Mali Seeds

Leon Bakker (2545184) Floris Gorter (2575145) Davey Quakkelaar (2575564) Robin van der Wiel (2573492)

C: Changelog

NOTE: The changelog of assignment 2 has been included in Appendix A.

Section	Change and rationale
Section 3	Fine-tuned to fit the current scope of the project.
Section 4	The script was revised to include the new call ending flow.
Section 5	The activity diagram was updated in order to reflect the final version of our application prototype.
Section 6	Added a UML class diagram that shows the information concepts of our application and added a rationale that explains how the information is connected.
Section 7	Added a diagram that sketches the technological infrastructure.
Section 8	Section added. We describe the cost considerations of our system.
Section 9	Section added. We explore the feasibility and sustainability of our system.
Section 10	Added two requirements that further narrow down future development of the application.
Section 11	The prototype description has been updated to match the final implementation. Front- and backend implementation results are described, as well as the data model.
Section 14	Short usage scenario updated to match the final prototype. Scenario now includes calls simulating both an advertising and searching farmer, to highlight the functional advertisement storage and retrieval.
Section 15	Incorporated feedback from assignment 1 and 2.
Section 16	Section added. We discuss the limitations and remaining improvements of the system.
Section 17	Section added. Unfortunately, we were not able to perform a user evaluation.
Section 18	A concluding section where we reflect on the project has been added
Prototype	KasaDaka has been extended to facilitate the functionality that we desire for our application. The prototype itself has been updated to use these new extensions, as well as to incorporate the received feedback. All of these changes are mentioned throughout this document, and mainly in sections 11 and 15.
Video	A video consisting of a small introduction and a demonstration of the system has been created.

0. Name

Mali Seeds: a seed advertisement system in Mali.

1. Summary of key idea

According to the AOPP, farmers in rural areas of Mali have to go through a lot of trouble to find other farmers to trade their seeds with, due to a lacking communication network, lack of internet connection, or any other facilities that allow them to advertise their products to other farmers for trade. In this report, we aim to present a voice-based application that allows farmers to report the products they want to trade (in terms of type and amount) with other farmers, and allows other farmers to find these advertisements. The system is intended to improve the food security among the people, especially in rural areas.

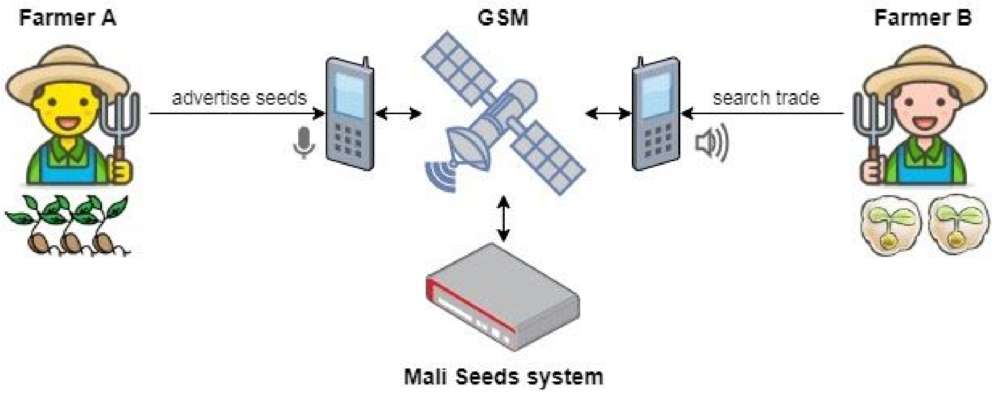
The application would be usable via the available mobile phones and mobile phone network due to the prevalent usage of mobile phones in these types of areas, ensuring that the inhabitants of rural areas also are able to utilise the system without any other available communication facilities. Because of the application, and by the use of this application, farmers would be able to easily advertise their products and find potential farmers to trade with, and farmers would also be able to find any advertisements of the products they desire.

2. Actors and goals

Stakeholder	**Operational goal**	**Responsibility in the envisaged system**
Farmers (advertising)	Advertising seeds to trade.	Calling to the system to place an advertisement for the type and amount of seeds they are willing to trade.
Farmers (searching)	Searching for a specific type of seed and trading.	Calling to the system to hear the currently advertised seeds, then contacting the advertising farmer to propose a trade.

3. Context and scope

a. Make a sketch of the interactions between the parties involved in the scenario



b. Who are the (external) stakeholders to the use case and what are their concerns?

- Farmers, who's main concern is being able to communicate with others to trade products, to ensure their own food security.
- Project creators, who's main concern is a successful deployment of the system.
- Project funders, who's main concern is a success story.
- Local operators, who's main concern is the continued availability of the system, maintenance, and technical support for the village. Additionally, the local operators may have to (indirectly) interact with us as developers of the system, if necessary.
- Other villages, who's main concern is their interest in also obtaining the system after seeing the successful integration of the system in another village, and examining the positive impact on food security in said village.

c. What is the scope of the scenario?
Currently, the scope of one instance of the system is a local community (one village). The system would act as a local advertising board on which the inhabitants of said village/community can advertise their products up for trade, or can look for products that they wish to trade for. Furthermore, the system is only meant for connecting farmers with each other, not for further facilitating or helping in the negotiations of the trade.

d. What are success or performance measures for the scenario?
One such performance measure is the ability to post an advertisement for your product, and if you were to call again seeking for this product, and your advertisement would be played back, the system has worked successfully.

Next, the speed and success rate (i.e., they successfully post an advertisement according to their wishes) with which a farmer can post their advertisement would be a performance measure. This speed would be evaluated by examining how long it takes a farmer to navigate through the prompted menus.

Finally, the deployment success would be measured in terms of the number of people that post an advertisement, and the number of people that search for an applicable advertisement in the system.

e. What are important (pre)conditions that must be or are assumed to be satisfied for the scenario?

- Farmers should have a functional mobile phone.
- A mobile phone (GSM) network must be in place.
- Farmers must be willing to call the advertising board in order to stay aware of the latest advertisements.
- Farmers require knowledge about numbers to effectively interact with the system.
- Farmers should be able to understand how the system works. Ideally, the system is designed such that no interpretation/functionality issues arise.
- The system must be up and running, i.e., it must be possible to post trade offers onto the advertising board, and it must be possible to receive these in return when the system is prompted to.
- The system must be able to communicate in (at least) the language of the village where it is deployed.

4. Use case scenario script

In this scenario, we make a distinction between farmers that are advertising seeds and farmers that are searching for seeds. In our scenario, the advertising farmer (referred to as Farmer A) wishes to trade 3 bags of rice seeds for another product. The searching farmer (referred to as Farmer S) is looking to obtain one bag of rice seeds. During all of the steps we assume that the users listen to the automated voice instructions prior to the steps.

1. Farmer A calls the telephone number of the *Mali Seeds* system and automatically enters the selection menu.
2. Farmer A presses the ‘1’ key to select French as language.
3. Farmer A presses the ‘1’ key to select the ‘new advertisement’ option.
4. Farmer A presses the ‘4’ key to select ‘rice’ as the product.

5. Farmer A speaks the amount of bags to be advertised.
6. Farmer A presses the ‘1’ key to confirm the advertisement.
7. Farmer A presses the ‘3’ key to not continue advertising or searching.
8. The *Mali Seeds* system disconnects the call with Farmer A.

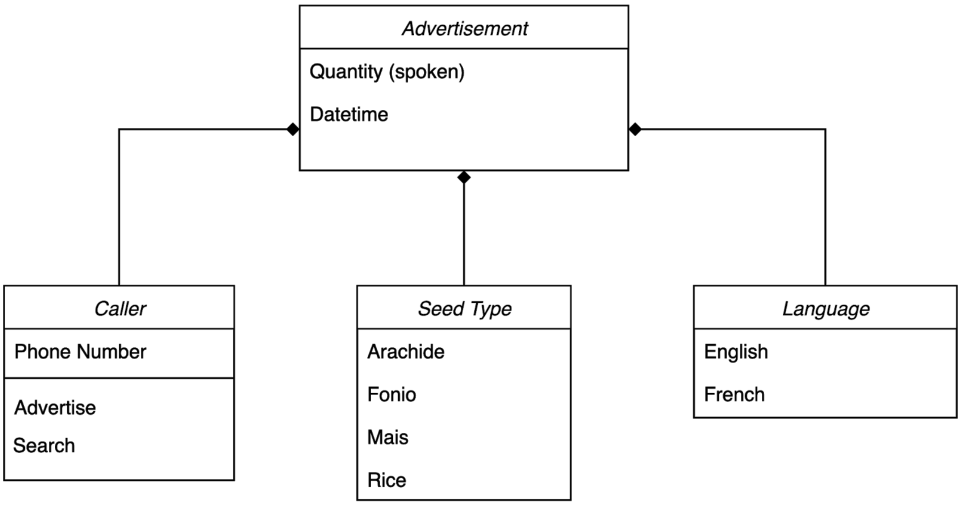
9. Farmer S calls the telephone number of the *Mali Seeds* system and automatically enters the selection menu.
10. Farmer S presses the ‘1’ key to select French as language.
11. Farmer S presses the ‘2’ key to select the ‘searching for seeds’ option.
12. Farmer S presses the ‘4’ key to select ‘rice’ as the product.
13. Farmer S hears an enumeration of all the advertisements regarding rice as the product.
14. Farmer S presses the ‘1’ key to indicate interest in the first advertisement.
15. Farmer S hears the phone number of Farmer A.
16. Farmer S presses the ‘3’ key to not continue advertising or searching.
17. The *Mali Seeds* system disconnects the call with Farmer S.

5. Interaction and communication

The activity diagram shown in Appendix B as Figure 1 represents a generalised execution of the final version of our application from both advertising and searching perspectives of the farmers. The use case scenario script in Section 4 gives an example of an execution, which conforms to the activity diagram. Figure 2 displays the activity diagram from our previous prototype and Figure 3 displays the activity diagram of our original first prototype.

6. Information concepts

The UML diagram shown below shows the most important information concepts of our application. The data that is stored is very straightforward: the person calling the application is the *caller*, and they are associated with a phone number. A caller chooses the action “advertise” or “search”. Upon advertising, an advertisement is created which is composed of the caller, a seed type (one of the four types supported by our application), a language in which the amount of seeds is spoken, and the date and time.

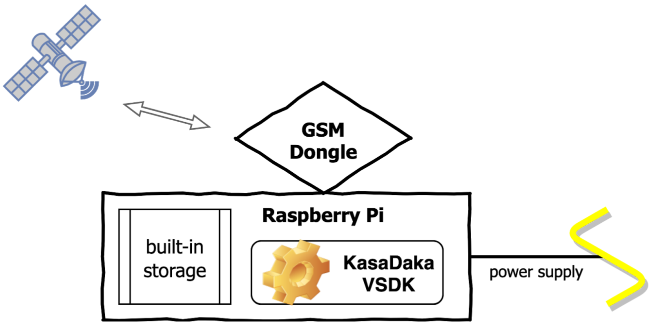


7. Technology infrastructure

Our voice service is designed using the KasaDaka VSDK, which allows us to deploy the application on a Raspberry Pi. This type of hardware uses very little resources (it only needs a power supply) and is reasonably resilient against power outages when running.

The technology infrastructure needed for our application is very limited. The application can be reached via basic phones that can make telephone calls. In order for the application to be reachable, it must be in an area that is covered by a GSM network. The system does not need an active internet connection, as advertisements are stored locally on the KasaDaka system. Automatically scheduled tasks take care of removing old advertisements, and no human interaction is needed for this.

The Mali Seeds voice service software can come pre-installed with the hardware upon deployment, future updates may involve the necessity of an internet connection. If a local administrator is appointed to manage and ensure the availability of the system, an additional computer is required to interact with the KasaDaka platform. Instructions for the administrator could be supplied through e-mail and offline manuals. We point out that the Mali Seeds system by itself can run independently without requiring an Internet connection, however the interaction surrounding the system may depend on a communication method such as the Internet.



8. Cost considerations

The estimated costs that are necessary for the application to be fully functional are very limited. The table below shows an estimation of the operational costs of the application.

The costs could be carried by an organization such as the AOPP, as the application would stimulate the different seeds that farmers grow, which could also be beneficial towards the formal seed value chain. Alternatively, the project could be funded by means of a donation or designated funding. There are no monthly costs for hiring employees to manage the system, as it is self-managed. In the (rare) case that the system malfunctions, a representative from the AOPP should be able to repair the malfunction with our instructions.

Component	Type of cost	Estimated cost
Raspberry Pi + SD card	One-time	€ 40
GSM Dongle	One-time	€ 40
SIM card + phone number	One-time	€ 1,50[1]
Power supply (est. 1Watt = 1€ for a year)	Annually	€ 4

[1] https://prepaid-data-sim-card.fandom.com/wiki/Mali

9. Feasibility and Sustainability

a. What is the technical feasibility of the scenario?

There exist a number of risks that need to be assessed in terms of the technical feasibility of the system. These include the following, displayed and judged on their probability and impact (i.e. qualitative risk assessment) in the table below.

Risk	Probability	Impact
Power outage	Very likely	Requires a reset/reboot, system temporarily unavailable.
Hardware overheating due to hot weather conditions at deployment location	Very likely	Hardware has to cool down. System temporarily unavailable. Investigate cooling solutions.
System malfunction / Software update required	Likely	Requires a reset, and possibly (software) maintenance from an external factor.
Internal (software) crash	Unlikely	Requires a reset/reboot, and possibly (software) maintenance from an external factor.
Storage capacity runs out	Unlikely	Delete the oldest advertisements that have not yet reached their deletion age.
GSM network temporarily unavailable	Unlikely	System temporarily unavailable.

Next to the observed risks, there also exist a number of technical obstacles to overcome. Firstly, one technical obstacle is the notion that the system must first be deployed and made functional in the target environment (e.g. a village or community). This requires the presence of (trained) individuals capable of deploying the technology, and instructing the target users of its purpose, its workings, and how it is to be used by them (and the potential benefits they could enjoy if they used it). Moreover, these individuals should be able to travel to these (often remote) communities, and in the case that no other individuals could be trained on-site, would have to travel there often more than once to perform maintenance.

Secondly, storage capacity should be managed as well. This is done automatically by the system. Regular clean-up tasks make sure that old advertisements are deleted by the system. Should it happen that the storage capacity runs out while all old advertisements have been deleted, then a number of the oldest advertisements (that have not yet reached their deletion age) should be deleted anyway. This scenario is very unlikely, as the advertisements take up very little memory and the application is deployed per (small) community, thus the number of advertisements will not be extremely high.

Thirdly, the KasaDaka VSDK is not very well documented, which increases the difficulty that goes along with the required updating/maintaining of the voice service in the long run. Together with the absence of the knowledge about this framework and the programming skills required to facilitate the updates in the local community, providing services or increasing the usability of the system would become difficult.

b. What is the business and economic feasibility and sustainability of the scenario?

Firstly, the system must be supported by the local community, since without its usage, the system would not be able to survive or be useful in the first place. Because the system is aimed at improving the food security of the local community in which it has been deployed, it would certainly positively impact the economical situation in this community, which in turn is a reason the users would resort to utilising the system. Because trading seeds between farmers becomes easier, farmers will be more inclined to grow new crops, and will be able to profit from these crops (either by selling the produce, or by saving money by using the produce themselves). Therefore, we think that the system will be used by the farmers, and thus will be kept online by its use there.

Secondly, the system itself is reasonably simple to keep online. The KasaDaka VSDK was built for the Raspberry Pi hardware, which is commonly available and easily usable in most situations. This allows for the local community, the AOPP, or a different organisation, to acquire the system without spending too much. Therefore, the system is economically viable, since it improves food security at a small cost.

c. What are possible goal conflicts and dependencies in the collaboration between the actors in the scenario?

Since both types of actors involved in the system (i.e., the farmers that are either searching for a trade, or offering up their seeds for trade) have to deal with each other in trading, their goal is the same. Both parties would profit from their trades, resulting in no direct goal conflicts between these actors. The system does depend on both actors being willing to trade, meaning that if a certain set of farmers refuses to trade, the system will not be useful to them (or other actors that might be potential traders with these farmers) in any shape or form. Furthermore, there must be advertisements to find in order for the system to operate. If farmers are only willing to search for trades instead of offering them, the system will not work effectively. Therefore, there is a search dependency between the actors.

Another dependency is that the actors must be in possession of a mobile phone to operate the system. It is possible that some members of the community do not own a mobile phone, and would rely on other actors that do own a mobile phone to make calls for them.

d. Are there important general preconditions for the scenario to work, and is it sufficiently interoperable with the wider context both in a business process and a technical sense?

The system requires farmers to have mobile phones, and requires the traders in a community to be willing to use the system for their trades. Without the explicit input from the farmers, the system will not work for both parties. However, since the system itself is aimed at improving the food security of the farmers, the farmers are very likely to attempt to use the system to increase the number of trades they make. We also believe that the system is relatively easy to use. Because of this, the system is likely to be sufficiently interoperable with the wider context in the existing business process.

A list of other preconditions that are important for the scenario to work was given in section 3e. These preconditions remain important in order for the scenario to be as successful as possible.

10. Key requirements (MoSCoW)

Requirement	M	S	C	W
1. Feature to enter information concerning an advertisement (i.e. type, amount of certain product).	X			
2. The possibility to search for seeds and hear the present advertisements in the system.	X			
3. A clear, short and structured DTMF selection menu and call flow.	X			
4. Support for multiple concurrent callers.		X		
5. Familiar voices performing the speech part in a local language.		X		
6. Support for multiple (local) languages.		X		
7. Functionality to let the owner of an advertisement delete their ad.		X		
8. PIN input at the start of the call, such that the service is not just accessible to any caller.			X	
9. Automatic call forwarding after selecting a specific advertisement.			X	
10. A service for helping farmers make the trade itself. They are only assisted in finding one another to ensure trading opportunities.				X

The idea of the application is to let farmers initiate the trade themselves. In order to keep the application simple, it is up to the farmers to discuss the details of a trade. As of right now, it is not possible for farmers to delete their own ad, in order to not complicate the system.

For now, it does not seem possible to support concurrent calls to the application. This would require a more complicated setup with additional SIM-cards that are all connected to the application.

11. Prototype description

To suit the region of potential deployment, the prototype has pre-recorded messages in French and English. For deployments in villages or areas using other languages, new audio files will have to be recorded.

The final prototype is able to guide users through setting up an advertisement which will result in the advertisement being stored in the database. Additionally, in contrast to the previous prototype, users can hear the stored advertisements that are present in the system. When searching for a specific type of seed, only the advertisements that were created using the same language as the current caller will be available to select from. After selecting an advertisement, the system will read out the phone number attached to this advertisement.

a. Description of the design decisions of your application.

We have actively made decisions about which user inputs should be entered using DTMF input and which ones should be user audio recordings. For the amount of bags a user wishes to advertise, the user is required to *speak* the quantity. This avoids users having to be able to understand how to write (potentially) large numerical amounts.

On the other hand, the desired seed type for an advertisement or search is not spoken but chosen by means of DTMF. This makes it significantly easier for the system to filter and match advertisements to searches, as we do not require any audio interpretation or speech-to-text.

b. Describe front- and back-end implementation results.

Front-end: We provide a simple and clear call flow through the selection menu. We aim to ensure an easy to understand enumeration of the available advertisements, followed by a slow and clear audio presentation of a phone number. We insert a significant delay between the system speaking each digit of the phone number, allowing for enough time for the user to process the information.

To improve the clarity of the system we add a “bags of <type>” audio fragment to the choice selection menu that lists the advertisements. The DTMF advertisement choice option is then structured as follows: [To select <user recording of quantity> *“bags of rice”* press 1]. This way the users only have to speak a numerical amount of bags. All further audio is pre-recorded and loaded by the system, reducing the risk of unclear recordings.

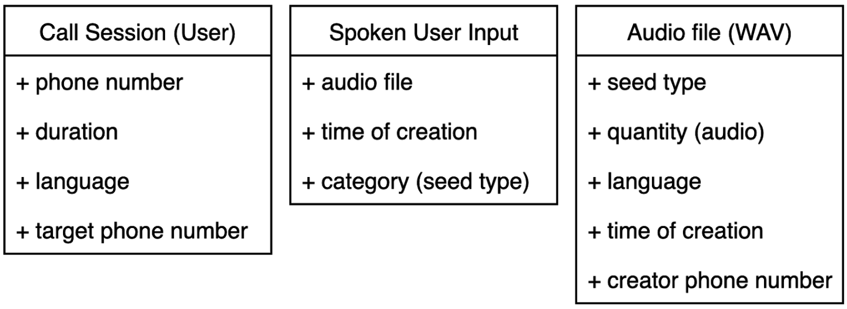
Back-end: We implement choice recollection and database support to accommodate for the storage of advertisements. We adapt multiple views and templates to allow for retrieving previously stored advertisements. More specifically, we:

1. Adapt the choice selection element to be able to list stored user recordings. Only the relevant advertisements for a search request will be selected. We select all the advertisements from the database where the user input category matches the selected seed type, and then make sure the search language matches the advertisement. This extension allows for us to list exactly as many DTMF input options as there are advertisements present in the system that match the search request.

2. Adapt the choice selection element to store the phone number attached with the selected advertisement, such that the phone number remains known in the next elements. A phone number is attached to an advertisement, so upon selection of an advertisement when searching for seeds, the phone number attached to this advertisement will get stored with the current call session in the database. Afterwards, a message presentation element can retrieve this phone number again.

3. Adapt the message presentation element to be able to read out the stored phone number corresponding to the selected advertisement. The audio for the phone number is generated using the stored WAV files for digits, in the appropriate language.

c. Data model.



The diagram above displays the data model of our application. We distinguish between three different main data elements: call sessions, spoken user inputs and audio files. Note that the spoken user input data contains a reference to the audio file. An 'advertisement' within the system can be viewed as the combination of a Spoken User Input instance and the associated audio file.

The information that is part of the audio file is stored as part of the file name. Storing this information inside the actual database is more elegant and suitable, for example as a part of the spoken user input table. However, due to time constraints, we consider the current data model sufficient. Furthermore, storing the target phone number (i.e. the phone number of the selected advertisement) as part of the call session is not the most appropriate location either, whereas a table dedicated to 'searches' would be more fitting for this data. However, storing the number in the call session table is convenient for easy access throughout the system. As a consequence of this decision, the only way to track users searching for seeds is inferring this knowledge from whether a call session contains a target phone number or not.

After setting up and launching the system, all of the data in the system becomes user-provided. The system does not require any other external data to serve its purpose as a marketplace. The application makes use of the (internally) stored user data to function as intended. That is, the local advertisement data is used whenever a user searches for seeds.

The data displayed in the data model could be shared as open data for others to research and base future projects on. If the data becomes openly available, we would have to consider the privacy of the users. Respecting privacy concerns may require measures such as phone number anonymization, and omitting user audio recordings from the data.

12. Pointer to the application code

https://github.com/lv-bakker/KasaDaka-VSDK

13. Pointer to how to access the application

http://mighty-bastion-29544.herokuapp.com/vxml/start/3

14. Short Usage scenario

When calling the application, the user will initially select the desired language of the system, for example English. First, in order to simulate an advertising farmer, the user selects the option that they are an advertiser (key 1). Secondly, the user is asked to select the seed they wish to advertise for (keys 1-4 for arachide, fonio, mais, and rice respectively). After this, the user is asked how many bags of the seed they have to offer, and after speaking this, has to confirm their advertisement (key 1). The advertisement is then stored in the database. The user decides how to continue the call: create (another) advertisement, search for seeds, or end the call.

After storing an advertisement, the "searching for seeds" branch will contain the newly created advertisement for users to hear. To hear it, call the application with the same language, select the searching branch (key 2), select the seed type that the advertisement was created for (key 1-4), and the advertisement created earlier will be played, along with other advertisements in the system. Select the advertisement by choosing its assigned key (depending on the enumeration), and the phone number attached to this advertisement will be spoken by the system (slow enough to allow the caller to write it down).

15. Feedback

We have received four questions related to our presentation, of which the first two are concerned with our activity diagram:
1. Group 11 (Dominique): "*When searching for seeds, an enumeration of advertisements is given*. What if there are like, 10 advertisements? Will they all be given? [...]"
2. Group 10 (David): "*Caller hears phone number of advertisement, then disconnects*. Usability question: can callers really remember the number? what if they want to hear it multiple times?"
3. Group 11 (Dominique): "Which languages does it support now? Which languages will it support after expanding it?"
4. Wilhelm Kutah: "With seeds, quality is key, is verifying the quality of the seed outside the scope of your work?"

For question **1**, we intend on enumerating advertisements in a chronological order based on the date and time the advertisement was created. We are still evaluating how to handle more than 9 advertisements. Question **2** has been partially answered throughout this document, we will attempt to add an automatic call redirect option to the system and also consider the option of repeating the spoken phone number. For question **3**, at the time of the presentation our system only supported French, whereas now we have also added English. KasaDaka makes adding more subsequent languages very easy. Lastly, regarding Wilhelm Kutah's question **4**, seed quality is not part of our intended use case, this is an affair the farmers must resolve among themselves.

Assignment feedback

After receiving some feedback regarding our technological implementation and documentation of the prototype of assignment 2, we decided to address the provided suggestions as much as possible. This resulted in the following changes:

A manually inserted beep has been added for the recording element, as the VXML beep that was built into KasaDaka does not appear to function. Additionally, the call flow near the end of the system has been adjusted to incorporate the feedback concerned with being able to return to the main menu after a run through the application.

To address the received feedback from a real world perspective on the language of the system, for now, the application works with a language selection menu. In an actual deployment of the system, different phone numbers could be set up for different instances of the application, where one phone number corresponds to a specific language.

16. Discussion of Scope and Fidelity

The largest portion of the proposed functionality for the application has been successfully implemented. However, there are still some limitations to the functionality of the system. Additionally, there are a couple of improvements that should be addressed before the system can be considered *fully functional*.

At this moment, the prototype does not remove advertisements periodically. This is a feature that can easily be implemented on a Raspberry Pi, for example by periodically running a (bash) script that checks the date and time of each advertisement. Advertisements that have a certain age, for example 7 days, could then be deleted in one batch operation. KasaDaka could be extended to perform this maintenance task.

The application does not provide an explicit option to repeat the phone number after the caller has selected a specific advertisement. This is a feature that can easily be implemented, but does not seem necessary at the time as the phone number is spoken very slowly. Currently, the phone number can be heard again by simply repeating the search.

A feature we wanted to implement was call forwarding, in order to save the caller the hassle of writing down the phone number of a specific advertisement. There is a command in VXML (*transfer*) to initiate a call forwarding, however this resulted in an error when using it in KasaDaka. Unfortunately, André and Francis were not aware of any existing workaround for this.

Our prototype only supports nine different seed types at this point, because we use simple DTMF input. This also means that only nine advertisements can be present for a specific seed type and language combination. This could be extended in the future by using a larger input range (e.g. 1-99), in order to support more seed types and to allow more advertisements per seed type. However, this would possibly introduce difficulties with illiteracy and large numbers. Additionally, listening to a very long enumeration of seed types or advertisements can be demotivating for potential users.

Concerning the scope of our project, the application could be extended by adding support for local languages. It is a reasonably trivial task, however because of time limitations we chose not to incorporate more languages into our final prototype. For a successful deployment in a target village, we would first have to cooperate with the local community to record the necessary audio fragments.

17. User evaluation

We did not perform a small-scale user evaluation, however we did frequently test the application ourselves in order to fix bugs and inspect its functionality. Unfortunately, we were not able to receive input from interested parties in Mali, due to the lockdown situation.

18. Conclusions

In conclusion, we are proud to introduce a functional marketplace system that manages to record, store, and present advertisements of different seed types in multiple languages. We acknowledge that there are a couple of remaining attention points that should be tackled, before we can consider an actual deployment of the system. We would have liked to interact with the interested stakeholders in Mali, such that we could incorporate their views and knowledge to further improve the usability and functionality of our system.

Throughout the project, we have learned various aspects related to local contexts and the development for rural areas. We believe we have successfully applied this new knowledge to deliver an application that takes the limitations of rural areas into account. At first, we experienced working in an unknown and undocumented development environment as difficult, however after the unfamiliarity fades, we manage to effectively use and extend the KasaDaka VSDK.

We hope that our work may inspire others to also contribute to the development of similar applications, projects and initiatives. We believe that with the right mindset and effective cooperation, the ICT skills and knowledge that come with our field can have a positive impact on the world. Especially in challenging times such as these, with COVID-19 on everyone’s mind, it becomes apparent that similar development efforts can provide the less fortunate with the information they need and deserve.

Appendix A: Changelog of assignment 2

The overall scope of our application has been increased by adding a second language for users to choose from. We have added functionality to the KasaDaka platform to match the already reasonably large functionality of our Assignment 1 prototype.

Our geographical scope was already limited to a local community (e.g. one village), originating from Assignment 1. In combination with no explicit feedback on Assignment 1, we do not further adjust the geographical scope.

We have prepared further topological and overall scope expansions for Assignment 3, where we will explore the possibilities of extending KasaDaka in combination with the available features that VXML offers. These expansions are mentioned throughout this document, and include concepts such as call redirecting and advertisement expiration.

Section	Change and rationale
Prototype	Ported our prototype to KasaDaka. We make use of all of the available elements in KasaDaka to implement our prototype: choices, languages, message presentations, and spoken inputs. We experience the KasaDaka VSDK as easy to use, but limited in functionality. The functionality of our system changes slightly, however these changes are mainly in the backend, whereas the user interaction remains almost identical.
Prototype	Added English, next to French, as a language in order to increase the scope of our application. Other languages (e.g. Bambara) can easily be added with the necessary WAV files.
Prototype	Changed default language voice clips to match our custom voice clips (same voice), to ensure consistency and to not confuse callers.
Prototype	Replaced multi-digit (DTMF) input of *number of bags* with a user recording to account for KasaDaka limitations and the local context with limited knowledge of numbers. By default, KasaDaka does not support multi-digit input.
Prototype	We no longer use a MySQL database to store advertisements due to KasaDaka not offering an easy way to interact with such a database natively. Instead, we store advertisements as WAV files in our file storage.
Prototype	Modified/extended KasaDaka to store necessary info regarding the advertisements that are now represented as WAV audio files. The audio file name contains the seed type, caller phone number and a timestamp, while the actual audio content is the spoken amount of bags. To realize this functionality, we implemented a choice recollection method such that the selection of the seed type choice element is known when recording the amount of bags. We retrieve the caller ID and the start time of the call from the *CallSession* object in the *record* view, before storing the WAV file.
Prototype	Dummy functions/elements are in place for the “searching for advertisements” branch. We have implemented storage of the advertisements, but the search branch cannot deal with retrieving the advertisements yet.
Section 4	The use case scenario script was updated to match our new prototype.
Section 5	The activity diagram was updated to match our new prototype.
Section 7	Updated the technological infrastructure with focus on how the application relates to low-resource hardware and the local context.
Section 11	Adapted prototype description to match current state of the prototype and to announce plans for extending the prototype.
Section 14	Updated short usage scenario to match our new prototype.
Section 15	Added and answered feedback/questions on our *presentation*.

Appendix B: Diagrams

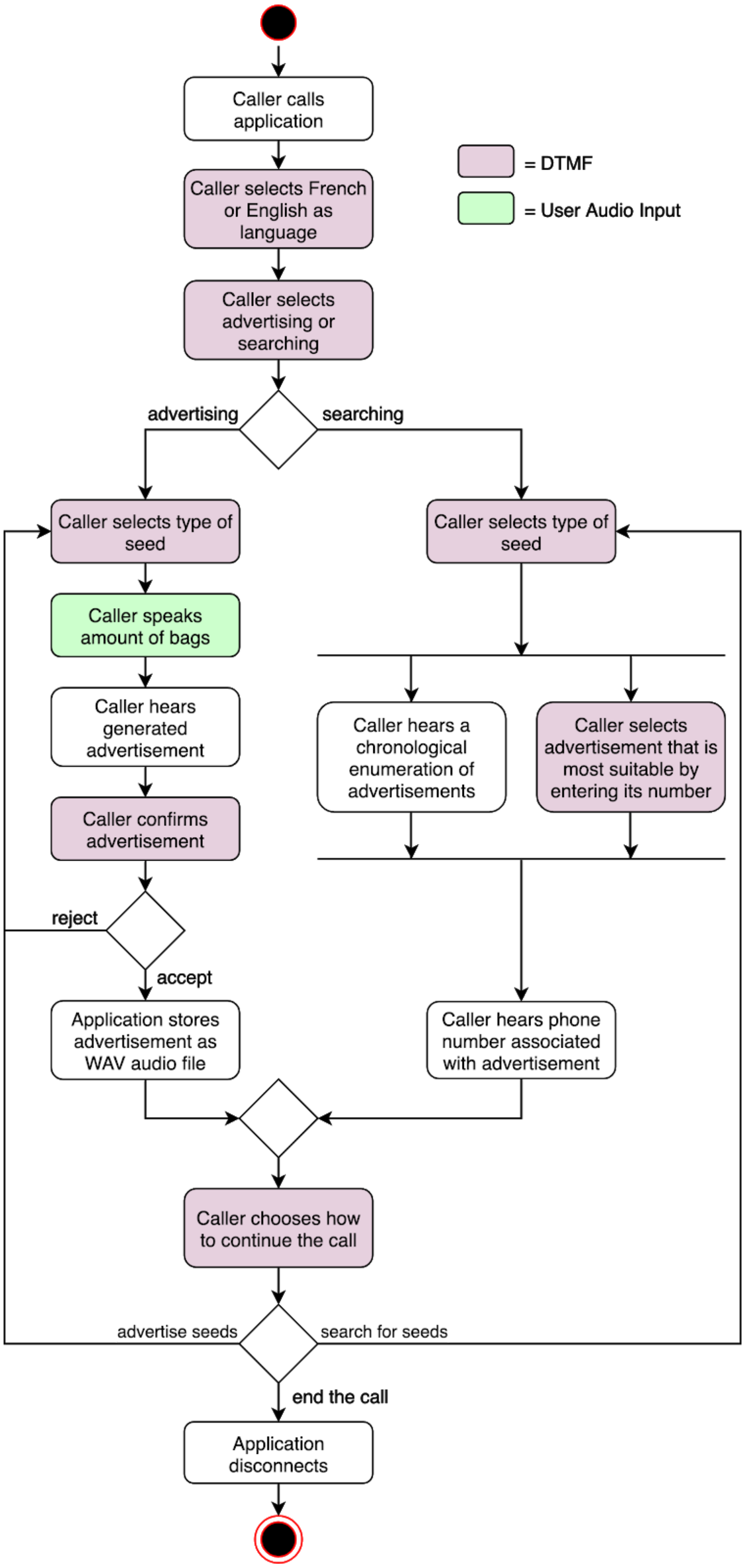

Caller calls application
Caller selects French or English as language
Caller selects advertising or searching
= DTMF
= User Audio Input
advertising
searching
Caller selects type of seed
Caller speaks amount of bags
Caller hears generated advertisement
Caller confirms advertisement
reject
accept
Application stores advertisement as WAV audio file
Caller selects type of seed
Caller hears a chronological enumeration of advertisements
Caller selects advertisement that is most suitable by entering its number
Caller hears phone number associated with advertisement
Caller chooses how to continue the call
advertise seeds
search for seeds
end the call
Application disconnects

Figure 1: Final activity diagram of the application

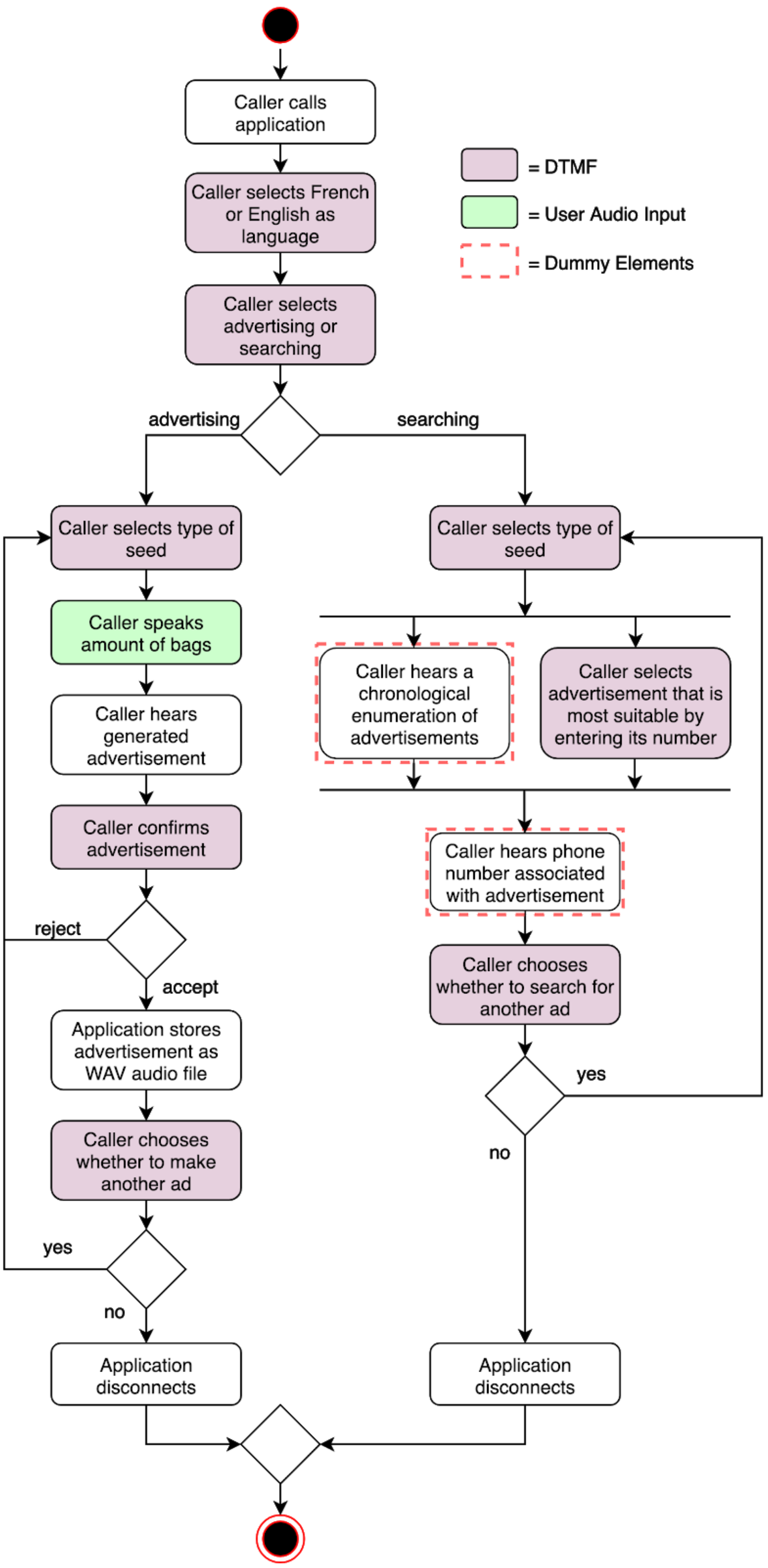

Caller calls application
Caller selects French or English as language
Caller selects advertising or searching
= DTMF
= User Audio Input
= Dummy Elements
advertising
searching
Caller selects type of seed
Caller speaks amount of bags
Caller hears generated advertisement
Caller confirms advertisement
reject
accept
Application stores advertisement as WAV audio file
Caller chooses whether to make another ad
yes
no
Application disconnects
Caller selects type of seed
Caller hears a chronological enumeration of advertisements
Caller selects advertisement that is most suitable by entering its number
Caller hears phone number associated with advertisement
Caller chooses whether to search for another ad
yes
no
Application disconnects

Figure 2: Second version of the activity diagram of the application (deprecated)

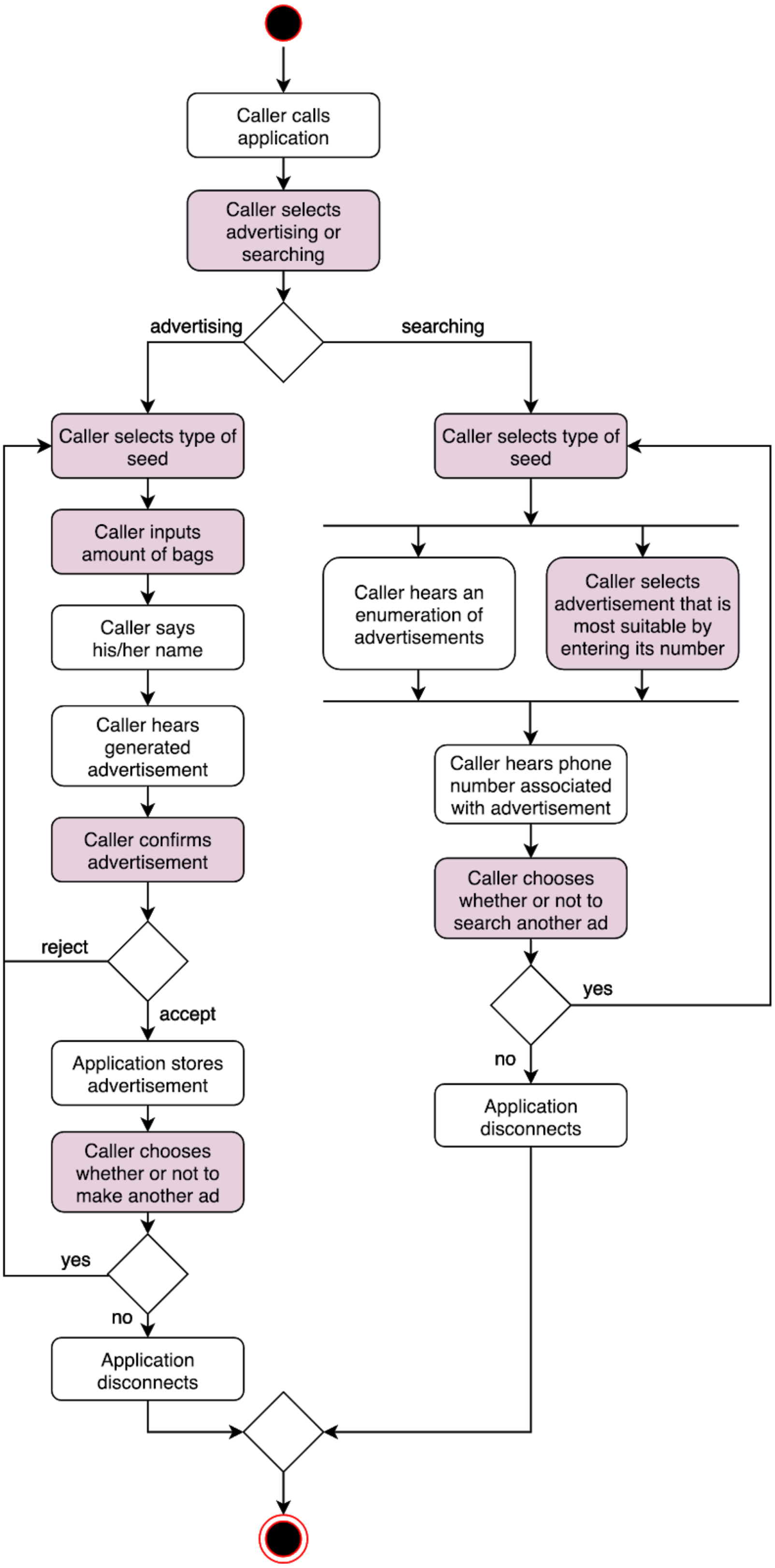

Caller calls application
Caller selects advertising or searching
advertising
searching
Caller selects type of seed
Caller inputs amount of bags
Caller says his/her name
Caller hears generated advertisement
Caller confirms advertisement
reject
accept
Application stores advertisement
Caller chooses whether or not to make another ad
yes
no
Application disconnects
Caller selects type of seed
Caller hears an enumeration of advertisements
Caller selects advertisement that is most suitable by entering its number
Caller hears phone number associated with advertisement
Caller chooses whether or not to search another ad
yes
no
Application disconnects

Figure 3: Activity diagram of the first prototype (deprecated)