

# ICT4D-2020

Requirements Document Foroba Blon Simple Poll

Students:

Tim Verlaan (11669128) tim.verlaan@student.uva.nl

Sven Preng (2614131) s.preng@student.vu.nl

Wouter Kok (2639768) w.j.kok@student.vu.nl

Pjotr Scholtze (2612369) pjotrscholtze1@gmail.com

Markus Funke (2644722) m.t.funke@student.vu.nl

Last Update: 20.05.2020

# Changelog

Date	Version	Changes	
2020-04-10	V0.1	Initial version of the document	
2020-04-30	v0.2	<ul> <li>Edit Ch. 7 Added relation to low-resource hardware/Kasadaka</li> <li>Edit Ch. 8 Cost consideration (add project runtime costs)</li> <li>Edit Ch. 11 Prototype description</li> <li>Edit Ch. 13 Pointer to how to access the application</li> <li>Edit Ch. 15 Feedback questions</li> </ul>	
2020-05-01	v0.3	Finalizing the document before handing it in	
2020-05-12	v0.4	Updated document to fit criteria: line spacing 1.0, Times 11 pt	
2020-05-14	v0.5	<ul> <li>Added Ch. 0</li> <li>Edit Ch. 1 Updated scope with focusing to closed questions</li> <li>Edit Ch. 2 Added more information</li> <li>Edit Ch. 3 Added questions and more information</li> </ul>	
2020-05-19	v0.6	<ul> <li>Edit Ch. 1 Further update scope and added image</li> <li>Edit Ch. 2 Further update goals and responsibilities.</li> <li>Edit Ch. 3 Updated network configuration diagram</li> </ul>	
2020-05-20	v1.0	<ul> <li>Edit Ch. 5 Updated activity and use case diagrams</li> <li>Edit Ch. 6 Updated the information concept and UML diagram</li> <li>Edit Ch. 11 Prototype description</li> <li>Edit Ch. 13, 15, 16 Add text to chapters</li> </ul>	

## 0. Name

FB-SPS: Foroba Blon Simple Poll System

## 1. Summary of key idea

Foroba Blon was designed to make the radio stations more interactive. People from remote villages are able to participate in news generation and distribution. The goal we've set for our project is to further increase the interactivity of community radio stations. Therefore we aim to produce a new type of system based on the existing Foroba Blon system that can be integrated by radio hosts. The idea for the system is based on experiences in Goma, DRC during the last presidential elections. One of the group members experienced native co-workers receiving many rumors through mobile and radio networks without having the possibility to fact-check them. In response to this need, this project aims to deliver a feature that takes in a question or statement from an individual (or group) and allows this question to be answered (with yes/no answers) by the broader listening community. In this sense, the **question** works very much like

**polls** that are popular on internet platforms such as Instagram and Facebook. Polls can have fixed dichotomous answers such as yes/no agree/disagree or verify/falsify: which allows them to be able to assess rumors/statements.

Calling the system might be too expensive for listeners; that's why we thought of adding an option for a listener to simply call the non-responding system, and being called back by the system so the charges for the phone call will be for the callee instead of the caller. But after some discussion with a teacher, we decided to discard the idea because the implementation will demand too much time. Therefore we let the users call the system and start the connection. The radio host can set up the question in an administration page of the web interface. Furthermore, it is possible to see the citizen's answers on a results page. In this way the radio host is able to interact with the citizens by talking about the results of a poll, and even so the citizens have an opportunity to let their voices be heard.

Below is a visualization of our new logo<sup>1</sup>. The same style and color scheme is used to create a new logo for our solution, which is shown on the right side.



# 2. Actors and goals

The table below gives the actors their goals and responsibilities in connection to the system.

Stakeholder	Operational Goal	Responsibility in the envisaged system
Radio Host	Increase the interactivity of radio stations, increase transparency of information and also add a fun new element to the radio-listening experience.	Update poll questions by using the system's interface. In addition, the poll information (including the poll question and results) should be spread over the radio and the system should be checked if it's working accordingly. The radio host needs to have some technical literacy to be able to work with the system.

<sup>&</sup>lt;sup>1</sup> The speech bubbles and phone icons from our logo are made by Flat Icons from: <u>https://www.flaticon.com/</u>

Citizens/ Listener	Give their opinion on polls or questions by using their cell phone to call the system.	Let their opinion be collected by calling the system and pressing the answer they have for the poll.
-----------------------	--	---

# 3. Context and scope

a. Make a sketch or diagram of the layout or network configuration of the interactions between the parties involved in the scenario.

A sketch of the layout of the interactions between the parties involved in the scenario is given below. Citizens can call our system with a connected mobile phone. Through the call, a citizen can give a question to the poll. The radio host is then able to interact with the system through a web interface in which the question of the day can be updated and the poll results can be seen.

The database is used to save polls and voice messages from citizens and the FTP Server is used to serve WAV files to the web-interface.



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

The stakeholders with their concerns are shown in the table below.

Stakeholder	Concerns
Citizen	Is the system easy to use? How will the phone interaction be? What will the costs be for mobile phone services? Is my phone number being saved?
Radio Host	How can I analyze the answers? How can I set-up a new poll? What will the costs be for me? Will citizens enjoy this new functionality?
Funding Agency	How can we successfully plan the project? What will the costs be for the project? What will the risks be? How many people do we need? How can we maintain the system? At which radio station will we test the system?
Development Team (VU)	How will we implement the system in an easily operable way?

c. What is the scope of the scenario (especially, what is outside it, not considered, system boundary)?

The scope should be small and controlled and the system should first be used in a non-political context with a group that is familiar with Foroba Blon and the goals of the project. Otherwise, the system could be unintentionally used for spreading propaganda, white-washing of lies, decreased transparency and polarisation. It is advised to enroll this system first with the initial four radio stations of the Foroba Blon project.

*d.* What are success or performance measures for the scenario (especially in relation to what a pilot demonstration should be able to show)?

The success and performance measures should be how well the system can handle the incoming calls and if the poll results are shown. In addition, radio hosts as well as citizens should be satisfied with the service and if this is not the case, it should be sorted out why that is not the case.

e. What are important (pre)conditions that must be or are assumed to be satisfied for the scenario (context features, e.g. needed resources or infrastructure or other characteristics of the environment)?

The citizens using the system should have a phone with a working connection to call to send in their poll answer. The radio host should at least have a phone and laptop to connect to the system and edit polls. Also, the radio host should have some technical literacy to work with the system.

# 4. Use case scenario script

Below are two use case scenario scripts that explain the central storyline of a citizen using the system and voting to a poll. We decided for simplicity to focus on yes/no questions only, but the other use case scenario might be a good idea for extra functionality.

#### Scenario for a closed question with yes/no answers:

<system></system>	'Welcome to Radio Segou's Daily question, the question of today is'
	[audiofile] Where are you celebrating Eid al-Fitr?
<system></system>	'Would you like to leave a voice-message to respond to the question?
	'Press 1 for: No'
	'Press 2 for: Yes'
<user></user>	*Presses 2*
<system></system>	'You can record a message after the beep, to stop the recording press #'
<user></user>	'I am celebrating Eid al-Fitr with my family and loved ones in my family home. I wish everyone a blessed day!' <i>*presses #*</i>
<system></system>	You recorded the following message [audiofile recorded above]
-	Press 2 to save and send this message to Radio Segou
	Press 3 to discard this recording and record again
	Press 4 to quite recording
<user></user>	*Presses 2*
<system></system>	'Thank you for your message, the responses to this question will be played on
	the radio today. We cannot guarantee your message will be broadcasted'

#### Scenario for a closed question with fixed answers:

<system></system>	'Welcome to Radio Segou's Daily poll, the question of today is'
	[audiofile] What sector should be focussed on distribution development aid?
	'Press 1 for: [audiofile] Infrastructure [audiofile]'
	'Press 2 for: [audiofile] Health services [audiofile]'
	'Press 3 for: [audiofile] Educational programmes [audiofile]'
	'Press 4 for: [audiofile] Food security [audiofile]'
<user></user>	*Presses 2*
<system></system>	'Thank you for your vote, you voted [Health Services].'
-	'Would you like to leave a voice-message to elaborate on your choice?
	'Press 1 for: No'
	'Press 2 for: Yes'
<user></user>	*Presses 2*
<system></system>	'You can record a message after the beep, to stop the recording press #'
<user></user>	"Health services is at the basis of every successful society' *presses #*
<system></system>	You recorded the following message [audiofile recorded above]
	Press 2 to save and send this message to Radio Segou
	Press 3 to discard this recording and record again
	Press 4 to quite recording
<user></user>	*Presses 2*
<system></system>	'Thank you for your feedback, the results of this poll will be announced on the
	radio within an hour'

## 5. Interaction and communication

The interaction and communication happens between the citizens and the radio host. The whole interaction and communication between those two participants can be found as a UML Use Case diagram in the Appendix 5. In addition, an Activity Diagram describes the call flow.

See more in Appendix 5

# 6. Information concepts

A UML class diagram for the Foroba Blon Simple Poll System can be found in Appendix 6. Since the application is based on KasaDaka-VSDK, the existing VSDK modules were used and only extended/modified if necessary. To achieve a persistent *User Choice*, a persistent layer was introduced. This layer forwards the chosen VXML item back to KasaDaka to handle this item and to store it into the database.

See more in Section 11 and Appendix 6.

# 7. Technology infrastructure



Via the Foroba Blon platform (see image above), the radio host allows low resource clients (phones) to connect to listeners in a switcher. Thus the incoming calls will be handled.

The Foroba Blon Simple Poll System is build on top of the Kasadaka<sup>2</sup> platform and can run on a Raspberry Pi which has low operating costs. When a call is finished, the answer is saved on The Radio host can connect with the Kasadaka platform via a pc or laptop. With this connection, the radio host is able to set up the question and look at the answers in a web-interface.

# 8. Cost considerations

We assume that all costs are paid by the funding agency and that teams from the Vrije Universiteit van Amsterdam will help with implementation and research. We estimate that the project would take about 3 months and that the implementation cost is about 3800 euro. To run the system, there is 100 euro needed for the equipment and we made an estimation for the calling costs.

<sup>&</sup>lt;sup>2</sup> See: https://www.kasadaka.com/

Project Implementation Costs		
Description	Costs in euros	
Developer Team of 2 for 2 months	Free (VU)	
Research Team of 2 for 2 months	Free (VU)	
Flights for 2 people	1.800	
Extra budget for stay and equipment etc.	2.000	
Total	3.800	

Project Planning		
Month	Project phase	
1	Planning, analyzing and development	
2	In the field testing, using and analyzing	
3	Research results and findings	

Project Runtime Costs		
Description Costs in euros		
Outgoing calls per minute per 1000 calls (if applicable) <sup>3</sup>	~25	
Raspberry PI + Equipment	100	

## 9. Feasibility and sustainability

a. What is the technical feasibility of the scenario?

- Right now we are not able to estimate the capability of the voice service and if it is able to handle multiple calls at the same time and how the load balancer is working.
- For the current project status we are not able to identify the risks and technical obstacles regarding the existing Foroba Blon voice- and web-system and how our developed system could be integrated to create an automatic process as much as possible.
- b. What is the business and (socio-)economic feasibility and sustainability of the scenario?
- The business feasibility is shown by it being a relatively cheap solution that can have a big impact, because it enables ordinary people in the community to have a voice that can be heard by the whole community. Besides this it also allows popular opinion to shine through via the poll system, both can help the whole community.
- There will be a social impact, there will be more communication within the community, this will create more discussion, information sharing and feeling of cohesion.

<sup>&</sup>lt;sup>3</sup> Based on a charge of 0.21 CFA per second according to Malitel

• Because the application is fairly simple and generic it can be used in many scenarios, besides this it will be lightweight, which makes it possible to run on many systems. This allows it to be used more often and runnable on older/replacement hardware.

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

- Language barrier: For the testing phase it could be possible that it will be difficult to implement the right service in the actual language and test it, because the developers do not understand the actual operating language.
- Multiple language support: It could be useful to create a service which is capable of handling the same service in multiple languages. Which would make the whole project much more complex and expensive.

*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 citizens need to have a (mobile) phone
- The radio hosts needs to have a computer with internet access, however for the scenarios in which this is not available we also have a "could have" requirement which allows a voice interface to also manage the system

	Citizen	Radio Host
Must have	<ul><li>Listen the current poll</li><li>Give vote for polls</li><li>Leave a message</li></ul>	<ul> <li>Create polls via the web interface</li> <li>Receive the poll results in a web interface</li> </ul>
Should have	• Select language	• Receive the poll results as diagrams
Could have	<ul><li>Remove previous vote</li><li>Change vote</li></ul>	• Advanced Voice Interface to create polls and change polls
Won't have	<ul> <li>Mobile App to vote</li> <li>Call for intermediate results</li> <li>Call for relistening to previous calls, from themselves or others</li> </ul>	<ul> <li>Mobile App to create polls</li> <li>Mobile App to receive poll results</li> </ul>

## 10. Key requirements

## 11. Prototype description

The final prototype is deployed on Heroku and works with the KasaDaka VoiceXML Switcher. In the KasaDaka application (administration page) we've created a new *Voice Service* 'Yes-No-Service' with several new *Voice Labels* (e.g. radio question, elaborate choice, record beep etc.) and two *Choice Elements* 'Yes-No-Choice' and 'Record-Message'. All voice labels / sound files are created, recorded and translated by our team. In addition, a dedicated web interface was developed to show and present poll results in an easy and understandable way.

a. Description of the design decisions of your application and where relevant how these decisions are made.

Design decisions	Reasoning	
One option was to have the citizen called back by the system instead of having the citizen pay for the entire call. This could be done, for example, by storing the caller number. This would enable the system to call the caller back and the mobile phone costs would be with the radio host. However, this option was not chosen.	We thought about this callback functionality but it would take too much time to implement since a callback functionality is not part of KasaDaka yet.	
<ul> <li>Updating the 'question of the day' by the radio host.</li> <li>We considered multiple options: <ol> <li>Creating an own KasaDaka Voice Service which is dedicated for the radio host to update the question by using a phone call</li> <li>Creating a web interface on which the radio host is able to either record the new message or upload the new message as audio file.</li> </ol> </li> <li>III. Take the assumption that the radio host gets access to the KasaDaka system and edits the question of the day directly at KasaDaka (by updating the Voice Label).</li> </ul>	Option II) introduces new problems like converting the audio files into the right format or ensuring that the uploaded file has the right format. Option I) would be the best option in terms of usability. However, we took option III) since this option has the optimal cost-time-relation.	
Since the system should be open for further implementations and features, we decided to store all chosen VXML input elements, even though they have no functionality yet and this might lead to more data.	This keeps the system open for further implementations like multiple-options-questions. Through a simple filter query, it is possible to filter by VXML input element.	
Multiple language support for the citizen (phone call) and the radio host (web interface for poll results).	Even though it costs more time and effort to implement such multi language support for both parties, we decided to implement it. It opens the usage of the system in different regions—regardless of the place of use.	

#### b. Back-end implementation

As already mentioned earlier, the Back-end is based on the KasaDaka-VSDK open source project. According to section 6 and the UML diagram below in section e, several VSDK modules have been reused. In addition, the ChoiceSaved module was introduced as a persistence layer to store the chosen VXML items. This persistence layer generates VXML files which use the *form-submit node* and the *namelist* attribute instead of the *goto-node*<sup>4</sup>. To achieve this, a Django model *ChoiceSaved* was implemented together with its model-manager. The general functionality and other modules of KasaDaka were not affected.

#### c. Front-end implementation

To create the dedicated web interface for analyzing the poll results, a new Django App was added, namely *Forobablon*. This app is structured as a basic Django App and is only responsible for fetching the poll results and showing these results as a web page.

The application is prepared for multi language. This means that every string and number or date formatting is based on the current language. Available languages for the current release are English, Dutch and German. The language is selected either by browser request (request header) or manually via the language selector at the bottom of the page. Additional languages can be installed and used by adding translations files—without actual code modifications..

The frontend consists of two different pages

- the *index* page lists all dates with available poll results. The dates are also linked to the actual result page.
- the actual *result* page shows the current state and results of the chosen date. The states are displayed as I) bar chart visualization and II) numbers. If available, the result pages consists also of a list with all recorded voice messages which the citizen/caller might have left. The voice message can be played directly on the website. Hence, if the computer is connected to the radio station, the radio host could play the audio messages directly on the broadcast station.

#### d. UI

As UI colors we have chosen a black/white inverted color scheme. To keep the interface as much clean as possible, only the necessary information are displayed. Due to the multi language support, dates and numbers are formatted automatically. As bar colors, we have avoided colors like green (for yes) and red (for no), since in an europene culture red and green are associated with good and bad. This meaning could be different in other cultures. Hence, we have chosen colors with 'no meaning'. Nevertheless, the colors should have a good contrast to provide explicitness.

√	Yes		12
×	No	5	

Further UI screenshots can be found in Appendix 11.

<sup>&</sup>lt;sup>4</sup> <u>https://www.w3.org/TR/voicexml20/</u>

e. Data model



#### **12**. Pointer to how to access the application

a. GitHub repository

URL: <u>https://github.com/FunkeMT/KasaDaka-VSDK</u>

b. Heroku - KasaDaka

URL: <u>http://foroba-blon.herokuapp.com/</u> User: group2 Pass: group2secure

- c. Heroku Simple Poll System Web Interface URL: <u>http://foroba-blon.herokuapp.com/en/forobablon/results/</u>
- d. KasaDaka VoiceXML Switcher URL: <u>http://foroba-blon.herokuapp.com/vxml/start/3</u> Call number: +31 020 369 76 64
- e. YouTube demonstration video URL: <u>https://www.youtube.com/watch?v=sUhDYByVE2s</u>

#### 13. Short Usage scenario

It would be advisable to deploy the application in a secure testing environment. Start with one or two radio stations which are known with the Foroba Blon project and explain to the radio hosts what the goal of the project is. Have the radio host introduce the new feature everyday for a week on the radio to make the public aware of the possible benefits. Start with simple non contentious questions based on current local news. The poll system could also be used for a meta question like: "Do you think the new poll system is a good way to voice concerns and opinions in the community? " to see what listeners think about this new technological feature of community radio. The results interface will give the radio host an idea about the popularity of the system with the radio audience.

## 14. Feedback questions

During the presentation we got questions about the cost aspect of our application. Are people willing to do expensive number calls to participate with their community radio? This question raises multiple others: is calling in for one minute to a radio station really that expensive? Do people think it is worth that price? If we devise a system in which the radio station pays the cellular charge, do they have the funds for this? How do we know people won't take advantage? Should we then implement a limitation on the amount of times a station calls back? Do African providers charge per minute or per second, if the latter, streamlining the choice-menu would be beneficial. Also something to think about it: we want to step away from the patronizing type of development aid the West has been exporting for years. In that sense, having africans pay (a small amount: 0.00034 euro per second according to the biggest provider) for their participation in the public debate would be justified, maybe even desirable. If needed, one solution that we could consider a callback system.

We also got some feedback from students about how we verify fake-news and that we maybe should focus more on one subject. Furthermore, we were unsure about the amount of languages the system needs to be able to handle: we are aware a multitude of languages are spoken in rural areas, however, does an individual radio show broadcast in more than one language? Are the different shows broadcasted in different languages from one radio station? If so, would it be instrumental for the radio host to select the language he wants to use for his show? Do all radio hosts have enough technical skills to make adjustments to the settings of the system?

## 15. Discussion of Scope and Fidelity

Initially we wanted questions to have more possible answers than just yes and no. The current prototype does not have the required back end for this. It is too difficult for the radio host to add this on the Kasadaka page because it takes a lot of clicks and different uploads. The problem with implementing a system with more answers is that the voice based application will be dynamic while Kasadaka is not really set up for this. The radio host could require a question with 3 options on a given date, and a question with 5 options on the day after. The implementation requires some form of poll creation through the web interface for the radio host which will be converted to vxml for kasadaka to use. If the user interface for the poll creation is fit for radio hosts in rural areas this would create a great tool for community radio but unfortunately, the scope was not fit for this course.

Another feature that was dropped due to time constraints was the possibility of an open question. This could be added by adapting the current prototype and removing the closed question and replacing it by a recording prompt. The system should place the recordings in a voicebox for the radio host to access.

#### 16. Conclusions

Creating a software system for any development project can be challenging as it requires a developer to think out of the box. A lot of the technologies that developers and engineers are used to in the global north are not available in the global south. This creates restrictions which have to be coped with. The voice based application described in this document is our interpretation of a very simple forum where people can express their opinions. The implementation is just a prototype but it shows the possibilities of Kasadaka in a limited scope of just 7 weeks of brainstorming and development. The possible extensions of the system, as discussed in this document, could be implemented to increase the scope of the application.

#### APPENDIX

## 4. Diagram for Use case scenario script

This is another example for a use case scenario script.



# 5. Use Diagram and Activity Diagram for Interaction and communication



# 6. UML Class Diagram for Information concept



## 11. Prototype Description and UI



Foroba Blon Simple Poll System				
$\checkmark$	Ja	12		
×	Geen 5			
Opnames				
• ▶ 0:00 / 0:06 - (•) : [18 mei 2020 12:31]				
NL EN DE				