

# Talking about the Petrichor Rain Service

## Creating a voice-based development kit for (rain) data service in rural Africa.

Master Thesis

Natasja Mooij, 2646719, n.mooij@student.vu.nl

July 18, 2019

### Abstract

Locally collected rainfall data is considered important information by local farmers in rural Africa, but the current way of sharing this data is complicated and insufficient. Because of this, the farmers don't always get the data they want in an accurate way and in time. This causes that they can't use their agricultural land in an optimized way. This Master Thesis outlines the research about the design and creation of a data collecting system to share local weather data among farmers in rural Africa, called the Petrichor Rain Service. For this service, a modular design is chosen, allowing it to be adjusted for other types of data and to make it more generic than just for rain data. The initial requirements for the service are that (i) the service should support multiple languages, including African languages, (ii) be easy to learn and use, also for people with low digital skills, (iii) that it should have a mobile voice-based interface, and that (iv) it can be used even in the absence of a steady internet infrastructure. From the evaluation of this service is concluded that the initial development of the service meets its initial requirements and that the service is ready to be implemented in the field.

## 1 Introduction

Weather information is important in rural Africa where more than 70% of the population depends on agriculture for their basic life needs [Yeboah, 2017]. Especially, for farmers who live from rain-fed agriculture, weather data is essential as rains are becoming less and less predictable over the years, causing a negative effect on their production of crops [Fox and Rockström, 2003]. In the rain season, keeping track of the amount of daily rainfall has become essential to manage crops properly. Recent field studies in rural areas of Burkina Faso and Ghana have shown how small scale farmers value the availability of timely, accurate and localized rainfall data for the optimization of their agricultural lands [Gyan, 2016].

Despite the need for weather information in rural regions of Africa, this kind of data is in 80% of the cases poor [Yeboah, 2017] or not available for various reasons. Weather forecasting services in Africa are not as accurate as in e.g. European countries, due to a lack of weather stations and interest of the African government [van de Giesen et al., 2014]. For farmers in remote regions, this lack of information is further magnified due to an absence of information channels such as the internet. Language, illiteracy and low purchasing power among rural farmers are other conditions

that obstruct access to online information or news services like the radio, newspapers, TV, libraries and online news sources [Uzuegbu, 2016].

Given the urgent need for weather information in various communities in Burkina Faso and Mali, innovative farmers have set up their own methods for data collection. In the absence of weather forecasts, they have focused on measuring the current amounts of fallen rain, often obtained by a manual rain gauge and manual data collection on paper. Knowledge about the cumulative rainfall over a certain period and in a certain region is used by the farmers to plan their cropping calendars. For example, one farmer in Burkina Faso shared that he would start to seed crops when 20 cm of rain had fallen in the month of June [Hans Akkermans, 2017]. To facilitate this data sharing among African farmers, the Petrichor Rain Service is designed and built within this master research project.

The Petrichor<sup>1</sup> Rain Service is a voice-based mobile platform, designed in the first place for collection and sharing of rain data. Because of its modular design and generic structure, the service can also be adapted for other types of data collection, for example, data on seeds, milk, non-timber forest products or other commodities that are often exchanged in rural Africa. The service can also be easily adapted to support other languages and can be seen as an add-on to the more generic Voice Service Development Kit(VSDK) [Baart, 2017].

In my research, I focus on a use case from rural Ghana. This has the advantage of using English as the primary language for the service. Moreover, the close collaboration of the research team W4RA with the University for Development Studies in Ghana and the presence of Francis Dittoh, one of the Ghanaian research supervisors of ICT4D projects at the VU Amsterdam makes Ghana a preferred country of this type of ICT4D research project.

## 1.1 Background

The farmers in Ghana don't have access to a wide range of technologies due to a lack of internet connectivity and low purchasing power [Bornman, 2016]. The only technology that is reasonably represented within the country is mobile phones. 80% of the adults in Ghana own a mobile phone in 2017, with the smartphones being the minority with only 35% [Silver and Johnson, 2018]. This is low compared to western Europe where on average 60% to 80% have a smartphone and 77% of Americans reported owning a smartphone in January 2018 [Poushter, 2017]. Even though 35% of the adults in Ghana has a smartphone, they don't always use them as smartphones. Because of the limited availability of internet in Africa (16% of the people have access [Frimpon and Adaku, 2017]) they can't always use the more extensive features a smartphone offers opposed to first generation mobile phones. Like visiting internet sites, for example, to look at the weather forecasts and sending data over the internet and using third-party applications. In e.g. European countries, a smartphone is a mobile phone fitted with a touchscreen and has access to the internet, while in Africa smartphones are purely mobile phones with a keypad and a connection to the internet. They look similar to the first generation mobile phones, but then with the added addition of having access to the internet. Below is a table (*'Table: 1 Smartphones sold in Ghana'*) with example phones that the farmers could own together with their estimated prices at the time of writing<sup>2</sup>.

---

<sup>1</sup>The Oxford English Dictionary describes 'petrichor' as 'A pleasant, distinctive smell frequently accompanying the first rain after a long period of warm, dry weather'.

<sup>2</sup>[www.zuricart.co.ke](http://www.zuricart.co.ke)

<i>Phones</i>	<i>Ghana pricing (Ghana Cedi)</i>	<i>European pricing (Euro)</i>
Tecno T472	116.17 GHS	€19,85
Nokia 105 (2017)	103.27 GHS	€17,64
Nokia 8110	376.92 GHS	€64,39
Tecno T484	136.83 GHS	€23,37
Tecno T349	74.87 GHS	€12,79
Tecno T465	113.59 GHS	€19,40 <sup>3</sup>

Table 1: Smartphones sold in Ghana

## 1.2 Related work

During research done on the currently available literature, one thing is discovered: the found research papers have in common that they only focus on getting data towards the farmer without the use of internet connectivity. Less attention has been given to the collection and sharing of locally collected data. For example in the area of meteo data in rural Africa, the paper '*The Trans-African Hydro-Meteorological Observatory (TAHMO)*' [van de Giesen et al., 2014] gives an explanation of the project where 20,000 hydro-meteorological stations are being installed and operated in sub-Saharan Africa. The objective is to have one station per 1000 km<sup>2</sup> to improve the quality of weather predictions, climate change assessments and development and management of natural resources.

Another paper, '*Using smart ICT to provide weather and water information to smallholders in Africa: The case of the Gash River Basin, Sudan*' [Amarnath et al., 2018] explains how remote sensing from satellite data can help to monitor crop growth and water consumption within agricultural fields. The satellite data can also predict when flood flows might occur. The study also operationalized information systems for farmers, which helped them to optimize farm profits and achieve more crop information for improving agricultural productivity and rural incomes in Africa.

Not only research done on meteorological services has a connection with this master thesis, but also voice-based systems play an important part. Like the paper '*Ney Yibeogo - Hello World: A Voice Service Development Platform to Bridge the Web's Digital Divide*' [Baart et al., 2018] which describes the technical implementation of the Kasadaka platform. A platform that together with the Voice Service Development Kit (VSDK) provides the residents in rural Africa without access to the internet to share knowledge and create content in a similar way to how the internet works. It also describes a case study to evaluate the VSDK on site in Mali to provide the villages with a system to submit spoken messages or important events that are offline stored in the system to be broadcast at a later time at the local radio station.

Another paper called '*A dialogue with linked data: Voice-based access to market data in the Sahel*' [De Boer et al., 2015] focuses on knowledge sharing services in the Sahel with the help of simple mobile phones, GSM architecture and voice technologies. It also describes a prototype voice application that provides voice-based access to collected linked data about the market.

The paper '*Voice-based Web access in rural Africa*' [Gyan et al., 2013] describes three systems, RadioMarché, Foroba Blon and Tabale which are designed and built to fit into the conditions in remote rural regions in Africa. RadioMarché is a system that can be used to distribute up-to-date market information via community radio in the area. With the system, voice messages can be

processed to create market messages which are then automatically generated. Foroba Blon helps the citizens of different communities in Mali by offering a way to send important news and stories to the local radio station to be broadcasted. This can be done with voice messages which are stored offline and can be announced when necessary. Lastly, Tabale is a system that can send out messages (in their own language) to the phones of members in case of an event happening. When they want, the members could listen to the message again or leave their own message.

Currently, Francis Ditto is doing a PhD research on how ICT can provide rural communities in developing countries with access to information which is relevant to their needs. The proposed system will create audio fragments with weather forecasts which are available via a phone call or an FM broadcast.

Because of the limited studies done on how farmers could send their own collected data, preferably in their own language [Bon et al., 2017], and without the need for internet, this leaves a gap to fill with this research.

### 1.3 The goal of this service

In this research, I present the Petrichor Rain Service. This is a voice-based service designed to retrieve local rain data from African farmers with the help of their mobile phones. The service is built according to requirements obtained during various workshops in Ghana, Mali and Burkina Faso where the farmers indicated what issues they encountered in their daily life and where ICT might benefit them. This service is built as an extension to the already existing Voice Service Development Kit (VSDK). This is a development kit which allows for web-based voice-service development with the use of building blocks [Baart, 2017]. There are a couple of building blocks already created, but a building block for entering data only using the keypad of the mobile phone is not yet developed.

The main goal of the Petrichor Rain Service is turning the manual action of writing down the collected amount of rain into an action where the farmers can use their mobile phone to submit the data. This will save them time, gives them more structure and they don't have to keep saving a piece of paper for later sharing of the data with other farmers.

This paper will include the development of a voice-based application which allows the users to use their keypad to insert data. The Petrichor Rain Service is only one of the many applications that can be built with this technique.

## 2 Problem Definition

In the current literature, there is little attention given to the rain information that the farmers can collect themselves. What was also noticed is that many of the currently created data-driven services rely on big data. Big data translates to large data volumes which are generated at an unprecedented and sometimes unpredicted rate. But this kind of data doesn't easily reach people in developing countries. They mainly make use of small data, data that results from the collected data of a human scale where the focus is on causation and understanding rather than prediction [Faraway and Augustin, 2018]. To create a meteorological, voice-based service to retrieve weather data from the farmers, the following research problem and research question(s) are composed.

## 2.1 Research Problem

The current way in which farmers in the Sahel pass on information about the fallen rain is through phone calls. At first, the amount of fallen rain is collected in the villages with the help of rain gauges. This data is aggregated in a central way at commune level by the village collector. After which the village collector sends the data by a phone call to an information manager at a central location who saves the collected data in an Excel file. This central location transmits the information to the local radio stations which broadcast the data at a fixed time. In some places in the Sahel, the collected information is sent every 10 days to the province level or to the national meteo service. See '*Figure 1: The current process of data collection*' for a visual presentation of this process.

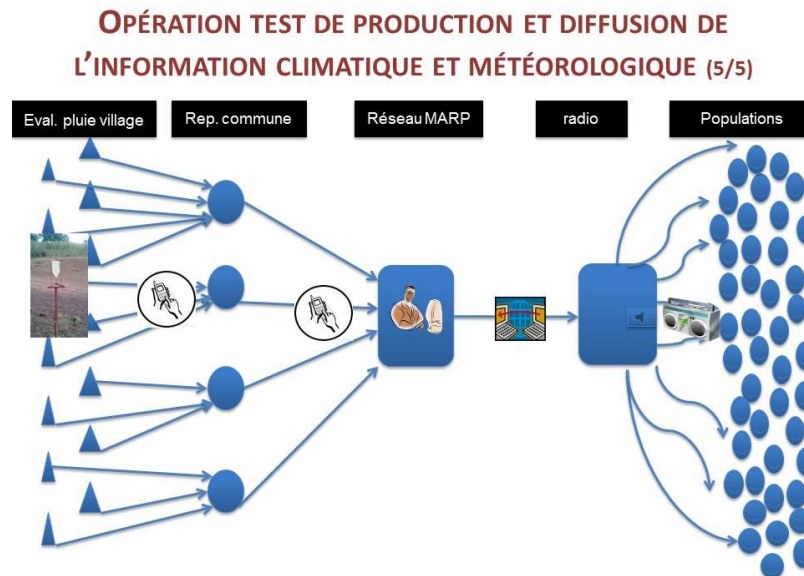


Figure 1: The current process of data collection [Hans Akkermans, 2017]

However, this method requires a lot of time from both the village collector and the information manager receiving the weather information. Not to mention, that the village collector and the information manager might not speak the same language and the high error rate that occurs when the data has to be transferred with speech through the phone [Bon et al., 2017].

The phone calls to the information manager cannot happen every single day, so the farmer or the village collector are almost forced to write down the rainfall every day on a piece of paper, store this paper and pass on the information at an agreed time. There are many things that could go wrong, like losing the piece of paper, not writing it down correctly, forgetting to check the rain gauges or missing/forgetting the call.

## 2.2 Research Question

The main question that is answered in this master thesis is:

*Can a data collecting system be designed that allows farmers to share local weather data?*

This question is a design question. This means that this master thesis will create the knowledge needed to support the development of a system or application, rather than only investigating and giving knowledge.

The main design question is answered with the help of the following sub-questions:

1. How can the local users receive requests to submit their data?
2. What is the process of sending the data to the system from the mobile phones of the local users?
3. What is an applicable method to store the retrieved data from the local users?
4. How can the system provide access to the collected data for interested parties?
5. How can the system be implemented locally in the Sahel?

A few conditions need to be taken into account when designing a system for the use in Rural Africa:

- Local Context
- Dynamics of local needs and requirements
- Local infrastructure and ecosystem

## 3 Methodology and Approach

The project is executed according to the ICT4D Field research methodology [Bon et al., 2016]. This methodology consists of 5 steps:

- **Understand the context in depth;** is about getting the same kind of view and bridge cultural distances with the environment the project is going to take place.
- **Elicit and assess needs;** finding out what the users want with the system; how they can work with it and what benefit it gives them.
- **Specify use cases and requirements;** developing use cases to translate the needs into understandable context for the development process.
- **Engineer, deploy and evaluate the system;** a working prototype is developed and tested in the field.
- **Assess sustainability;** takes a look at the business model and determines if the proposed solution is workable in the field. [Bon et al., 2016].

The first two steps have already been done by the W4RA workgroup during workshops in The Sahel [Bon et al., 2017]. This master thesis will use the outcome of these workshops to gather the wishes and needs regarding the system.

During a literature review of the documentation gathered from the workshops in The Sahel [Bon et al., 2017], requirements were gathered to which the Petrichor Rain Service needs to comply. This is all done in an iterative way, where the requirements will continue to evolve during the development of the system.

The focus of this research is on engineering the system in an iterative way. The main objective of this research is to create a system which can be used to collect data via mobile phones. In order to do so, a few different steps have been done to get to the end result of the creation of the Petrichor Rain Service.

For the sustainability, I looked at if the developed system would be usable and workable in the field and what would be needed to actually implement the service in rural Africa.

### 3.1 Programs, techniques and methods used

During the development of the Petrichor Rain Service, a couple of different programs, techniques and methods are used to create the prototype. The following lists contain all the different programs and techniques used:

- **The code**

- **Python**<sup>4</sup> The programming language Python is learnt to be able to create the system. This language is the core programming language used in the development of the Kasadaka and the Voice Service Development Kit.
- **Django**<sup>5</sup> is a web application framework which allows for faster and easier websites and applications development.
- **VoiceXML**<sup>6</sup> the Voice Extensible Markup Language for creating audio dialogs.
- **Fusioncharts**<sup>7</sup> has a collection of JavaScript charts which can be implemented in websites and use data stored in an online database.
- **Mapbox**<sup>8</sup> an open source mapping platform for custom designed maps.
- **Bootstrap**<sup>9</sup> an open source toolkit for developing with HTML, CSS, and JS.

- **Deployment**

---

<sup>4</sup>[www.python.org](http://www.python.org)

<sup>5</sup>[www.djangoproject.com/](http://www.djangoproject.com/)

<sup>6</sup>[www.w3.org/TR/voicexml20/](http://www.w3.org/TR/voicexml20/)

<sup>7</sup>[www.fusioncharts.com](http://www.fusioncharts.com)

<sup>8</sup>[www.mapbox.com](http://www.mapbox.com)

<sup>9</sup>[www.getbootstrap.com](http://www.getbootstrap.com)

- **Heroku**<sup>10</sup> is a cloud platform as a service (PaaS) used to build, deliver, monitor, and scale apps. For the Petrichor Rain Service, Heroku is mainly used to deploy the application.
- **GitHub**<sup>11</sup> is a repository hosting service where the (open source) code for the Petrichor Rain Service is stored.
- **pgAdmin**<sup>12</sup> a tool to monitor and access the PostgreSQL database of the Petrichor Rain Service.
- **Voxeo VoiceXML and CCXML developer site**<sup>13</sup> a remote server to test and create access to the Petrichor Rain Service.
- **Postman**<sup>14</sup> a program for interacting with HTTP APIs. This program is mainly used for the testing of the keypad input part and finding out where errors were located.

## 4 Building the service

This section elaborates on the various steps that are taken in order to create the Petrichor Rain Service. First, the scope of the system is determined in '*Subsection 4.1: Context and needs assessment*'. In '*Subsection 4.2: Requirements analysis*', the requirements collected in use cases and in the literature are written down and explained what they mean for the development of the Petrichor Service. Then '*Subsection 4.3: Developing, testing and deploying*' tells the actual developing of the service and some user tests that are executed. Lastly, '*Subsection 4.4: Sustainability assessment*' explains the feasibility and the implementation of such a service in rural Africa.

### 4.1 Context and needs assessment

The scope of this system is determined based on use cases done in Africa. Two important use cases are written down in the book '*Intervention or Collaboration*' by Anna Bon [Bon, 2019]. During the years 2010-2017, the W4RA team has visited farmers in Ghana, Mali and Burkina Faso to do workshops in which they discussed a variety of topics, among others the weather and climate. During the workshop in Ghana, the farmers in the focus group reported that information regarding the rainfall duration and daily weather forecasts would be useful for them. This kind of information will help the farmers in planning when and where to plant their crops to get the best and most result out of their planted crops and seeds.

During the workshop in Burkina Faso, the farmers made clear that they were interested in receiving daily accurate information on the (cumulative) amount of rainfall fallen in their own region. Especially during the rainy season, it is important to receive this data on a daily basis because during this period the majority of the farming takes place. Like the planting of the seeds, if needed cropping and the harvesting. During a workshop in 2014, while visiting the fields of farmers, it became

---

<sup>10</sup>[www.heroku.com/](http://www.heroku.com/)

<sup>11</sup><https://github.com/nmooij/petrichor-rain-system>

<sup>12</sup>[www.pgadmin.org/](http://www.pgadmin.org/)

<sup>13</sup>[www.evolution.voxeo.com/](http://www.evolution.voxeo.com/)

<sup>14</sup>[www.getpostman.com/](http://www.getpostman.com/)



clear that the farmers find it difficult to share their collected rain data (collected with the help of rain gauges) with other farmers because there are no clear communication channels available. The only way the farmers get rain data is during the rainy season. Twice a week via the regional radio stations. While their ideal situation would entail getting daily and weekly weather forecasts in the rainy season, get heavy-rain alerts and weekly accumulated rain data. They prefer to get all this data either via their mobile or radio.

The scope of this research is on data input, data storing and data output only. The focus lies on a group of farmers who already collect rain data manually with a rain gauge and process this in a non-digital way, like writing it down on a piece of paper.

#### 4.1.1 Actors and goals

This section provides an overview of the main stakeholders of the Petrichor rain service along with a brief description of their goals for the service. 'Table: 2 Actors and their responsibilities' below shows these actors and their corresponding goals:

<i>Actors</i>	<i>Responsibilities and goals</i>
Farmers	Would like to send and receive information regarding the rain-fall and weather.
Petrichor company	Providing the Petrichor Rain Service.
Radio stations	Receiving and broadcasting data.
Administrators	Keeping the data and the service itself up to date. Also providing access to the users of the service. A manual for the different actions the administrators can do can be found in 'Appendix C: A guide for administrator rights'.
Weather stations/services	Getting more and specific insight into the fallen rain in a particular areas.
NGO, National Meteo Service or Governmental Department	Providing financial support for the Petrichor rain service. Providing support to farmers by making weather prediction information available.

Table 2: Actors and their responsibilities

## 4.2 Requirements analysis

This subsection first explains the business idea in '*Subsection 4.2.1: Business idea*'. Then the use case of the Petrichor Rain Service is analyzed in '*Subsection 4.2.2: The use case*' with the different ways the service can interact with outsiders. '*Subsection 4.2.3: Infrastructure*' shows the more technical infrastructure of the Petrichor Rain Service. And lastly, the most important requirements for the service are explained in '*Subsection 4.2.4: Key requirements*'.

### 4.2.1 Business idea

The general idea is to build a service that can be used to collect data from its users via their mobile phone. As an example, a service is created to collect the amount of fallen rain from farmers in Ghana, but as mentioned before, the service can be altered to suit the need to collect other kinds of data as well.

### 4.2.2 The use case

The use case for the Petrichor Rain Service is to improve the process written down in '*Section 2: Problem Definition*'. Therefore, a new process needs to be created to collect the data. An image of this process can be found in '*Figure 2: The application process*'. The arrows simply mean the direction where the information is moving towards.

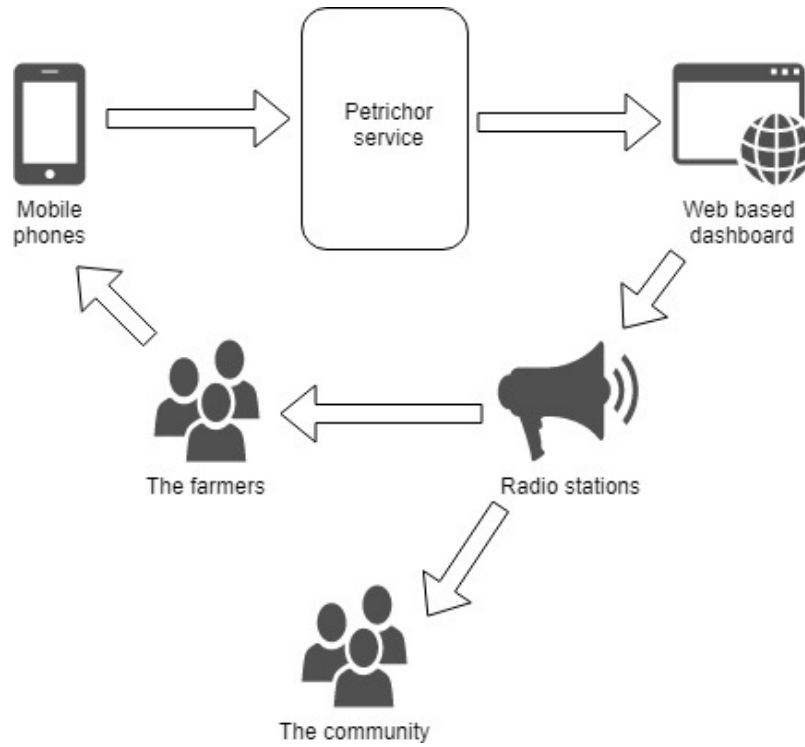


Figure 2: The application process

The farmers collect the data and send it, using their mobile phones, to the Petrichor Rain Service, then the collected data is stored in a database inside the Petrichor Service application. This data is later used to create a dashboard located on the web or locally on a computer. This dashboard can then be viewed and the information can be broadcasted by the local radio stations. In this way, the collected data will return to the community and the farmers who subjected the data in the first place.

#### 4.2.3 Infrastructure

To give a more in-depth view of the Petrichor Rain Service, an activity diagram is made to show the different steps of collecting, storing and broadcasting the collected rain data. The activity diagram can be viewed in '*Figure 3: The activity diagram*'.

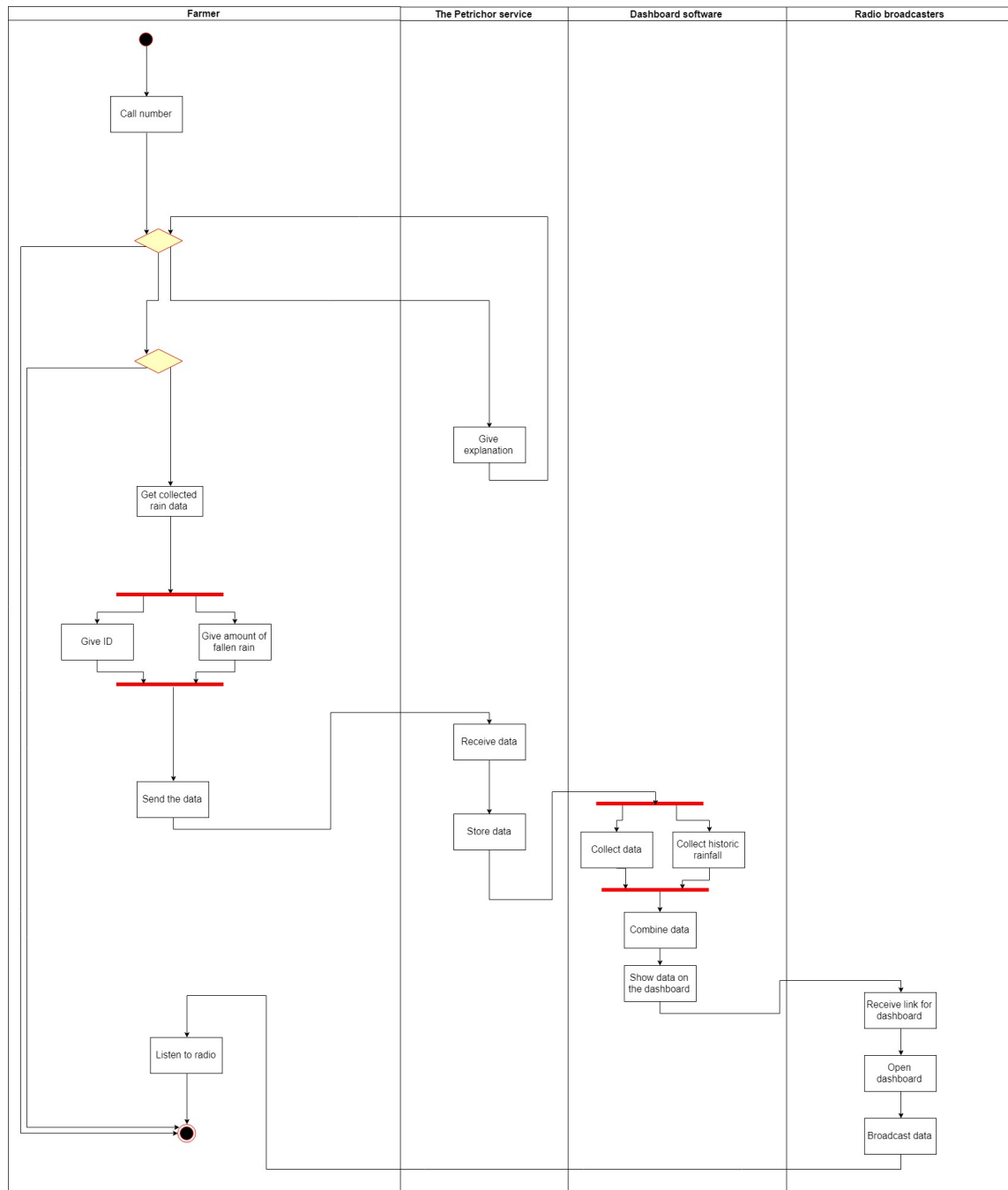


Figure 3: The activity diagram

At first, the farmer can interact with the Petrichor Service by calling the number themselves. The farmer is given the choice to either stop the call, listen to an explanation or fill in their data. When they choose to listen to the explanation, an explanation about the Petrichor Rain Service is played before the farmers are presented with the same choices as before. When the farmer chooses to end the call, they are asked if they are sure before the call is terminated. If they continue with the service, the next thing to do for the farmers is to gather the collected rain data from their rain gauges. When they have their data available, the service asks the farmers to fill in their Farmer ID. The farmer ID is a unique number for each farmer that helps them to be identified by the service. This number is linked to the coordinates of the place where the farmer is located. The following number the farmers have to fill in is the amount of fallen rain in millimetres. Only numbers are allowed to be used and the farmer can confirm by pressing the hashtag button. The data is then sent to the database to be stored. The dashboard collects this data, together with historic rainfall data already possessed and combines this data on the dashboard. After the radio station (or other interested parties) have received a link to the dashboard, it can be viewed and its data can be broadcasted back to the farmers.

#### 4.2.4 Key requirements

The core of the Petrichor Rain Service is collecting data from the users, in this case, the farmers in Ghana. It is important that a couple of requirements are fulfilled to make sure that the service is usable for the intended users. The requirements come from the W4RA workshop held in 2017 [Hans Akkermans, 2017] and the W4RA use cases held between 2015-2017 [Bon et al., 2017]. The requirements are divided between functional and non-functional requirements. Below the list, a couple of important requirements are explained further because of their importance. All requirements in the list are Must Have requirements based on the MoSCoW list of requirements [Asghar et al., 2016].

- Functional requirements
  - Rain needs to be measured in one unit, either in centimetres or millimetres.
  - The service needs a steady number of farmers with rain gauges to fill in their data.
  - Limited reading because of literacy barriers
  - Use DTMF as the input method.
  - The service should be suitable for agile development.
  - It needs to be easy to extend the service with further development.
  - It needs to be suitable to be deployed on the Kasadaka.
  - The data needs to be stored in a protected way.
  - The way in which the data is visualized needs to be clear for the users.
- Non-functional requirements
  - The service needs to be easy to use without a complicated process.
  - No internet is needed to use the service.

- The service needs to be usable on all kinds of mobile phones.
- The costs for development need to be as low as possible.
- The service needs to support multiple languages.

**Literacy & Education rates among farmers** The education of farmers plays an important role in how they use their mobile phones. Research has shown that people with lower incomes and less education are less likely to use the internet or own a smartphone than people with higher education and income [Poushter, 2017]. Besides, the adult literacy rate in 2014 was 71% in Ghana [Unicef, 2016], meaning that around one-third of the population doesn't know how to read or write. With an average of only 53% of the population having enrolled in secondary education in the last 10 years [Unesco, 2018], it is safe to say that farmers in Ghana are more likely to be part of the lower income/less education group than the higher education & income group and therefore are less likely to own a smartphone. In Burkina Faso, the proposed numbers are a bit higher but are still low in comparison with e.g. European Countries.

**Existence of multiple local languages** Also, language plays an important role when developing a system for Ghana. In Ghana, the different languages spoken can be divided into ten major language groups, ranging from a couple of thousand to millions of mother-tongue speakers. Besides these West African languages, English is also an important language in Ghana. English has been used as an official language since it was colonized and is still the most used language in the government and education. Also French and Arabic are foreign languages used often in Ghana [Bodomo, 1996]. Because of the big variety of spoken languages, it is not feasible to build the system in all of the languages. Therefore English is chosen as the main language during the development of the Petrichor Rain Service.

**The use of the DTMF technique** The technology that the farmers use to fill in their input is DTMF. DTMF stands for Dual-tone multi-frequency signalling. It is a method in which a telephone switching system retrieves commands through the keypad of the telephone. When a button is pressed, a DTMF tone is produced which is then sent in the voice frequency range over the telephone lines [Cruz et al., 2016]. The tone is then passed through a DTMF decoder on the receiving end of the line and if necessary an action is done based on the given commands [Getu et al., 2015].

**Protecting the stored data** After the data has been collected from the farmers via the voice application, the collected data needs to be properly stored confirming the laws and regulations. Within the Netherlands and Europe, The General Data Protection Regulation (Regulation (EU) 2016/679) (GDPR) is the most important law regarding data protection. The GDPR contains rules that protect the processing and free movement of personal data. Although the Petrichor Rain Service doesn't contain much personal data, it does contain enough personal data so that the GDPR needs to be taken into account. According to the GDPR, personal data means any information that can be used to identify a natural person, like a name, an identification number or location data. The processing of data means any operation which is performed on the personal data like collecting, recording, storing or use [Parliament, 2016].

This means for the Petrichor Rain Service that a couple of rules need to be taken into account when it is decided that the service is going to be (partly) maintained or even extended at some point within The Netherlands.

Ghana also has its own law that protects the use and processing of data. It is called the Data Protection Act and was first introduced in 2012. Ghana also has a Data Protection Commission that monitors compliance with the provisions of this Act and investigates when someone has violated the Act. The Data Protection Act makes sure that personal data is processed in a lawfully and reasonably way and without infringing the privacy rights of the person. Personal data may only be collected and processed if the purpose is necessary and if the personal data is collected directly from the person itself. The Act also mentions that foreign data subjects are processed according to the data protection legislation from the country where the personal data originates [Commission, 2012].

According to the Data Protection Act, the act is applicable when either of the following rules applies:

1. The controller of the data and the data processing is established in Ghana.
2. Equipment or services of a data processor located in Ghana are used.
3. The information processed originates partly or wholly from Ghana.

All of the rules can be applied to the Petrichor Rain Service. The collected rain data will eventually be collected and processed in Ghana and the service is hosted locally. Therefore it is important that the Data Protection Act is kept in mind when the service is being created and maintained. Together with the European GDPR law, the Data Protection Act will protect the privacy and personal data of farmers who provide rain data to the Petrichor Rain Service.

**Visualize the stored data** The data entry module is just the first step in the process. In order to give something back to the farmers who use the system, something has to be done to present the retrieved data in a sufficient way. For example, to be broadcasted by the local radio stations. After all, studies have shown that people are more inclined to do something if they are given something in return. Rewards have a positive influence on overall motivation and performance, regardless of the person's internal motivation to do the task [Hendijani et al., 2016] [Cameron and Pierce, 1994].

There are a couple of different ways in which the farmers can get the data back:

- Sending the data back by phone
- Creating dashboards for radio stations
- Creating an excel file for radio stations
- Spreading it through social media

Each of the different ways is checked on a couple of factors:

- Internet access
- Illiteracy from the users
- User friendliness

After the assessment of the factors, the most applicable method is chosen to develop in '*Section 4.3: Developing, testing and deploying*'.

**A mobile voice-based interface** By offering a way to send the data back to the phones of the users, it means that the farmers can call a telephone number that will give them a voice recording with the amount of fallen rain in a certain period. This is a very simple solution that will work for the farmers when, for example, they are in the field and like to know how much rain has fallen. It also doesn't require them to do any complex acts, just calling a telephone number will suffice. This method doesn't need internet access to work and even farmers who are illiterate can use the service because the forecast is in their own language. However, this might be expensive for the farmers because they have to pay for the call charges themselves. They have to call to submit their data and call to receive the submitted data separately. The farmers are not able to alter the data they get to hear. For example, on one day the farmer likes to know how much rain has fallen in a week, but the other day he likes to know it for the past month. Or he likes to hear the rainfall for another region because that might give him important information about what rainfall is about to come. Another downside of this method is that there won't be a central place where someone can see all the submitted data from all the farmers if they are not administrators of the Petrichor Rain Service.

**A web-based interface** Another solution is to create Business Intelligence(BI) dashboards which can be used by the radio stations to broadcast the weather data to all the villagers and farmers. Business Intelligence is the processes, technologies and tools needed to turn data into information and information into knowledge [Loshin, 2012]. Although the processes, technologies, tools and dashboards are merely the means to BI and not the intelligence in essence. Intelligence is about how to use the available information, not about how to acquire the information [Nerkar, 2016]. Dashboards provide a visual means to organize and interact with data, allowing users to drill down into data sets and to overlay and interconnect disparate data. Dashboards act as cognitive tools that improve the user's control over a large repository of varied and quickly transitioning data without the need for specialist analytics skills [Kitchin et al., 2016]. Within a dashboard, there are many different ways in which the data can be displayed, like pie charts, line charts, bar charts and gauges [Conway, 2018]. The use of these different graphs and charts depends on which data is available and how this data can be shown in the most efficient way. A benefit of using dashboards to show the data is that there is little written text needed to show the data in an effective way, therefore also people who are illiterate are able to understand the dashboards. The dashboards are also very easy to use and all the collected data is stored and shown in a central place. A downside of using dashboards is that the farmers won't have access to them and that they are dependent on the radio stations to get their data.

**An Excel-based interface** An alternative output of the Petrichor Rain Service is as an Excel file because radio stations or meteo services prefer this type of output. This method means that just like the current way of working, the radio stations are presented with the data in a table format which they then can read aloud during the broadcast. The same benefits as using a dashboard can be applied here, only the data collected in the table can also be used for developing new speech functionalities like sending the data back by phone calls to the farmer.



**A social media-based interface** Some other output method is to spread the retrieved data through the social media channels that the farmers use. Some farmers have access to specific social media channels through their smartphone. These smartphones come preinstalled with the needed social media channels like WhatsApp and Facebook. The major issue with this solution is that the farmers need to have an account on any of the social media channels, something that is a bit harder when they only have a first generation mobile phone. This means that the majority of the farmers won't be able to receive the weather data if this solution is the only solution used. But it could be a viable solution for the farmers that do have access to the social media channels. An example of a use case could be that the administrator of the Petrichor Rain Service could create groups on WhatsApp or Facebook and post the weather data from the previous day or update the data as soon as a farmer sends in his data. However, this might be very time-consuming for the administrator. Another drawback of this solution is that this won't be a technical solution that can be provided by the Petrichor Rain Service. The farmers themselves or the administrator must show the initiative to make this solution into a success.

### 4.3 Developing, testing and deploying

This subsection is all about the development of the Petrichor Rain Service. From the input (*'Subsection 4.3.1: Input'*), through the storing of the data (*'Subsection 4.3.2: Storing'*) to the output (*'Subsection 4.3.3: Output'*) that the service will give to its users. Also the testing of the service is in *'Subsection 4.3.4: Testing'* taken into account.

#### 4.3.1 Input

Once the requirements are clear it is time to create the first (new) part of the service, namely the component that lets the users fill in their input.

The first thing I do is install and explore the Kasadaka Voice Services Development Kit (VSDK). The current version of the VSDK already has a couple of pre-made building blocks installed, like creating a service and collecting data via voice messages. Once the VSDK is installed on my laptop, I start to create a new building block to collect data via DTMF. To do so, some example code is studied just as the current code for retrieving data via voice messages because the code for collecting with voice messages or DTMF are rather similar in nature. With the help of code found on the internet, a new Django Template is created and with some more bug solving the first version of the input component worked. The data entry part is finished and tested a couple of times by uploading the code to a remote server and calling that server.

After the needed input functionality is created and tested, it is time to create the Petrichor Rain Service. This is done with the following elements, some are already preinstalled in the VSDK.

- **Voice Labels**
- **Choice Elements**
- **Message Presentation Elements**
- **Input Elements**

The Voice Labels contain all the prerecorded audio that is needed to add audio to the other elements in the rest of the service. The Choice Elements will give the users the possibility to choose between a couple of choices with the help of their keypad, while the Message Presentation Elements just present a straight forward Voice Label to the users. Lastly, the Input Elements are the new components of the system which will help the users in forwarding their data with their keypads.

In order to get a clear overview of how the different elements need to be linked to each other, a call flow diagram is created (see 'Figure 4: The call flow of the Petrchor Rain Service'). This call flow diagram shows how the call flow within the Petrchor Rain Service is constructed.

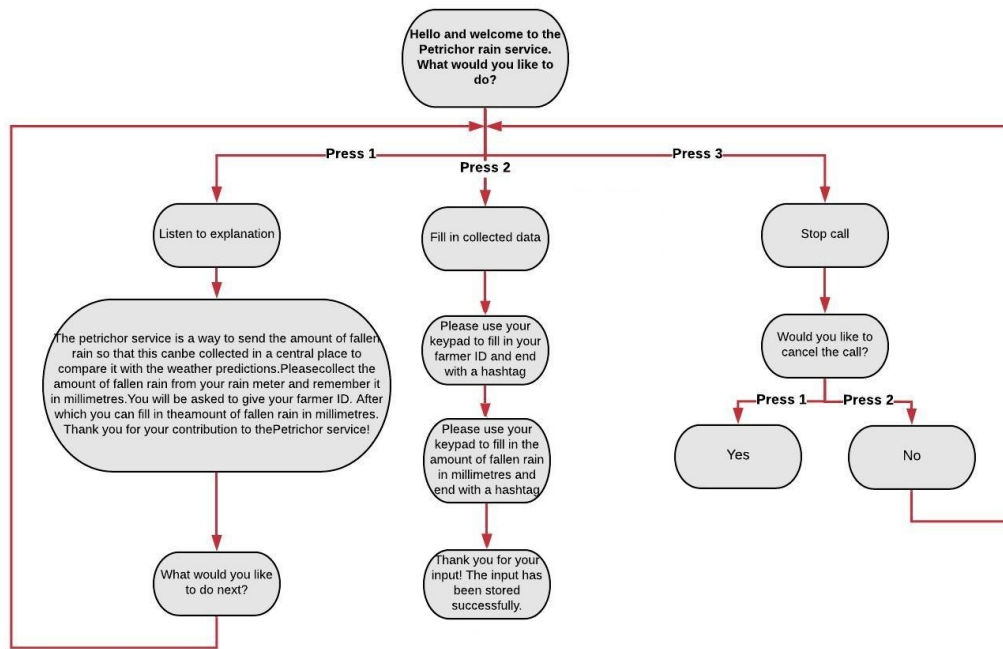


Figure 4: The call flow of the Petrchor Rain Service

The service begins with a welcome message which gives the user three options: listen to the explanation, fill in the collected data or stop the call. When the user wants to listen to the explanation, he presses one on his keypad after which the following explanation is played:

*The Petrchor Rain Service is a way to send the amount of fallen rain so that this can be collected in a central place to compare it with the weather predictions. Please collect the amount of fallen rain from your rain meter and remember it in millimetres. You will be asked to give your farmer ID. After which you can fill in the amount of fallen rain in millimetres. Thank you for your contribution to the Petrchor service!*

After the explanation, the user has the same choices as before. When the user then picks option two, fill in collected data, he is guided through a process in which first his farmer id is asked and then the amount of fallen rain. Each input needs to be followed by a hashtag (#) to let the service

know that the input is completed and that it can move on to the next part. Once all the data has been filled in, the service gives the user a thank you message before terminating the call. When the user is at the choice menu, he can also press three to immediately end the call. The service asks for a confirmation from the user that this is correct. If the user presses one, then the call is terminated, otherwise, the user is brought back to the choice menu.

### 4.3.2 Storing

After collecting the data, it is important that the data is stored in a permanent way in order to keep continue using the data. Because the Petrichor Rain Service is built with the VSDK framework [Baart, 2017], which is built with the Django framework, this limits the way in which the collected data can be stored. Not every database program (engine) works together with the Django framework, which is a web framework for rapid application development [Plekhanova, 2009]. Django supports four database engines [Holovaty and Kaplan-Moss, 2009]:

- PostgreSQL is an open source SQL database server that can be used on a wide range of platforms and is known for its performance and stability [Riggs et al., 2015].
- SQLite is an embedded SQL database engine that reads and writes the data directly onto disk files and is stored as a single disk file that can be used cross-platform [Owens and Allen, 2010].
- MySQL is a SQL client/server relational database management system and includes an SQL server, programs to access the server and a programming interface [DuBois, 2008].
- Oracle is a fully scalable relational database architecture and is often used by large enterprises to manage and process data over networks [Greenwald et al., 2013].

With the installation of the VSDK framework, the standard database is included in SQLite. But after some research, it is discovered that the SQLite database is not suitable for the Petrichor Rain Service. This because with SQLite, the data is stored in a single disk file which cannot store the collected data when the service is running on a server. This means that the rain data that the farmers will fill in cannot be stored or retrieved from the SQLite database. Therefore it is decided to use a PostgreSQL database instead. This database also works well with Heroku, the platform that is used to deploy the Petrichor Rain Service.

In '*Appendix A: The class diagram*', the class diagram for the Petrichor Rain Service can be found. A class diagram is a diagram that shows the structure of a system and shows the relationships between the different tables/instances that are saved in the system [Hans-Erik Eriksson, 2000]. Because the Petrichor Rain Service is built with the help of the VSDK framework, some tables of the framework are not used for the Petrichor Rain Service. But there are also tables that are especially created to be used by the Petrichor Rain Service. The most important table is the `service_development_input_data` table in which the data that the farmers contribute is stored. Other important tables are:

- `service_development_voiceservice`: the table in which the voice service itself is stored.
- `service_development_kasadakauser`: the table where the information of the farmer is stored.
- `service_development_callsession`: the table where the different calls from farmers are stored.

- `service_development_userinputcategory`: the table to distinguish the different input values.

### 4.3.3 Output

Looking at all the different possible solutions for visualizing the data in '*Subsection 4.2: Visualize the stored data*', one jumps out as the best option for now. That is a web-based interface. This would be the best solution because it provides a fast yet simple solution to show the weather data. At a later stage, the service could be extended to provide the weather data in other ways to the farmers and the rest of the community.

The definite dashboard has been built using the Fusioncharts extension for Django, in combination with Mapbox and Bootstrap for the layout. The Fusioncharts extension is a collection of JavaScript charts which can be used and inserted in webpages. By connecting Fusioncharts with the PostgreSQL database, the data inside the database can be used and displayed in the charts. Because Fusioncharts has over 95 chart types, it is important to figure out what the best ways are to display the data. When going through the data collected, a couple of important data combinations can be made:

- The locations of the farmers
- The amount of fallen rain per time period
- How much rain has been submitted per farmer
- Total fallen rain this month in history
- Total fallen rain this year in history

Each data combination is made into a chart. All charts make use of the same dataset, a view consisting out of the `voiceservice`, `kasadakauser`, `callsession`, `userinputcategory` and `userinput` tables. These tables are selected because they hold the most valuable information to be displayed on the dashboard.

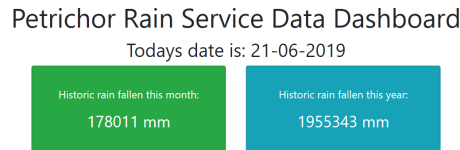


Figure 5: Historic rain data

**Historic rain** The first block in '*Figure 5: Historic rain data*' shows the average rain fallen in the past equal to the current month. While the second block shows the total rain fallen historically for the entire year up to this month. This data will help the user to compare the current data with the historic in order to determine if this is/will be a good or bad year for the crops.

The two blocks are created with the help of Bootstrap. Bootstrap is a toolkit to develop HTML pages with the help of a grid. The grid allows for easy placement of the different components and

helps to structure the page. Next to offering a grid, Bootstrap also has many standard components, such as buttons, navigation bars and other website elements that are easy to use. I used the Information and Success Card elements to display the two blocks in different colours. By using an if-statement, the numbers in the cards change according to the current month.

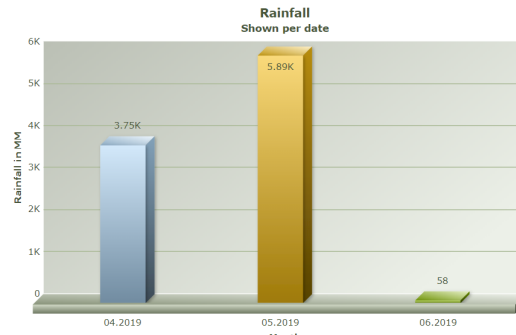


Figure 6: The bar chart

**Bar chart** The bar chart in '*Figure 6: The bar chart*' shows the amount of fallen rain per month, as submitted by the farmers. In a bar chart, the data is presented with rectangular bars with heights corresponding to the values they represent. Each time a farmer submits data, the corresponding bar chart of the current month will update to include the newly added data.

The bar chart is created with Fusioncharts. At first, a set is created containing only the rain data from the Petrchor Service. This data is put into the bar chart and rendered on the HTML page.

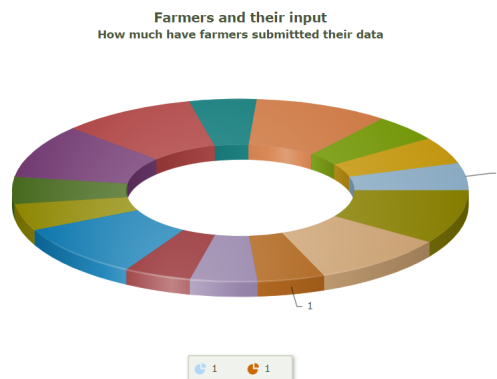


Figure 7: The donut chart

**Donut chart** Although a donut chart is considered bad practice, because it is difficult to compare the different sections [Franzblau and Chung, 2012], it is still decided to use this type of chart to present the amount of rain submitted per farmer. This because of the variety it shows compared with the other charts. The kind of data comparison shown in this donut chart is valuable because it will display the amount of rain submitted per farmer. If many farmers in a larger region are cooperating with the Petrichor Rain service, this donut chart will show which farmers submitted the majority of the fallen rain so that the average doesn't give a wrong image of other regions with less fallen rain.

The donut chart in 'Figure 7: The donut chart' is created with Fusioncharts. At first, a set is created containing only the rain data from the Petrichor Rain Service, in combination with the FarmerID's. This data is put into the donut chart and rendered on the HTML page.

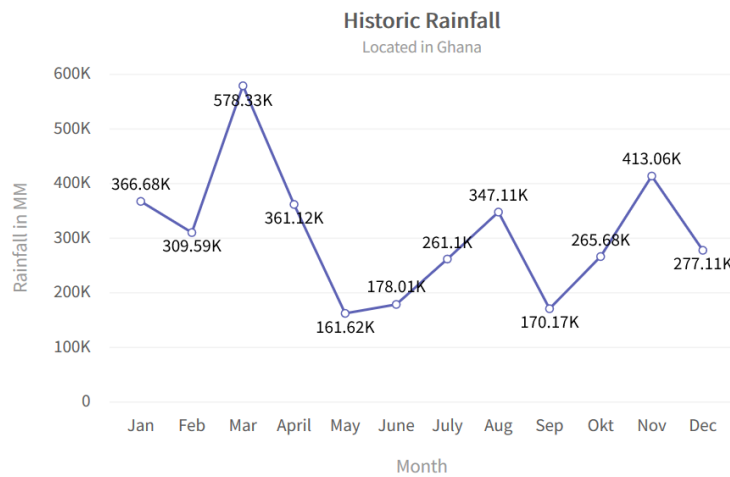


Figure 8: The historic line diagram

**Line diagram** A line diagram is a chart that displays the development of a variable of time, in this case, the historical development of rainfall throughout the year (see 'Figure 8: The historic line diagram'). This kind of view is valuable for comparing the rainfall in the present (in the bar chart (see Paragraph 4.3.3: Bar chart) with the past. The data is retrieved from the Climate Change Knowledge Portal of the World Bank Group<sup>15</sup> and contains the average of fallen rain between the years 1901 and 2016 in Ghana.

The line chart is created with Fusioncharts. The rainfall between 1901 and 2016 is averaged per month and this number is entered into the data source to be plotted onto the line chart.

<sup>15</sup>[www.climateknowledgeportal.worldbank.org/download-data#](http://www.climateknowledgeportal.worldbank.org/download-data#)

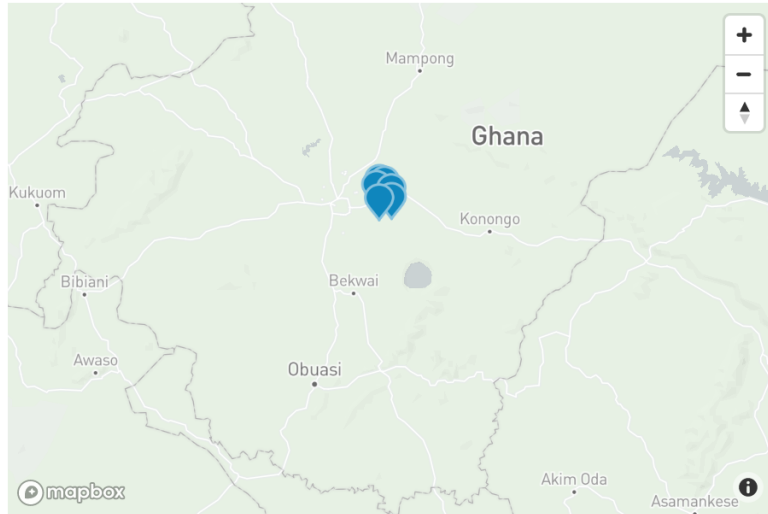


Figure 9: A map showing the locations of the farmers

**Map** For showing the locations of the farmers, a map is the most obvious choice. The map, as seen in *'Figure 9: A map showing the locations of the farmers'* uses the coordinates from the farmer to plot a marker on top of their location. In combination with the other data collected, this map will help to put a perspective on the data. For example, when it is noticed that one farmer submits larger amounts of rain this can be caused because he is located in another region of the country.

The map is created with Mapbox. In Mapbox the first step is to fill in the coordinates into a data source, which is then plotted onto a map. By incorporating the unique access token and style URL into the code, the map is loaded onto the HTML page.

The dashboard is available for not only the administrators of the service but for anyone who has a login for the service. It can be viewed at a different URL than the administrator part of the service so that the two parts can remain separate. For more information on what the administrator can do with the Petrichor Rain Service, see *'Appendix C: A guide for administrator rights'*

#### 4.3.4 Testing

The intention is to test the service locally in Ghana. A copy is made of the data collection parts and placed on a Raspberry Pi, called the Kasadaka. This Raspberry Pi is then sent to Ghana to do some tests with local farmers and/or students. But these tests are not completed in time to be incorporated into this master thesis.

To still do some tests, I decided to create some user tests in The Netherlands. I asked my colleagues to call the service and fill in some (random) data. After they call the service, I give them a small survey to fill in consisting out of five questions. Each question can be answered using a Likert scale from 1 (strongly disagree) to 10 (strongly agree). Likert scales are used to measure attitude,

providing a range of responses to a given question or statement [Jamieson et al., 2004]. They also have the possibility to fill in remarks in a blank textbox at the end of the survey.

1. The voice service is easy to understand.
2. The voice service is easy to use.
3. The given explanation is clear enough for you to understand.
4. The way to fill in your FarmerID and fallen rain works as you would expect.
5. The general impression of the voice service.

In total, 19 people tested the service and filled in the survey. Images of the graphs that came out of the survey can be found in '*Appendix B: Survey output*'. Overall the scores on the questions were fairly good. 36,8% of the respondents gave the ease to understand the service a seven, but there were also some aberrations that gave it a ten and even a four. The same goes for the second question, about the ease of use. Here there were some deviations ranging between four and ten while seven is the most picked response with 31,6%. The respondents were all in agreement and gave the explanation a grade between seven(52,6% ) and ten. Again, differences were shown when asked about entering the data. There is a four given but the mostly picked grade is with 31,6% a seven. The respondents gave the service a seven with 47,4% when asked about the general impression of the voice service.

Some remarks were given regarding how the voice service works. One respondent gave the following feedback: *The voice service mentions 'in a moment you can enter your farmer id/ fallen rain'. I expected that some announcement needed to be there before I could enter the data, this was not the case.*

While another mentioned that the period of rainfall needed to be specified: *Maybe specify the period of the rainfall, e.g. 1 day, or a specific date?*

Some suggestions were given by yet another respondent:

- *Maybe close the numbers you filled in with a hash. So it's clear in the screen which numbers you filled in.*
- *When everything has been filled in, repeat the Farmer ID and amount of rain so you are sure that what you filled in was correct.*
- *Is there an option to do it again when you made a mistake? Good luck!:)*

**User tests** Next to a survey, I also conducted a couple of user tests where I let the respondent use the service in my presence. I let them call the service and try to record their reactions and ask what they think when using the service. In total, I have done two user tests with various people who all have in common that they both have experience in IT.

**User test one** The first user test is with a student who also built a similar service for rural Africa during a course at the VU Amsterdam. She tested the service in my presence and mentioned some good and bad points.



She found the pronunciation of the voice very clear and likes the tempo with which the voice speaks, it makes it very understandable. She finds the application easy to use with understandable choices. The choices are repeated automatically when the caller doesn't choose an option, which is a plus for her because a caller might want to hear the options again. She tried to choose another option, number four, which doesn't exist and is faced with an error handling which she liked. After she chose number two, she found that everything works perfectly when you enter correct data. She appreciates the confirmation question when going to option three to end the call.

One thing that came forward when I sat next to her is that sometimes the application might give an internal error. When she starts over with the voice application, the internal error was gone. Overall she gave the impression that it is sometimes difficult to determine whether the internal error is because of a mistake she made as a caller or something inside of the service itself. After the test, one thing that came forward is that she wondered what would happen if a farmer would make a mistake in either filling in their farmer ID or fallen rain into the service.

**User test two** The second user test is done with an IT manager who also has some years of experience with developing software systems.

He mentioned that he really likes the voice of the service. He is not a native English speaker but thinks that for both native and non-native speakers the voice is very understandable. Also understandable are the options the service provides. He likes the simplicity and that you don't get too many options where you don't know the first option when you have reached the end.

After pressing one, he reached the explanation. He finds it very clear, but wonders if maybe an extended explanation could be created for the farmers who would like to know exactly what happens to the data collected and where they could view the gathered data.

After the explanation, he went back to the main menu and into option two. Overall the process is clear, especially when he would become more familiar with filling it in multiple times per week. He did mention that at first, it is not clear when he can begin typing his data. He is curious about what will happen when data is filled in wrong or what will happen when he fills in too many characters. After I informed him that nothing is yet built in to deal with faulty data, he gave the feedback that the user might like to hear the data they have filled in to check it before they have to confirm it. Only then, the data is stored in the database.

## 4.4 Sustainability assessment

The Sustainability assessment looks at two important features when trying to get the Petrichor Rain Service to work in rural Africa. First, the feasibility in '*Subsection 4.4.1: Feasibility*' to look if it is practical to implement this kind of service in rural Africa. '*Subsection 4.4.2: Implementation*' acknowledges the process of implementing the Petrichor Rain Service in rural Africa.

### 4.4.1 Feasibility

The feasibility of the Petrichor Rain Service can be divided into the technical feasibility and the business and (socio-) economic feasibility. The technical feasibility is about if the service would

work in the proposed environment while the economic feasibility pays more attention to the costs of implementation.

For the implementation of the Petrichor Rain Service in Ghana, a couple of points need to be taken into account which can have an impact on a successful utilization. The first point is the availability of a Kasadaka<sup>16</sup>. The Kasadaka is a Raspberry Pi, a low-cost computer which runs on Linux. In combination with a GSM dongle with SIM card, the Kasadaka creates a GSM network which the farmers can call towards. The SIM card should contain a Ghanaian number to make the costs for the calls for the farmers as low as possible. If needed, an external screen can be added to make using the Kasadaka more convenient.

Another point for notice is the availability of an administrator. An administrator for the Petrichor Rain Service is needed for the maintenance of the service. The maintenance can include adding new farmers, explaining what they have to do, share FarmerID's and giving access to the dashboard. The administrator should also have some technical knowledge to solve any (minor) errors that might occur during the deployment of the service in the field. A possibility would be someone from a Non-governmental organization or someone who works at the radio station. The Kasadaka should be located in a central place. Somewhere where all the farmers have access to call to and where the administrator can access the Kasadaka in person when needed.

The costs for the implementation of the Petrichor Rain Service can be divided into three main components:

- Maintenance costs
- Infrastructure costs
- User costs

**Maintenance costs** As of now, there are no development or maintenance costs because this is a master thesis. However, if the proposed service turns out to be a desired addition for the farmers in the field, it would be very interesting to see if continuity of this project is possible. In order to support the maintenance costs that would be made, an assessment needs to be done to see if the farmers, Non-governmental organizations or the radio stations would be willing to pay a small amount of money each time they use the service. Because of the variety of users, it is difficult to determine which user is the main user and who would have to pay money to keep this service working. The money obtained from this would partially fund new development and needed maintenance at a later point in time.

**Infrastructure costs** The infrastructure costs of the Petrichor Rain Service would consist out of the purchase of a Kasadaka, a GSM dongle and a SIM card. A Kasadaka or Raspberry Pi can already be purchased between GHS 255.00 (€42) and GHS 396.00 (€65)<sup>17</sup> in Ghana or between €40 and €60 in The Netherlands at the moment of writing<sup>18</sup>. Within The Netherlands, a supported GSM dongle can be bought for somewhere between €21 and €57 depending on the model. In

---

<sup>16</sup>[www.kasadaka.com](http://www.kasadaka.com)

<sup>17</sup>[www.inventelectronics.com/product-category/raspberry-pi/](http://www.inventelectronics.com/product-category/raspberry-pi/)

<sup>18</sup>[www.tweakers.net/pricewatch/492795/raspberry-pi-3-model-b-1gb.html](http://www.tweakers.net/pricewatch/492795/raspberry-pi-3-model-b-1gb.html)

Ghana, the prices range between GHS 134 (€22) GHS 282 (€46)<sup>19</sup>. The SIM card needs to be purchased in Ghana because it needs to contain a Ghanaian number. SIM cards can be bought in Ghana for under €5.

The (average) total costs to implement the Petrichor Rain Service in Ghana would lie around €94 when everything is bought in The Netherlands and GHS 563 (€92) when the items are purchased locally in Ghana.

**User costs** The service itself will mainly be used by local farmers who like to submit their rain data to the service and by radio broadcasters and interested parties who like to view the dashboard. Right now, the only costs the farmer will make is the cost of a phone call to the provided number of the Kasadaka containing the Petrichor Rain Service. However, if the service will continue in the future, small donations might be asked to keep the service running.

#### 4.4.2 Implementation

The first step that needs to be taken to implement the service locally in Ghana is to prepare the Kasadaka. The Raspberry Pi itself needs to be installed with Raspbian, the official operating system for the Raspberry Pi. Next to Raspbian, Asterisk, chan\_dongle, Voximal and Apache2 need to be installed as well. These are all programs and services needed to properly run the Petrichor Rain Service on the Kasadaka. After the Kasadaka is prepared, an image of the Petrichor Rain Service needs to be placed onto the Kasadaka. Once this is done, the Kasadaka must be brought to Ghana, preferably by someone who is able to install the remaining components and solve any errors that might occur during the first tests. The local administrator needs to be involved in this process. After the Kasadaka is prepared and the GSM dongle with the SIM card is installed and works, the Kasadaka needs to be placed on its strategically chosen location to create an optimal connection between the Kasadaka and the farmers.

As mentioned before in the technical feasibility, the two main concerns for the implementation of the Petrichor Rain Service in Ghana are the availability of an administrator and a Kasadaka for the hosting of the service. After both of these points have been taken care of, farmers need to be found who are willing to invest their time to cooperate with the service. It is important to start with a small group of farmers (around 5-10) as some kind of pilot or beta-testing phase. Data can be collected from the farmers and the administrator can monitor the use of the service to see if there are any technical issues, like errors occurring or how the Kasadaka deals with the load and the environmental conditions. Based on the feedback from the pilot farmers, redesigns of the service can take place before the service can be released to a larger group of farmers.

## 5 Discussion & conclusion

*'Subsection 5.1: Discussion'* tells what this research means and why it is important, while *'Subsection 5.2: Conclusion'* gives the conclusion of this research.

---

<sup>19</sup>[www.jumia.com.gh/computer-external-component/](http://www.jumia.com.gh/computer-external-component/)

## 5.1 Discussion

There are many points that need to be taken into account in order to create a properly working system for Rural Africa. Like the (lack of) use of internet, language, mobile devices and the ease with which the service can be operated. Knowing these bottlenecks, building a good solution to collect data from the local users and also give the data back to them is a challenge, just as the implementation. Therefore, this challenge is chosen as the main research question for this research.

Although this research can be seen as a confined research, it does not stand entirely on its own. It should also be seen as an extension to current research. The research from Francis Ditto for example where he looks at how people in developing countries can get access to information that suits their needs. But also the research from André Baart can be seen as preliminary research. Where he created the Kasadaka, a Raspberry Pi suited for use in rural Africa, and the Voice Service Development Kit (VSDK), a voice-service development kit that uses voice interaction building-blocks for the creation of voice services. But also some applications built with the VSDK, a couple of requirements and some locally conducted workshops all provide valuable input for this research. What is missing from the already executed research is a better way to submit data with a mobile phone. So the Petrchor Rain Service is developed and tested.

A service to collect rain data from African farmers with the help of DTMF (dual tone multi-frequency), the signal that the phone generates when one of the mobile phone's keys is pressed. The developed input method that is used to collect rain data can also be used to retrieve other information from local users and to get that data stored in a central place. Available hardware in the form of a Kasadaka with GSM dongle and enough users are of great importance to make the implementation of the Petrchor Rain Service in Ghana a success.

During this research, some attention has been given to testing the Petrchor Rain Service with outside users. A survey and some user tests were the methods to ask feedback from Dutch users. Overall the users were very pleased with the service, but they still proposed some feedback to make the service more user-friendly like asking the user for a confirmation before the submitted data is stored permanently. To perform fast and reliable tests, the tests were only conducted in the Netherlands. Because of a lack of testing the service in the field in rural Africa, no extra iteration could be done where the feedback of the users is implemented into the current version of the service. This is also a point for further research into this subject: retrieve feedback from local African users and improve the service based on their findings. The Petrchor Rain Service can also be extended with more features like sending users reminders when they haven't made a call to the system during a certain period, give users an option for adjusting their data before submitting and extend the service with more ways in which the submitted data can be viewed and shared with interested parties.

## 5.2 Conclusion

This research gives an answer to the question if it is possible to design a data collecting system that allows local weather data sharing among farmers in rural Africa. This system is an extension to the existing Voice Service Development Kit with modules to input data with DTMF and output data via a dashboard. Although the question is focused on a data collecting system to share local weather data, the proposed solution would also work for the collection of other kinds of data, as long

as users can use the keypads of their mobile phones. A search for the most important requirements yields that the internet connection, language, ease of use and the mobile devices available play an important role when creating a voice-based service for rural Africa. Prior to this research, little research is been done on how farmers can send their own collected data in their own language without the need for internet. The researches that have been executed on the subject of weather data all show that locally collected weather data is not as widely accessible as it is in more developed countries, but the need for this kind of data is high. With the development of the Petrichor Rain Service, both of these needs, sending their own collected data without internet and share locally collected weather data, are combined.

To continue the development of the Petrichor Rain Service a few steps still need to be taken. First, more extensive tests need to be done on location in rural Africa to determine what the eventual users think of the service. For example, a couple of farmers who live in the same village. Once this is done, their feedback needs to be evaluated and the service can be altered accordingly. Then the user group of the service can be extended to include more farmers located in a larger area. Eventually, the Petrichor Rain Service hopes to not only be a solution for weather data collection but to be the beginning for more development of these kinds of data retrieval services.

## References

- [Amarnath et al., 2018] Amarnath, G., Simons, G., Alahacoon, N., Smakhtin, V., Sharma, B., Gismalla, Y., Mohammed, Y., and Andriessen, M. (2018). Using smart ict to provide weather and water information to smallholders in africa: The case of the gash river basin, sudan. *Climate Risk Management*, 22:52–66.
- [Asghar et al., 2016] Asghar, A. R., Tabassum, A., Bhatti, S. N., Sultan, Z., and Abbas, R. (2016). Role of requirements elicitation & prioritization to optimize quality in scrum agile development.
- [Baart, 2017] Baart, A. (2017). *KasaDaka: a sustainable voice-service platform*. PhD thesis, Master Thesis Vrije Universiteit Amsterdam.
- [Baart et al., 2018] Baart, A., Bon, A., de Boer, V., Tuijp, W., and Akkermans, H. (2018). Ney yibeogo–hello world: A voice service development platform to bridge the web’s digital divide. In *WEBIST 2018, 14th International Conference on Web Information Systems and Technologies, Sevilla, Spain*.
- [Bodomo, 1996] Bodomo, A. B. (1996). On language and development in africa: The case of ghana. *Nordic journal of African studies*, 5(2):31–51.
- [Bon, 2019] Bon, A. (2019). *Intervention or Collaboration?* Pangea, first edition.
- [Bon et al., 2016] Bon, A., Akkermans, H., and Gordijn, J. (2016). Developing ict services in a low-resource development context. *CSIMQ*, 9:84–109.
- [Bon et al., 2017] Bon, A., Boer, V. D., Aart, C. V., Akkermans, H., Lô, G., Baart, A., Blankendaal, R., Robenhagen, A., Ferguson, J., Wekken, M. V. D., Schlobach, S., Gordijn, J., Tuyp, W., Gyan, N. B., Dittoh, F., Tangara, A., Tessougué, A., Ouedraogo, J., Ouedraogo, M., and Dittoh, S. (2015-2017). W4ra use cases/services co-created for/with african farmers and radios.
- [Bornman, 2016] Bornman, E. (2016). Information society and digital divide in south africa: results of longitudinal surveys. *Information, Communication & Society*, 19(2):264–278.
- [Cameron and Pierce, 1994] Cameron, J. and Pierce, W. D. (1994). Reinforcement, reward, and intrinsic motivation: A meta-analysis. *Review of Educational research*, 64(3):363–423.
- [Commission, 2012] Commission, D. P. (2012). *Act of the Parliament of the Republic of Ghana Entitled Data Protection Act, 2012*. <https://www.dataprotection.org.gh/sites/default/files/Data%20Protection%20Act%20%2C%202012%20%28Act%20843%29.pdf>.
- [Conway, 2018] Conway, P. (2018). The effective use of graphs and charts on dashboards.
- [Cruz et al., 2016] Cruz, A. R., Garraton, S. D., Jose, J. M., Malolos, A., Panganiban, A., and Caldo, R. B. (2016). Model case of remote switching application system via telephone line using dtmf. *DLSU Research Congress*, 4.
- [De Boer et al., 2015] De Boer, V., Gyan, N. B., Bon, A., Tuyp, W., Van Aart, C., and Akkermans, H. (2015). A dialogue with linked data: Voice-based access to market data in the sahel. *Semantic Web*, 6(1):23–33.
- [DuBois, 2008] DuBois, P. (2008). *MySQL*. Pearson Education.

- [Faraway and Augustin, 2018] Faraway, J. J. and Augustin, N. H. (2018). When small data beats big data. *Statistics & Probability Letters*, 136:142–145.
- [Fox and Rockström, 2003] Fox, P. and Rockström, J. (2003). Supplemental irrigation for dry-spell mitigation of rainfed agriculture in the sahel. *Agricultural water management*, 61(1):29–50.
- [Franzblau and Chung, 2012] Franzblau, L. E. and Chung, K. C. (2012). Graphs, tables, and figures in scientific publications: the good, the bad, and how not to be the latter. *The Journal of hand surgery*, 37(3):591–596.
- [Frimpon and Adaku, 2017] Frimpon, M. F. and Adaku, E. (2017). A comparison of high-speed internet service in ghana: an analytical hierarchy approach. *Information Technology & People*, 31(1):181–198.
- [Getu et al., 2015] Getu, B. N., Hamad, N. A., and Attia, H. A. (2015). Remote controlling of an agricultural pump system based on the dual tone multi-frequency (dtmf) technique. *Journal of Engineering science and Technology*, 10(10):1261–1274.
- [Greenwald et al., 2013] Greenwald, R., Stackowiak, R., and Stern, J. (2013). *Oracle essentials: Oracle database 12c*. O’Reilly Media, Inc.
- [Gyan, 2016] Gyan, N. B. (2016). *The Web, Speech Technologies and Rural Development in West Africa An ICT4D Approach*. Amsterdam: Vrije Universiteit.
- [Gyan et al., 2013] Gyan, N. B., De Boer, V., Bon, A., Van Aart, C., Akkermans, H., Boyera, S., Froumentin, M., Grewal, A., and Allen, M. (2013). Voice-based web access in rural africa. In *Proceedings of the 5th annual acm web science conference*, pages 122–131. ACM.
- [Hans Akkermans, 2017] Hans Akkermans, Anna Bon, A. B. W. T. M. O. J. O. (2017). W4ra workshop gourcy-meteo, w4ra trip, tmt-bf second workshop on meteo is.
- [Hans-Erik Eriksson, 2000] Hans-Erik Eriksson, M. P. (2000). *Business Modeling with UML: Business Patterns at Work*. Wiley, first edition.
- [Hendijani et al., 2016] Hendijani, R., Bischak, D. P., Arvai, J., and Dugar, S. (2016). Intrinsic motivation, external reward, and their effect on overall motivation and performance. *Human Performance*, 29(4):251–274.
- [Holovaty and Kaplan-Moss, 2009] Holovaty, A. and Kaplan-Moss, J. (2009). *The definitive guide to Django: Web development done right*. Apress.
- [Jamieson et al., 2004] Jamieson, S. et al. (2004). Likert scales: how to (ab) use them. *Medical education*, 38(12):1217–1218.
- [Kitchin et al., 2016] Kitchin, R., Maalsen, S., and McArdle, G. (2016). The praxis and politics of building urban dashboards. *Geoforum*, 77:93–101.
- [Loshin, 2012] Loshin, D. (2012). *Business intelligence: the savvy manager’s guide*. Newnes.
- [Nerkar, 2016] Nerkar, A. D. (2016). Business analytics (ba): Core of business intelligence (bi). *International Journal of Advanced Engineering, Management and Science*, 2(12).
- [Owens and Allen, 2010] Owens, M. and Allen, G. (2010). *SQLite*. Springer.

- [Parliament, 2016] Parliament, T. E. (2016). *Regulation (EU) 2016/679 of the European Parliament and of the Council*. <https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng>.
- [Plekhanova, 2009] Plekhanova, J. (2009). Evaluating web development frameworks: Django, ruby on rails and cakephp. *Institute for Business and Information Technology*.
- [Poushter, 2017] Poushter, J. (2017). Smartphone ownership and internet usage continues to climb in emerging economies.
- [Riggs et al., 2015] Riggs, S., Ciolli, G., Krosing, H., and Bartolini, G. (2015). *PostgreSQL 9 administration cookbook*. Packt Publishing Ltd.
- [Silver and Johnson, 2018] Silver, L. and Johnson, C. (2018). Majorities in sub-saharan africa own mobile phones, but smartphone adoption is modest.
- [Unesco, 2018] Unesco (2018). *Data for the sustainable development goals*. <http://uis.unesco.org/en/country/gh>.
- [Unicef, 2016] Unicef (2016). *State of the world’s children country statistical information*. <https://data.unicef.org/resources/state-worlds-children-2016-statistical-tables/>.
- [Uzuegbu, 2016] Uzuegbu, C. P. (2016). Effective information service delivery to rural dwellers in sub-saharan africa: Whose job? *IFLA journal*, 42(1):49–58.
- [van de Giesen et al., 2014] van de Giesen, N., Hut, R., and Selker, J. (2014). The trans-african hydro-meteorological observatory (tahmo). *Wiley Interdisciplinary Reviews: Water*, 1(4):341–348.
- [Yeboah, 2017] Yeboah, S. (2017). How a lack of access to reliable weather data is hurting african farmers. *The Conversation*.



# Appendix

## Table of Contents

---

<b>A</b>	<b>Appendix 1: The class diagram</b>	<b>2</b>
<b>B</b>	<b>Appendix 2: Survey output</b>	<b>3</b>
B.1	Question 1: The voice service is easy to understand . . . . .	3
B.2	Question 2: The voice service is easy to use . . . . .	3
B.3	Question 3: The given explanation is clear enough for you to understand . . . . .	4
B.4	Question 4: The way to fill in your FarmerID and fallen rain works as you would expect . . . . .	4
B.5	Question 5: The general impression of the voice service . . . . .	5
<b>C</b>	<b>Appendix 3: A guide for administrator rights</b>	<b>6</b>
C.1	Log in & and the user interface . . . . .	7
C.2	Add new farmer . . . . .	7
C.3	Check current farmers & Share FarmerID . . . . .	9
C.4	Look at dashboard & share dashboard . . . . .	11
<b>D</b>	<b>Appendix 4: A Manual for adding a new service</b>	<b>13</b>
D.1	Choice Elements . . . . .	19
D.2	Message Presentation Elements . . . . .	21
D.3	User Input Categories . . . . .	23
D.4	Input Elements . . . . .	25
<b>E</b>	<b>Appendix 5: A Manual for adding a new language</b>	<b>28</b>

---

## A Appendix 1: The class diagram

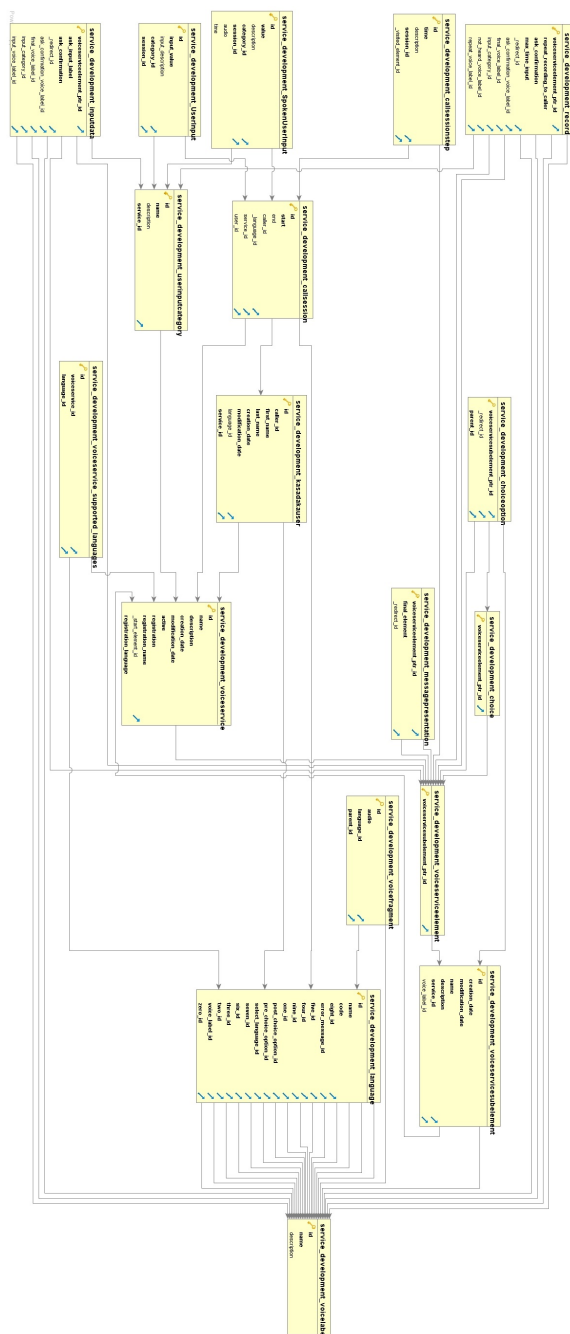


Figure 10: The class diagram

## B Appendix 2: Survey output

### B.1 Question 1: The voice service is easy to understand

The voice service is easy to understand

19 responses

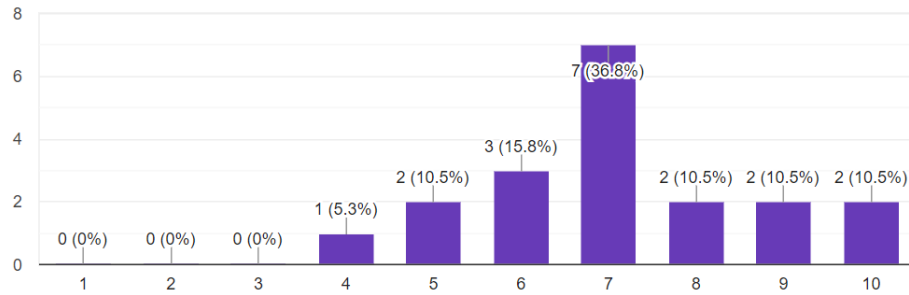


Figure 11: The voice service is easy to understand

### B.2 Question 2: The voice service is easy to use

The voice service is easy to use

19 responses

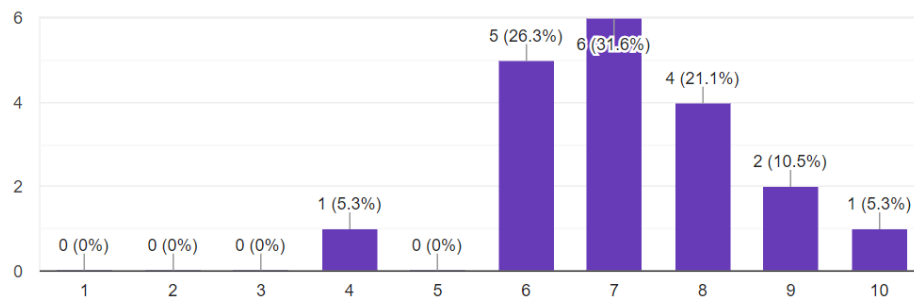


Figure 12: The voice service is easy to use

### B.3 Question 3: The given explanation is clear enough for you to understand

The given explanation is clear enough for you to understand

19 responses

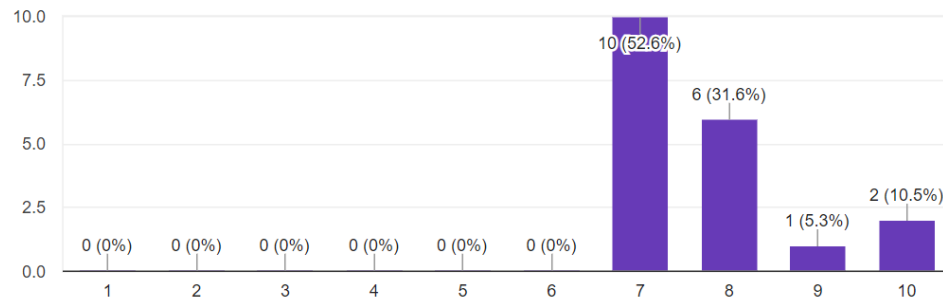


Figure 13: The given explanation is clear enough for you to understand

### B.4 Question 4: The way to fill in your FarmerID and fallen rain works as you would expect

The way to fill in your FarmerID and fallen rain works as you would expect

19 responses

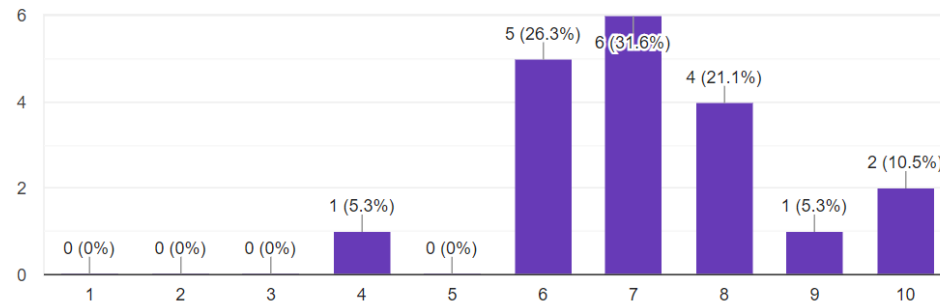


Figure 14: The way to fill in your FarmerID and fallen rain works as you would expect

## B.5 Question 5: The general impression of the voice service

### The general impression of the voice service

19 responses

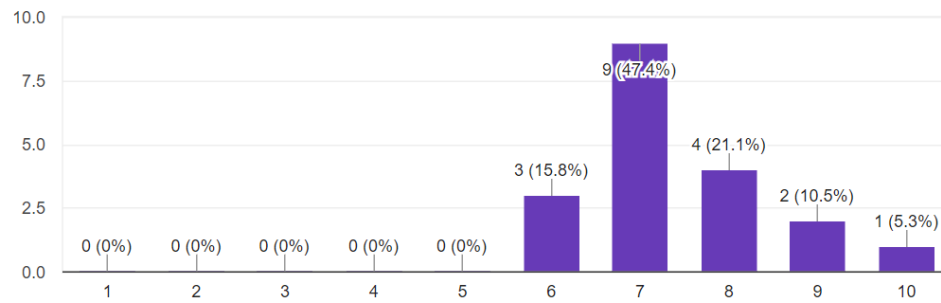


Figure 15: AThe general impression of the voice service

## C Appendix 3: A guide for administrator rights

The following image (*'Figure 16: The admin use case'*) shows the different responsibilities of the administrator of the Petrichor Rain Service. Each of the responsibilities is explained in depth in the following sections.

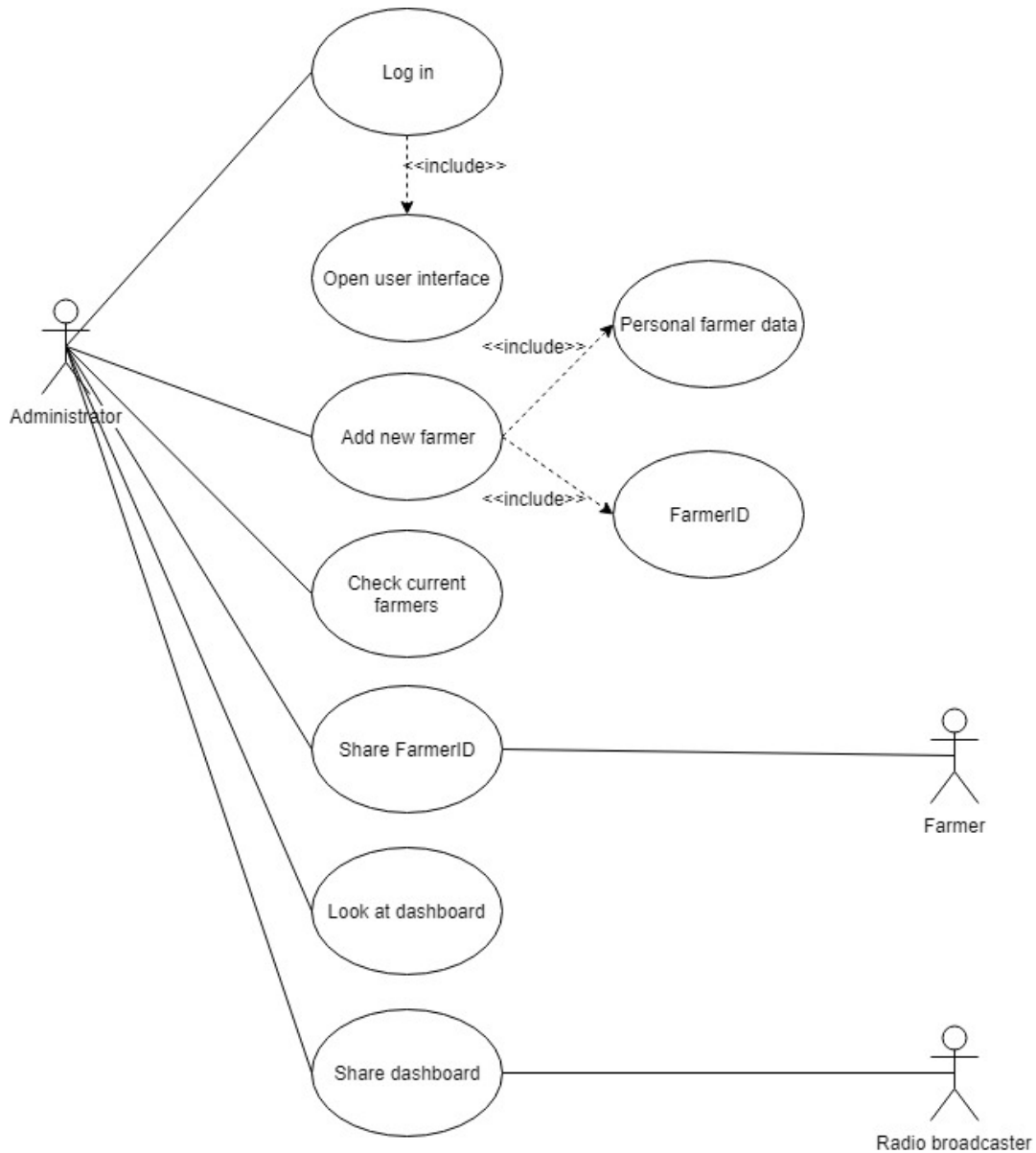


Figure 16: The Admin Use Case

## C.1 Log in & and the user interface

The admin has the possibility to log in into the admin page of the service via the following link and credentials: <http://petrichor-rain-system.herokuapp.com/login/?next=/>

**Username:** Administrator

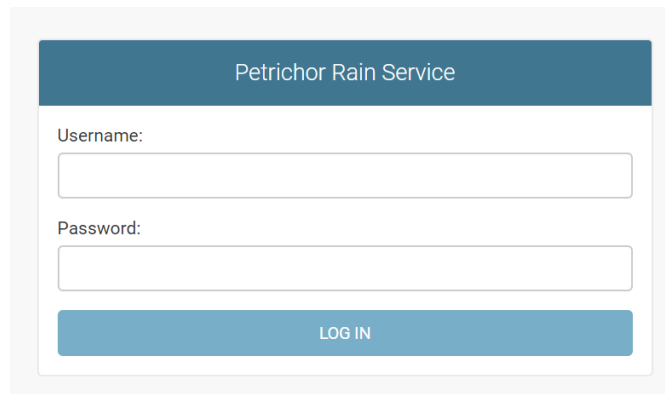
**Password:** Petrichor123

This gives access to all administrator tasks, such as creating new services, adding farmers and giving access to the dashboard.

## C.2 Add new farmer

The admin can add new farmers to the service. The following steps need to be taken to do so:

1. Log in into the application with an admin account.



Petrichor Rain Service

Username:

Password:

LOG IN

Figure 17: The login page

2. Go to the Callers sub menu.

## Petrichor Rain Service

### Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Users	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
VOICE SERVICE DEVELOPMENT		
Call Sessions		<a href="#">✎ Change</a>
Callers	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Choice Elements	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Input Elements	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Languages	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Message Presentation Elements	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
User Input Categorys	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
User inputs		<a href="#">✎ Change</a>
Voice Labels	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Voice Services	<a href="#">+ Add</a>	<a href="#">✎ Change</a>

Figure 18: The main menu

3. Click on the 'Add Caller' button.



Figure 19: The add caller button

4. Fill in the required fields
  - (a) Add the general information like their phone number and name. Their FarmerID is automatically generated.



## Add Caller

General	
Phone number:	<input type="text"/>
FarmerID:	2
First name:	<input type="text"/>
Last name:	<input type="text"/>

Figure 20: The general section

- (b) Fill in the farmers location information like the city they are located in, the region, country and their GPS coordinates if applicable.

Location	
City:	<input type="text"/>
Region:	<input type="text"/>
Country:	<input type="text"/>
Latitude:	<input type="text"/>
Longitude:	<input type="text"/>

Save and add another Save and continue editing SAVE

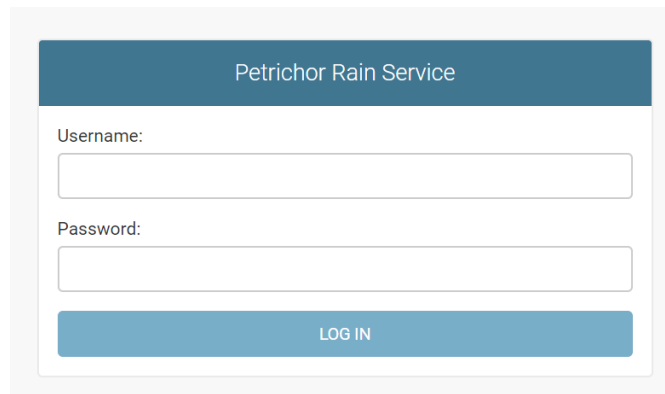
Figure 21: The location section

5. Save the farmer and create more if needed.

### C.3 Check current farmers & Share FarmerID

Next to adding new farmers, the admin can also look at and alter existing farmers. This requires the following steps:

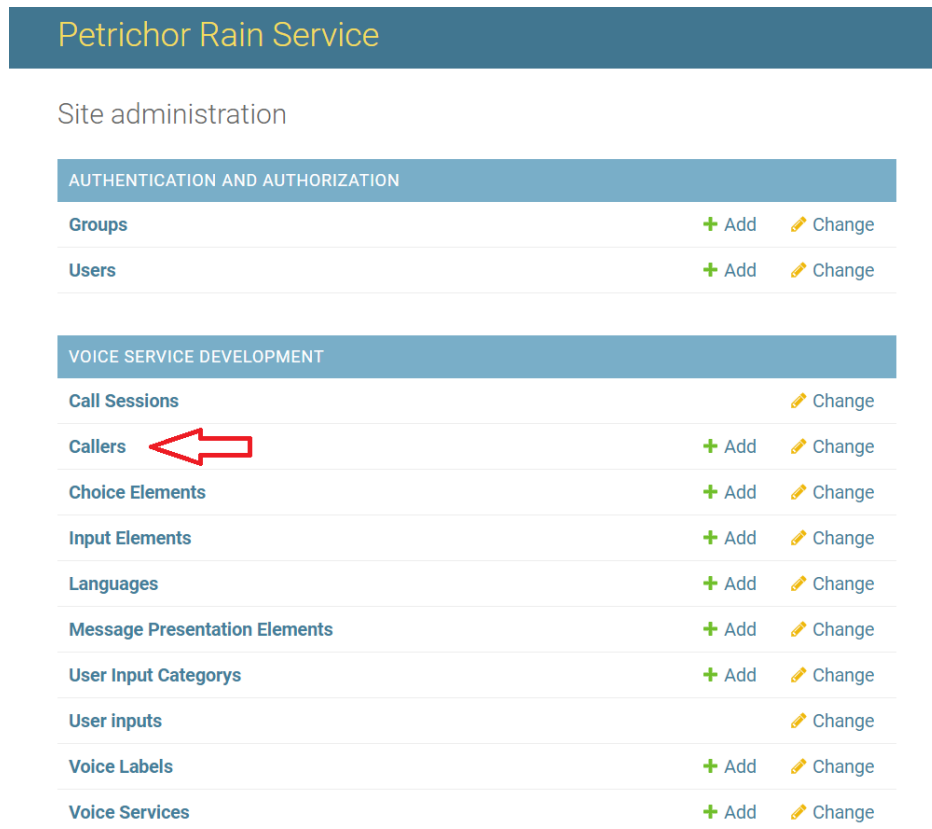
1. Log in into the application with an admin account.



The login page for Petrichor Rain Service. It features a dark blue header with the text "Petrichor Rain Service". Below the header, there are two input fields: "Username:" and "Password:". At the bottom, there is a blue button labeled "LOG IN".

Figure 22: The login page

2. Go to the Callers sub menu.



The main menu for Petrichor Rain Service. It has a dark blue header with the text "Petrichor Rain Service". Below the header, there is a section titled "Site administration". Under this section, there are two main categories: "AUTHENTICATION AND AUTHORIZATION" and "VOICE SERVICE DEVELOPMENT".

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	✎ Change
Users	+ Add	✎ Change

VOICE SERVICE DEVELOPMENT		
Call Sessions		✎ Change
Callers	+ Add	✎ Change
Choice Elements	+ Add	✎ Change
Input Elements	+ Add	✎ Change
Languages	+ Add	✎ Change
Message Presentation Elements	+ Add	✎ Change
User Input Categorys	+ Add	✎ Change
User inputs		✎ Change
Voice Labels	+ Add	✎ Change
Voice Services	+ Add	✎ Change

A red arrow points to the "Callers" item in the "VOICE SERVICE DEVELOPMENT" section.

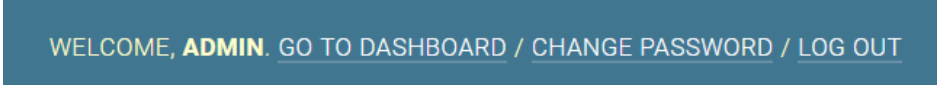
Figure 23: The main menu

3. Click on the farmers name you'd like to view and/or alter.
4. Alter the needed fields.
5. Save the farmer.

When a farmer is added to the service, it is important that the farmer is informed about their FarmerID as this is their method of identification when they start using the service.

## C.4 Look at dashboard & share dashboard

To look at the dashboard, the administrator can follow the link located on the top right side of the header.

A horizontal blue bar with white text. The text reads: 'WELCOME, ADMIN. GO TO DASHBOARD / CHANGE PASSWORD / LOG OUT'. The word 'ADMIN' is in bold. The links 'GO TO DASHBOARD', 'CHANGE PASSWORD', and 'LOG OUT' are underlined.

WELCOME, **ADMIN**. [GO TO DASHBOARD](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Figure 24: The admin header

To give outsiders access to the Petrichor Rain Service Dashboard, the administrator can provide them with the following link: <https://petrichor-rain-system.herokuapp.com/dashboard/>

This leads them to the HTML page found in '*Figure 25: The Petrichor Rain Service Data Dashboard*'

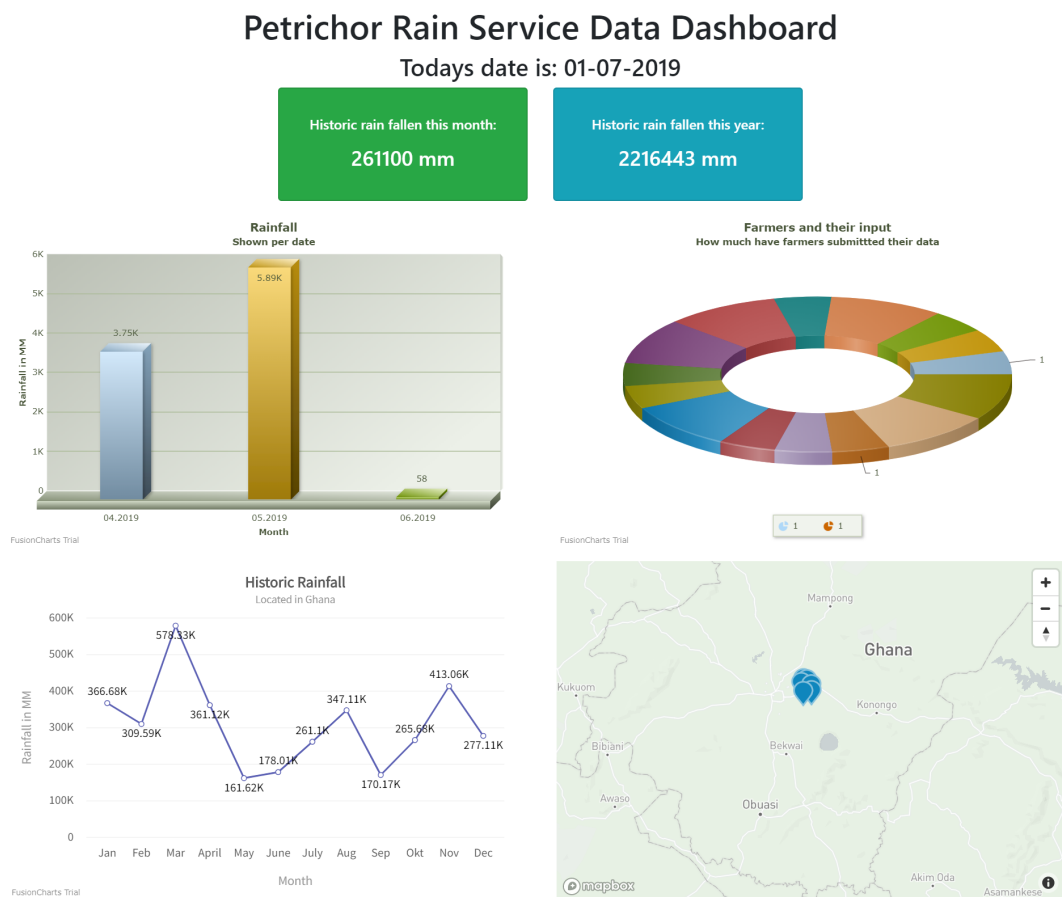
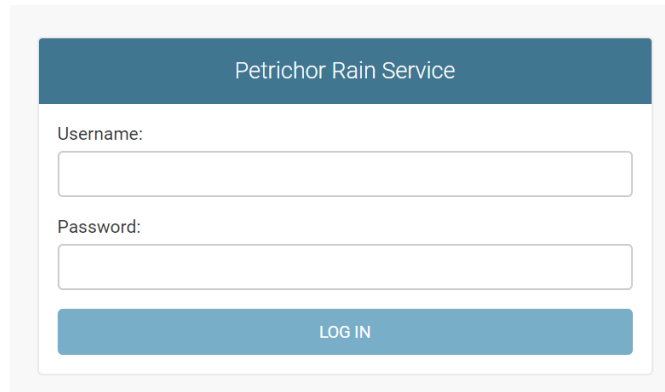


Figure 25: The Petrichor Rain Service Data Dashboard

More explanation about the dashboard and the different charts can be found in '*Subsection 4.3.3: Output*'.

## D Appendix 4: A Manual for adding a new service

1. Create a flow diagram to determine how the service is going to look like.
2. Log in into the application with an admin account.

The image shows a login form for 'Petrichor Rain Service'. It has a dark blue header with the service name. Below the header, there are two input fields: 'Username:' and 'Password:'. At the bottom of the form is a blue button labeled 'LOG IN'. The entire form is set against a light gray background.

Petrichor Rain Service

Username:

Password:

LOG IN

Figure 26: The login page

3. Go to the Voice Service sub menu.

## Petrichor Rain Service

### Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Users	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
VOICE SERVICE DEVELOPMENT		
Call Sessions		<a href="#">✎ Change</a>
Callers	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Choice Elements	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Input Elements	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Languages	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Message Presentation Elements	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
User Input Categorys	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
User inputs		<a href="#">✎ Change</a>
Voice Labels	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Voice Services	<a href="#">+ Add</a>	<a href="#">✎ Change</a>

Figure 27: The main menu

- Click on the button 'Add Voice Service'.

ADD VOICE SERVICE +

Figure 28: The add voice service button

- Fill in the required fields.
  - Fill the general section.

## Add Voice Service

The screenshot shows the 'General' tab of the 'Add Voice Service' form. It includes fields for 'Name' and 'Description', a 'VoiceXML endpoint URL' field with the value 'unknown', a checkbox for 'Voice service active' with explanatory text, 'Is valid:' and 'Validation errors:' fields both showing a dash, and a 'Supported languages:' section with a dropdown menu containing 'Nederlands (nl)', 'English (en)', and 'Francais (fr)'. A green plus icon is next to the dropdown. Below the dropdown, a note says: 'Hold down "Control", or "Command" on a Mac, to select more than one.'

General

Name:

Description:

VoiceXML endpoint URL: unknown

☐ Voice service active  
A voice service that is active is accessible to users. Marking this service as active (which is only possible when it is valid) will activate this service and deactivate all other services.

Is valid: -

Validation errors: -

Supported languages: 

Nederlands (nl)  
English (en)  
Francais (fr)

 +

Hold down "Control", or "Command" on a Mac, to select more than one.

Figure 29: The general voice service section

(b) Fill registration process section.

The screenshot shows the 'Registration process' and 'Call flow' sections of the 'Add Voice Service' form. The 'Registration process' section has a 'User registration:' dropdown and a checked checkbox for 'Register Language preference' with explanatory text. The 'Call flow' section has a 'Starting element:' dropdown. At the bottom, there are three buttons: 'Save and add another', 'Save and continue editing', and 'SAVE'.

Registration process

User registration:

☒ Register Language preference  
The preferred language will be asked and stored during the user registration process

Call flow

Starting element:

Save and add another Save and continue editing SAVE

Figure 30: The registration process section

(c) Continue the Call flow by adding with which element the voice service needs to start.

(d) Save the voice service by clicking on the save button.

6. Create the needed speech audio files.

(a) , For example, use: **notevibes.com** and create an account.

Convert Text to Speech (WAV)

Paste Text Here or Write something...

Number of characters: 2000/0

Select Voice English(US) - David Voice Speed 1

audioProfile Default Pitch 0

Convert

Play Your File

0:00 / 0:00

Download file

Download Wav

Figure 31: The main page of **notevibes.com**

- (b) Switch to WAV download.
- (c) Fill in the text you want to turn to speech in the textbox.
- (d) Choose voice and voice speed.
- (e) Click convert.
- (f) Listen to the file and when you are satisfied, click on Download WAV.
- (g) Use a file converter to convert the WAV to the appropriate settings. For example **g711.org**.



This free tool will convert just about any DRM-free media file into audio that's compatible with BroadWorks or Asterisk Music on Hold and IVR Announcements.

Source File

**Browse**

Note: 50MB Maximum File Size

Output Format

☐ BroadWorks Classic (8Khz, Mono, u-law)  
☐ BroadWorks 17sp4+ SD (8Khz, Mono, 16-Bit PCM)  
☐ BroadWorks 17sp4+ HD (16Khz, Mono, 16-Bit PCM)  
☒ Asterisk Standard (8Khz, Mono, 16-Bit PCM)  
☐ Asterisk HD (16Khz, Mono, G.722)  
☐ Asterisk G.729 (8Khz, Mono, G.729)  
☐ Asterisk RAW (8Khz, Mono, RAW)

Volume

☐ Quiet   ☐ Lower   ☒ Medium   ☐ High   ☐ Maximum

☒ Optimize Audio for Phone (Bandpass Filter)

4f312d904ea87311-AMS

**Reset**   **Submit**

Figure 32: The main page of `g711.org`

- (h) Select the source file.
  - (i) Choose 'Asterisk Standard' and Volume as Medium.
  - (j) Press submit and download the file.
7. Go to the Voice Label sub menu.

## Petrichor Rain Service

### Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Users	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
VOICE SERVICE DEVELOPMENT		
Call Sessions		<a href="#">✎ Change</a>
Callers	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Choice Elements	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Input Elements	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Languages	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Message Presentation Elements	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
User Input Categorys	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
User inputs		<a href="#">✎ Change</a>
Voice Labels	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Voice Services	<a href="#">+ Add</a>	<a href="#">✎ Change</a>

Figure 33: The main menu

- Click on the button 'Add Voice Label'.

ADD VOICE LABEL +

Figure 34: The add voice label button

- Fill in the required fields.

Petrichor Rain Service
WELCOME, NATAS VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Voice Service Development > Voice Labels > Add Voice Label

### Add Voice Label

Name:

Description:

VOICE FRAGMENTS				
LANGUAGE	IS VALID	AUDIO	AUDIO FILE PLAYER	DELETE?
<div> <div></div> <div></div> </div>	✓	<div>Bestand kiezen</div> <div>Geen bestand gekozen</div>	None	
<div> <div></div> <div></div> </div>	✓	<div>Bestand kiezen</div> <div>Geen bestand gekozen</div>	None	

+ Add another Voice Fragment

Save and add another

Save and continue editing

SAVE

Figure 35: The voice label page

- (a) Add name and description.
  - (b) Add the different voice fragments and select the appropriate language.
  - (c) Save the voice label and create more if needed.
10. According to the flow diagram, create Choice Elements, Message Presentation Elements, User Input Categories and Input Elements.
  11. Make sure to link the Elements in the proper way as written down in the flow diagram. Also, note that it is not possible to put two input element directly behind each other as they will create an infinite loop.

## D.1 Choice Elements

1. Go to the Choice Elements sub menu.

## Petrichor Rain Service

### Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Users	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
VOICE SERVICE DEVELOPMENT		
Call Sessions		<a href="#">✎ Change</a>
Callers	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Choice Elements	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Input Elements	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Languages	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Message Presentation Elements	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
User Input Categorys	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
User inputs		<a href="#">✎ Change</a>
Voice Labels	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Voice Services	<a href="#">+ Add</a>	<a href="#">✎ Change</a>

Figure 36: The main menu

- Click on the button 'Add Choice Element'.

ADD CHOICE ELEMENT +

Figure 37: The add coice element button

- Fill in the required fields.
  - Fill in the general fields.

## Add Choice Element

General	
Name:	<input type="text"/>
Description:	<input type="text"/>
Service:	<div><div>-----</div><div>✎ +</div></div> <div>The service to which this element belongs</div>
Is valid:	<div>✖</div>
Validation errors:	No VoiceLabel in: ""
Voice label:	<div><div>-----</div><div>✎ + ✖</div></div>

Figure 38: The general information of the choice element

(b) Add the possible choices in the order you want to offer them to the user.

POSSIBLE CHOICES				
SERVICE ⓘ	NAME	DESCRIPTION	VOICE LABEL	REDIRECT ELEMENT
<div><div>-----</div><div>✎ +</div></div>	<input type="text"/>	<input type="text"/>	<div><div>-----</div><div>✎ + ✖</div></div>	<input type="text"/>
<div><div>-----</div><div>✎ +</div></div>	<input type="text"/>	<input type="text"/>	<div><div>-----</div><div>✎ + ✖</div></div>	<input type="text"/>
<div>+ Add another Possible choice</div>				

Figure 39: The choices section

(c) Save the choice element and create more if needed.

## D.2 Message Presentation Elements

1. Go to the Message Presentation Elements sub menu.

## Petrichor Rain Service

### Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Users	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
VOICE SERVICE DEVELOPMENT		
Call Sessions		<a href="#">✎ Change</a>
Callers	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Choice Elements	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Input Elements	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Languages	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Message Presentation Elements	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
User Input Categorys	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
User inputs		<a href="#">✎ Change</a>
Voice Labels	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Voice Services	<a href="#">+ Add</a>	<a href="#">✎ Change</a>

Figure 40: The main menu

2. Click on the button 'Add Message Presentation Elements'

ADD MESSAGE PRESENTATION ELEMENT +

Figure 41: The message presentation button

3. Fill in the required fields.

Add Message Presentation Element

General	
Name:	<input type="text"/>
Description:	<input type="text"/>
Service:	<input type="text"/> <span>✎</span> <span>+</span> <small>The service to which this element belongs</small>
Is valid:	<span style="color: red;">●</span>
Validation errors:	No VoiceLabel in "Message: " Message does not have a redirect element and is not a final element
Voice label:	<input type="text"/> <span>✎</span> <span>+</span> <span>✕</span>
Message Presentation	
Redirect element:	<input type="text"/> <small>The element to redirect to after the message has been played.</small>
<input type="checkbox"/> This element will terminate the call	
<div> <span>Save and add another</span> <span>Save and continue editing</span> <span>SAVE</span> </div>	

Figure 42: The message presentation page

4. Save the message presentation element and create more if needed.

### D.3 User Input Categories

1. Go to the User Input Categories sub menu.

## Petrichor Rain Service

### Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Users	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
VOICE SERVICE DEVELOPMENT		
Call Sessions		<a href="#">✎ Change</a>
Callers	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Choice Elements	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Input Elements	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Languages	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Message Presentation Elements	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
User Input Categorys	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
User inputs		<a href="#">✎ Change</a>
Voice Labels	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Voice Services	<a href="#">+ Add</a>	<a href="#">✎ Change</a>

Figure 43: The main menu

- Click on the button 'Add User Input Category'.

ADD USER INPUT CATEGORY +

Figure 44: The add input category button

- Fill in the required fields.



Petrichor Rain Service

WELCOME, NATAS VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Voice Service Development › User Input Categories › Add User Input Category

Add User Input Category

Name:

Description:



Voice service:   

Figure 45: The input categories page

- (a) Fill in the name of the category.
- (b) Fill in for which voice service it is needed for.
4. Save the user input category and create more if needed.

## D.4 Input Elements

1. Go to the Input Elements sub menu.

## Petrichor Rain Service

### Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Users	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
VOICE SERVICE DEVELOPMENT		
Call Sessions		<a href="#">✎ Change</a>
Callers	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Choice Elements	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Input Elements	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Languages	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Message Presentation Elements	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
User Input Categorys	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
User inputs		<a href="#">✎ Change</a>
Voice Labels	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Voice Services	<a href="#">+ Add</a>	<a href="#">✎ Change</a>

Figure 46: The main menu

2. Click on the button 'Add Input Element'.

ADD INPUT ELEMENT +



Figure 47: The input element button

3. Fill in the required fields.

Add Input Element

Service:

-----






The service to which this element belongs

Name:

Description:




Voice label:

-----



Final voice label:




-----



The voice label that is played when the user has completed the input process. Example: "Thank you for your message! The message has been stored successfully."

Input category:


-----



The category under which the input will be stored in the system.

Redirect element:

-----



The element to redirect to after the message has been played.

Save and add another

Save and continue editing

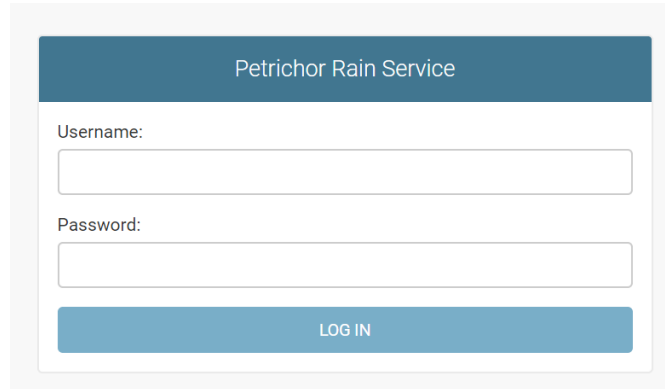
SAVE

Figure 48: The input elements page

- 4. Save the input element and create more if needed.

## E Appendix 5: A Manual for adding a new language

1. Log in into the application with an admin account.

The image shows a login form for 'Petrichor Rain Service'. It has a dark blue header with the service name. Below the header, there are two input fields: 'Username:' and 'Password:'. At the bottom of the form is a blue button labeled 'LOG IN'.

Petrichor Rain Service

Username:

Password:

LOG IN

Figure 49: The login page

2. Go to the Languages sub menu.

## Petrichor Rain Service

### Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Users	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
VOICE SERVICE DEVELOPMENT		
Call Sessions		<a href="#">✎ Change</a>
Callers	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Choice Elements	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Input Elements	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Languages	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Message Presentation Elements	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
User Input Categorys	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
User inputs		<a href="#">✎ Change</a>
Voice Labels	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Voice Services	<a href="#">+ Add</a>	<a href="#">✎ Change</a>

Figure 50: The main menu

3. Click the 'Add language' button.

ADD LANGUAGE +




Figure 51: The add language button




4. Fill in the required fields like name and code.




Add language

Name:

Code:

Language voice label:      
A Voice Label of the name of the language

Error message voice label:      
A general error message

Select language voice label:      
A message requesting the user to select a language

[Save and add another](#) [Save and continue editing](#) [SAVE](#)

Figure 52: Voice Service page

5. Create audio files for the Language voice label, Error message voice label and Select language voice label.
  - (a) , For example, use: [notevibes.com](https://notevibes.com) and create an account.
  - (b) Switch to WAV download.
  - (c) Fill in the text you want to turn to speech in the textbox.
  - (d) Choose voice and voice speed.
  - (e) Click convert.
  - (f) Listen to the file and when you are satisfied, click on Download WAV.
  - (g) Use a file converter to convert the WAV to the appropriate settings. For example [g711.org](https://g711.org).
  - (h) Select the source file.
  - (i) Choose 'Asterisk Standard' and Volume as Medium.
  - (j) Press submit and download the file.
6. Create the voice labels. This can be done by clicking on the plus button on the right side of the dropdown or via the Voice Label option in the main menu.

☒ **Voice service active**  
A voice service that is active is accessible to users. Marking this service as active (which is only possible when it is valid) will activate this service and disactiva

---

Is valid: ✔

---

Validation errors:

---

Supported languages:

English (en)

french (fr)

+

Hold down "Control", or "Command" on a Mac, to select more than one.

---

**Registration process**

**User registration:** preferred ▼

---

☐ **Register Language preference**  
The preferred language will be asked and stored during the user registration process

---

Figure 53: Add new language to voice service

7. Then save the language.
8. Go to the Voice Service submenu.

## Petrichor Rain Service

### Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Users	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
VOICE SERVICE DEVELOPMENT		
Call Sessions		<a href="#">✎ Change</a>
Callers	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Choice Elements	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Input Elements	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Languages	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Message Presentation Elements	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
User Input Categorys	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
User inputs		<a href="#">✎ Change</a>
Voice Labels	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Voice Services	<a href="#">+ Add</a>	<a href="#">✎ Change</a>

Figure 54: The main menu

9. Select the service for which the new language is installed.
10. In the 'Supported languages' box, add the newly added language.
11. If wanted, check the 'Register Language preference' checkbox underneath the User registration to store the user's language preference.
12. Save the voice service.
13. Determine which voice labels are used in the service and what they say.
14. Create new audio files for each voice label in the new language. (see step 5 for an explanation).
15. Go to the Voice Label submenu.
16. Find one of the voice labels that need changing and click on the name.
17. Then in the voice fragments section, add a new fragment.
  - (a) In the dropdown, select the new language.



- (b) Then select the created audio file.
- (c) Save the voice label.

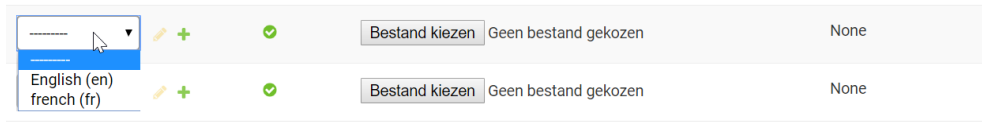


Figure 55: Add new fragment to voice label

- 18. Edit the rest of the voice labels as well.