where there is rarely rain anyway and temperature plays a bigger role. By facilitating the exchange of accurate and localised data between farmers and weather organisations, Rain Check would become the central information point for weather related information.

17. User evaluation

The current prototype was not evaluated by the intended end-users (i.e. the farmers in sub-saharan Africa). However, some acquaintances of the authors tested the application. The testers got a small explanation of the context. They also got information about the rain gauge and received an assignment such as “retrieve the weather forecast for Bassi for the next 24 hours”. Afterwards, the testers were asked what they thought of the application and whether they had any issues or feedback.

The main conclusion from the user evaluation was that all testers were able to accomplish the requested task with ease. Some comments they gave were that Rain Check had a good pace and short sentences. One of the testers thought that it would be better to use one single voice for everything instead of two different voices (male for the standard KasaDaka voice labels, and female for the Rain Check specific voice labels). But, another tester said that it was nice to have two different voices, so it is clear what the options are. Some other feedback was that the english voice is a little unclear and quite fast, while the french voice is easier to understand. However, most of the French voice labels were literally translated using Google Translate, and because of that the vocabulary in French was sometimes not accurate.

18. Conclusion

The goal of this project was to develop a working prototype of an automated voice system that provides weather forecasts. This was decided based on an demonstrated need of farmers in rural Africa to have predictions of weather conditions in order to determine changes in seasons and extreme weather conditions to better plan their sowing, cropping and harvesting. Decisions on how to best respond to this need with the resources available have been made by meticulously following the ICT4D 3.0 framework. This is a collaborative, adaptive and iterative process that consists of five phases: context analysis, needs assessment, use case & requirements analysis, sustainability assessment and engineering deployment evaluation. Over the scope of 8 weeks, three iterations have been completed to come up with the current prototype.

In each iteration, new design decisions were made and some old decisions had to be reconsidered. It was particularly when implementing the solution on KasaDaka and when implementing the API that the scope and fidelity of the prototype had to be adjusted. An example of a trajectory of design decisions is that of our localisation method. At first, the idea was to allow for the user to retrieve weather forecasts from anywhere in Ghana. The user would have to select their region and receive the corresponding weather forecast. This information was however too general, so in the second iteration we came up with the idea of using postal codes after having read that they have officially been introduced nationwide since 2017. However during a feedback session, Francis Dittoh informed us that in reality most people have no idea what their postal codes are, so we had to dismiss that option. Finally, we decided to consider the localisation techniques that we did have at our disposition and decided to reduce our scope to a few preselected locations.

This trajectory exemplifies our decision making process and the type of compensations that were made throughout. The ICT4D 3.0 framework was helpful in keeping us focussed on responding to actual needs and with actual resources in a sustainable manner. Overall, learning to implement solutions with limited resources helped us better understand the importance of contextual knowledge. Because we did not have access to end users ourselves, we based our contextual knowledge on the insights of Francis Dittoh. In a future iteration, interviews and testing with actual end users would be highly valuable. For the purpose of the course fulfillment, KasaDaka was an efficient tool to develop a prototype within a restricted time period of 8 weeks. In future iterations, other software might allow for more flexibility.

Bibliography

[1] Koptjevskaja-Tamm, M. (2015) *The linguistics of temperature*. John Benjamins Publishing Company

[2] Beaufort scale. (2020) Retrieved 18 May 2020, from https://en.m.wikipedia.org/wiki/Beaufort_scale

[3] Wood, M. (1992). THE NATIONAL METEOROLOGICAL ARCHIVE. *Weather*, *47*(7), 247-250. doi: 10.1002/j.1477-8696.1992.tb07169.

[4] Languages of Burkina Faso. (2020). Retrieved 18 May 2020, from https://en.wikipedia.org/wiki/Languages_of_Burkina_Faso