<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Modification</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>10/04</td>
<td>Doc creations</td>
<td>10</td>
</tr>
<tr>
<td>0.2</td>
<td>29/04</td>
<td>Milestone 2</td>
<td>6</td>
</tr>
</tbody>
</table>
| 0.3     | 20/5   | ● Implemented all feedback.  
      ● Changed diagrams and aligned the textual explanation with the feedback from Francis and from peers.  
      ● Changed storyboard  
      ● Added deployment diagram.  
      ● Added a complete technical documentation | 12    |

<table>
<thead>
<tr>
<th>Student</th>
<th>ID</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jasper Theunissen</td>
<td>2682807</td>
<td><a href="mailto:j.h.theunissen@student.vu.nl">j.h.theunissen@student.vu.nl</a></td>
</tr>
<tr>
<td>Nina Wilschut</td>
<td>2617077</td>
<td><a href="mailto:n.wilschut@vu.nl">n.wilschut@vu.nl</a></td>
</tr>
<tr>
<td>Pauline Remizowa</td>
<td>2618728</td>
<td><a href="mailto:p.remizowa@student.vu.nl">p.remizowa@student.vu.nl</a></td>
</tr>
<tr>
<td>Osayd Al Absi</td>
<td>2739102</td>
<td><a href="mailto:o.n.a.alabsi@student.vu.nl">o.n.a.alabsi@student.vu.nl</a></td>
</tr>
<tr>
<td>Katarina Vleugels</td>
<td>2590538</td>
<td><a href="mailto:e.k.d.p.vleugels@student.vu.nl">e.k.d.p.vleugels@student.vu.nl</a></td>
</tr>
</tbody>
</table>
Table of content

1. Summary of key idea 3
2. Actors and goals 3
3. Context and scope 4
4. Use-case scenario script 6
5. Interaction and communication 7
6. Information concepts 9
7. Technology infrastructure 10
8. Cost considerations 11
9. Feasibility and sustainability 11
10. Ethical discussion 12
11. Conclusion 12

Appendices
1. User feedback 13
2. E3 Value models 16
3. MoSCoW 18
4. Prototype description: Textual and diagrammatic description of your project result. 18
5. Pointer to the Application code (Can be a URL or appendix) 22
6. Pointer on how to access the application 22
7. Short Usage scenario 22
8. Feedback questions. 23
9. Discussion of Scope and Fidelity. 23
10. User evaluation and reflection 23

Table of figures & tables:

Figure 1: Use-case diagram 5
Figure 2: Storyboard 7
Figure 3: Start/stop activity-diagram 7
Figure 4: Voting state-diagram 8
Figure 5: Voting sequence diagram 8
Figure 6: Voting class-diagram 9
Figure 7: Deployment of our system 10

Table 1: Actors and Goals 3
Table 2: Costs 11
Table 3: E3 value netcashflow 12
1. Summary of key idea

The key idea of the envisaged system is to provide a utility tool for community radio stations to make their broadcasts and programs more engaging and interactive by enabling their listeners to participate in cost-free voting. Votes are simple yes/no questions asked during a radio broadcast. During a broadcast the radio host will ask a question to the listeners and provide two phone numbers, A and B; A to vote “yes”, and B to vote “no” to the question asked on the radio. Next, the listeners will bip-call either one of those numbers to cast a vote. Bip-calling is calling a phone number and disconnecting immediately to avoid costs. Number A and B will still receive the missed calls, and the amount of missed calls on each phone number represents the amount of votes casted for that option.

The old system had some major drawbacks for the radio station. Votes had to be counted and checked for duplicates manually, which is very time consuming and prone for human error that can lead to wrong voting results and decreased trust in the system. The major improvement of our new system is the automated counting and validation of the votes. The automation makes our new system significantly more fail-proof to human error and increases time efficiency, which enables the radio stations to organize multiple votings during a single broadcast. In addition, our new system is more secure regarding the privacy of the listeners. With the old system the phone numbers of listeners who voted are exposed to the radio station, which has a negative effect on the listeners privacy. This privacy concern is solved with our new system by encrypting all the incoming numbers so it is never known which phone number voted, and the listeners don’t have to worry that their vote can be held against them. Lastly, our system is fully manageable through both voice and web interface. This makes our application more inclusive to illiterate and less digitally skilled people by providing optionality regarding the ICT used for managing the votings. By running our application on a Kasadaka, internet access is not even required to use our web-interface, just a device that can run a web browser and connect to a local network. This improves the flexibility, availability, accessibility and usability of the system by not depending on ICT technology that is not always available in rural Mali such as the internet.

2. Actors and goals

Our use-case describes two types of main actors within our context; the organizers of a voting, which can either/or both be the radio host and/or a radio journalist, and the listeners who want to participate in votings.

<table>
<thead>
<tr>
<th>Actor</th>
<th>Operational Goal</th>
<th>Responsibility in the system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organizer</td>
<td>• Organize bip-votings in a cheap, easy and reliable way.</td>
<td>• Preparing the system for voting (setting a start time/date), broadcasting the voting phone numbers, shutting down the voting (setting an end time/date), and retrieving the vote counts.</td>
</tr>
<tr>
<td></td>
<td>• Start/stop a voting and access the results and publish the results.</td>
<td></td>
</tr>
</tbody>
</table>
Radio hosts
- Ask bip-voting questions to the listeners during a radio broadcast.
- Broadcast the results of the bip voting, per voting option.
- Interact with listeners during a radio broadcast
- Repeat the voting question often enough during the radio broadcast so listeners know what they are voting on
- Manage voting through a voice- or web-interface.

Radio Journalist
- Provide interesting statements/questions/topics for the radio station to vote on.
- Manage voting through a voice- or web-interface.

Voting participants
- Have the opportunity to vote without worrying about cost or their vote not being registered.
- The ability to vote yes or no.
- Find out what the voting numbers are and make one bip one voting number, per person.
- Participate in voting.

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.

To clarify our use-case and the included scenarios we have created a UML use-case diagram (Figure 1). The diagram shows that either a journalist or radio host can start the use-case “organize voting”. This is always managed through a web or an IVR voice interface which includes authentication (i.e. login sequence). Furthermore, organizing a voting will always include the use-case “broadcast voting questions” and “disseminate voting phone numbers” as these are essential to enable listeners to participate. Moreover, the diagram shows that organizing a vote has some utilities: the ability to start or end a voting and have the voting results calculated automatically. The diagram also shows that the voting participants can cast a vote which is either a yes or no vote. However, the specific use-case where the voting participant hears the questions and the numbers to cast a vote is out of scope of our system, we deem this as the responsibility of the listener as they are required to have a working radio, which is not covered by our bip-vote system. Subsequently, disseminating the voting question and the numbers to cast votes is within our scope. This is an essential use-case for the radio stations because it's part of organizing a voting, and that's exactly what our system will be used for and improves.
Who are the (external) stakeholders to the use case and what are their concerns?
The external stakeholders are the listeners who have the need to vote without having to worry about the costs or whether their vote is counted correctly or not. They also need to know what the voting phone numbers are. The organizers of the bip-voting want to improve their listeners’ participation and engagement by supplying a free, easy and reliable way to interact with the radio station. They also want to know the results of votes in a swift and automated manner to organize multiple votings during one broadcast. The organizers want to be able to manage and organize the voting with a web interface to exclude calling costs; however, this would require a computer or smartphone which may not be accessible and some organizers may be illiterate. Therefore, they also want to be able to manage and organize their votes through a voice interface (IVR).

What is the scope of the scenario?
The main scope of the scenario will focus on the phone numbers that are broadcast, the ability to receive votes and how the results are automatically counted and accessible to the radio station that organizes the voting. The boundary of our scope is the local community radio station that will implement the system in a rural context. This includes the technical, social and financial constraints of community radio stations and the low-resource context that they operate in. However, we do not have any scope on the kind of events or topics that are voted on during a broadcast since the radio station covers many different topics and this does not affect the technical side of our system. The possibility for listeners to cast an incorrect vote if they bip-call the wrong number, we also consider out of scope of the system.

What are success or performance measures for the scenario?
The success of the scenario will be measured by the value it adds to common methods, practices and activities required for a broadcast and to organize a voting. Additionally, the amount of free time radio
stations will obtain by having the votes automatically calculated and how the system can be implemented in their current broadcast activities will determine its success. Therefore, the success of the bip-vote system will depend on a couple of quality requirements: *time-efficiency*, *resource-utilization*; time-efficiency will be measured by measuring and comparing the time it takes to organize a voting between the old and new system. Resource utilization will be measured by the amount of ICTs required to run and operate our system, which in our case should be minimized because in Mali ICT availability cannot be taken for granted. Lastly, in Mali illiteracy is common and on average people lack digital skills. Therefore the usability of the system is very important and the deployment of our automated bip-vote system should not require a change in behavior from listeners in order to cast a vote and participate. Listeners could get confused on how to cast a vote and lose trust in the system if the usual way changes too much.

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

Electricity is a key necessity, not only for our system but also for the radio station to operate in general. All actors must have access to a phone with GSM-network connection. Our system will be deployed using a Kasadaka Raspberry Pi. The skills and knowledge required to connect our system to the GSM network to use the voice interface, setting up a local network to manage the voting with the web interface is an essential requirement for the organizers. Additionally, the organizers have to have a sufficient mobile credit amount if they wish to manage the voting with their phone. If they wish to manage the voting via the web interface, organizers must have a device that runs a web-browser and be literate. Voting participants must have access to a radio to be able to hear the voting question and to know the numbers that have to be called to vote “yes” or “no”.

4. **Use-case scenario script**

The use-case scenario script in Figure 2 describes the communication and interaction between the organizer of a voting, the web interface and the listeners that cast the votes. First the voting is opened and the question is disseminated during the broadcast. Listeners use their phone to bip-vote either yes or no. Then the radio station closes the voting and checks the results. Although not pictured in Figure 2, opening, closing, and checking results can also be done over the phone.

![Figure 2: Storyboard](image-url)
5. Interaction and communication

The activity diagram in Figure 3 shows our two different applications and its respective activities. We’ve decided to combine start/stop the voting with voting “yes”, and retrieving the results with voting “no”, to minimize the number of applications to run and maintain. When a user performs the first activity, the application checks whether the caller is a registered user. If the phone number is from a registered user (e.g. the organizer), the application will forward to the start/stop section. If the caller is not a registered user, the application casts a “yes” vote and disconnects. This is particularly important to stop a listener from incurring unwanted costs and stops them from accessing voting states. When a user performs the second activity, similar events happens except the voting results are played if the user is registered and a “no” vote is cast if the user is unregistered.

Figure 3: Start/stop activity diagram

Figure 4 shows the two different states that a voting can be in, a voting can be “closed” or “open”. Before a voting starts the initial state of a previous voting that took place will always be “closed”, if we want to start a new voting the first state the voting will be in is “open”. When we enter the “Open” state, the start_new_voting() activity is initiated and sets the marker of the voting_status to “O”, meaning open and truncates the database in which we register all the incoming votes and phone numbers to prevent duplicates. While the voting is “open” it allows incoming votes. When the organizers decide to stop the voting and trigger the Stop event, the guard is checked if the voting_status is “O”. If that is the case the activity close_voting() is triggered. On entering the
“closed” state the voting_status is set to “C”, meaning closed and the application will deny any incoming votes to be added to our database. The voting is closed and we can exit by leaving the voting state closed.

**Figure 4: Voting state diagram**

Figure 5 shows the sequence diagram of our bip-vote application. It displays the use-case where an organizer wants to start/stop a voting. The organizer calls the voxeo application, if the caller ID is not recognized as a registered number the application will make a call to the backend of our system to store a “yes” vote, else the menu for opening and closing a voting will be prompted. It should be noted that currently our systems backend is hosted in a cPanel cloud service purely for demonstrative purposes. If our application would be implemented at a radio station, cPanel would be exchanged for the Kasadaka raspberry pi to host our backend and database locally and connect to GSM network to interface with Voxeo.

**Figure 5: Voting sequence diagram**
6. Information concepts

Our bip-vote application uses two database tables which are shown in figure 6. One to represent a voting (votingQ), its results and the status the votingQ is in (open or closed), the other table is to store and encrypt the incoming phone numbers that voted. This allows us to easily check for duplicates by having the phone number as a primary key of the table which automatically disallows duplicate table inserts.

![Class diagram of our data structure](image)

**Figure 6: Class diagram of our data structure**

7. Technology infrastructure

As of now, the system is designed using a cloud solution (cPanel cloud hosting) as it is the best way to work with Voxeo when using dynamic data and server side languages (PHP in our case). Due to this being a shared project, it was easier to operate in a cloud hosting environment rather than a local device which will cause a multitude of issues with accessibility. A cloud environment often ensures that your data will not be wiped out in a case of hardware failure or malfunction. However, deploying this system in Mali would work best in a local storing environment as provided by Kasadaka. The reason why it would work best in a local hosting environment is because it is cheaper to gain a larger amount of storage space with storage devices than it is with the cloud. Moreover, a local hosted system would only be a one time investment for the radio stations, whereas cloud hosting services are monthly recurring costs which over time are less financially attractive because this can affect the affordability if hosting prices change. Also, the cloud is highly dependent on internet connectivity which in this case is a liability to the system due its uncertain availability. Therefore, to increase the deployability we want our system to utilize technologies of which we now are available, such as the GSM network. If internet availability were a necessity for the system to work, our system would be unavailable in remote and rural areas of Mali and therefore exclude community radio stations without internet access to acquire and use our system, while these radio stations might be very interested in increasing their listeners' engagement and interaction. The figure below shows how our system would be deployed in the field and which technologies it utilizes.

1. The community radio station utilizes radio frequencies to start the voting and disseminate the voting question and the phone numbers to cast a vote.
2. Listeners who have a radio can receive and listen to the question posed by the radio station, then they can cast a vote by utilizing the GSM network with their mobile phone to bip-call cost-free.
3. The GSM network forwards the votes of the listeners to the Kasadaka that runs our application.
4. The Kasadaka is connected to the GSM network and creates a local network. This enables the application to be managed by any device that runs a web browser and any phone that connects to the GSM network. By using a local network, internet access is not required.
5. The radio station is able to manage the voting (start/stop, results) through either or both web- and voice interface. A GSM phone connects with the GSM network which connects to the Kasadaka to interface with each other. A laptop connects to the local network and uses the web-interface.

Figure 7: Deployment of our system
8. Cost considerations

For our system to work, the radio station that implements our bip vote system has to make costs. First, they would have to purchase the Kasadaka system which allows them to use our application. Furthermore, our bipvote system is manageable though our IVR system which requires phone calls to be made and initiates calling costs. On the other hand, we also provide a web interface to manage our bip-vote application, because our system is running on a Kasadaka that outputs a local network, internet access is not required to manage our application through the web interface. This provides the option to manage our bip-vote application only through the web interface, excluding all the costs that are required to manage our system through IVR, hence, the web interface is financially more attractive for radio stations. Table 1 shows our estimations of these costs and are based on the paper from J. Gordijn et al. 2019 in which an application with similar properties was used in Mali within the same context.

Table 2: Costs

<table>
<thead>
<tr>
<th>1 Phone call</th>
<th>650,- fCRA (1 euro)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kasadaka with Bip-vote application</td>
<td>100.000,- fCRA (€150)</td>
</tr>
</tbody>
</table>

9. Feasibility and sustainability

Because our system is extending the interactive possibilities of a radio broadcast it is difficult to make direct revenue with our system because it will be implemented in common broadcast practices of the radio station. However, our system does provide the radio stations to extend their current business model, for example, the opportunity for customers to obtain the public and local opinion about some binary question (yes/no). This service could be sold for money in a similar way that is currently done with selling local announcements or air-time. Therefore our customers could be citizen journalists, advertisers, NGOs, farmers or anyone else who is willing to pay to know the public opinion about a question. The extra revenue our application can generate is dependent on the amount of customers we can sell our public opinion service to. We have estimated to host 2 bipvotes per broadcast per day, which we estimate as feasible due to the improved time-efficiency through automation.

However, to accurately say something about the feasibility and sustainability, it also depends on how the radio station is going to use our system. If the radio station would manage our application by only using the IVR, the required calls would initiate a large amount of costs for the radio station. To organize 1 voting, at least 3 phone calls have to be made which would cost around 1.950 fCRA (3 euro) per voting. If the radio station would only use the web interface, the costs of calling are excluded, however the radio station would in both cases need to purchase the Kasadaka and set up the system which might initiate costs if the required skills and knowledge are not present.

To get a broader understanding of how the different ways of using our bip-vote application affect costs we have created two e3 value models and calculated the net cash flow for the radio station. The first model shows the costs for a radio station that would only use the IVR of our system, the second e3 model shows a situation where the radio station would only use the web-interface. We have based our quantification numbers on the information available in the paper from J. Gordijn et al. 2019 that describes a similar case and average hardware prices of the Kasadaka. What is really interesting to see in the comparison of the two models, is the difference of the amount of monthly revenue of the radio station that only implements the IVR and the one only using the web-interface. Of course we could argue that our estimations are inaccurate, especially because we have modeled a
situation where a radio station organizes 2 votings per day for which they both get paid. In reality this will be different because it depends on the amount of paid customers the radio stations can attract. However, the model does clearly show how using the IVR has a significant negative effect on monthly revenue while the web-interface has a significant positive effect on monthly revenue. To see the full E3 value models and its corresponding net cash flow calculation, please see appendix 2.

Table 3: E3 value net cash flow calculation for radio station using our application

<table>
<thead>
<tr>
<th></th>
<th>Net Cash Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radio station, using IVR only</td>
<td>+23,000,- fCRA (€35)</td>
</tr>
<tr>
<td>Radio station, using web-interface only</td>
<td>+140,000,- fCRA (€213)</td>
</tr>
</tbody>
</table>

10. Ethical discussion

The system we are devising aims to streamline the voter and counting process of a radio station. It enables the radio station to save time without leading to the reduction of involved people. As it is a very specialized system, it has a fairly low risk factor; however, if the system malfunctions or is hacked, polling outcomes can have a great effect on public opinion (Dalhgaard et al., 2016). While this is always a risk when sharing polling outcomes, this can be particularly negative if the change of public opinion is based on information that is false.

Because our system can be used by more people than just the radio station, it becomes more important that the results are correct and not tampered with. This is especially the case when results are sold in a similar way to announcements. Another thing that should be considered is the privacy of the people voting. Like in most things nowadays this should be anonymous. To combat this problem the numbers are saved using a hash technique (md5), only with the current voting, to ensure there are no double votes etcetera. After the voting ends the numbers are also deleted.

11. Conclusion

For this project we created a bip-voting system in which voters can safely send in a bipvote, and radio stations or third parties can broadcast a question and manage the information efficiently. Where there was first a time-consuming process where the votes had to be counted by hand, we are now able to offer a high-value, relatively low-cost option by doing this with an automatic solution. After switching from four applications to two, the application became a simple, accessible and innovative way to interact with radio listeners and manage information, which is important as the areas we are focusing on are regions with a lower level of internet access, and literacy obstacles. With the use of PHP, SQL, Voxeo, and a Kasadaka the applications are able to offer a more time-efficient process, no human errors, the security of privacy of both parties and accessibility. On the other hand, there are more costs and ICT knowledge involved in the process, as there are now localhost servers (Kasadaka) and applications that are necessary, in addition to the team and people being able to understand the process, but in our opinion with the use of the right means the costs and required knowledge should be achievable. During this process we wanted to create a website together with the voting system, but due to the time pressure we were unable to integrate the prototype website to the working voting system. In the future this may be something to look at, but for now we omitted it; however, we did manage to supply a web-interface to manage our application.

With this bip-voting application we were able to create an accessible platform for data passage, for radio stations to better manage and organize their voting questions, and anyone who wants to vote in a safe and secure method.
Appendices

1. User feedback

1. As of now, the radio stations can hear the result of the current open voting question, will that suffice?

   **Answer:** It is very important that the radio station can hear the result of the *last* voting question regardless of the status of the voting question.

   **Response:** From that feedback we now have an SQL query that checks the result of the *last* voting question. The code can be found below.

   ```sql
   $sql = "Select Yes_Votes FROM VotingQ ORDER BY Voting_Q_ID DESC LIMIT 1";
   $result = mysqli_query($conn, $sql);
   if(mysqli_num_rows($result) <= 0) {
      die("<script>alert('No data in the database!');</script>");
   }
   while($row = mysqli_fetch_array($result)) {
      $yesVotes = $row['Yes_Votes'];
   }
   
   $sql2 = "Select No_Votes FROM VotingQ ORDER BY Voting_Q_ID DESC LIMIT 1";
   $result2 = mysqli_query($conn, $sql2);
   if(mysqli_num_rows($result2) <= 0) {
      die("<script>alert('No data in the database!');</script>");
   }
   while($row = mysqli_fetch_array($result2)) {
      $noVotes = $row['No_Votes'];
   }
   
   ?>

2. Right now the voice application relies on select inputs for the voice menus instead of voice inputs, does that work well enough for the case?

   **Answer:** It's preferable to have the option to navigate the voice menu with voice inputs,
however it isn’t an extremely important feature.

Response: As of now, we found that using voice inputs provide for a very inconvenient user experience as the voice recognition on most phones and especially older phones are inaccurate.

3. Will there be a need to implement another language besides English and French (voice menu will be completely in French with the final implementation)?

Answer: While French will suffice, it would be a huge addition to have an option for further customisation.

Response: In our start/end voting question applications, we have links for audio files in French that are stored in our cPanel file manager. All the radio stations will have to do is upload a pre-recorded audio file into the cPanel and change the link of the audio “src” to the pre-recorded audio files they want.

Response: In our start/end voting question applications, we have links for audio files in French that are stored in our cPanel file manager. All the radio stations will have to do is upload a pre-recorded audio file into the cPanel and change the link of the audio “src” to the pre-recorded audio files they want.

4. How is everything going with the count system?

Response: The count system is working perfectly, with validating duplicates and increasing the count in the database when a call is received for either a “yes” or “no” vote.

5. How to publish the questions?

Response: We only provide a system where the radio stations can open a voting question so that they could track incoming bippotes from participants. We also provide them the feature to be able to see or hear the results. They still broadcast the question themselves through their radio, broadcasting the question itself is not within our scope.

6. This system seems not easy to implement. Have you encountered any difficulties?

Response: For sure, we encountered a problem early in development in attempting to host backend files through Voxeo. Voxeo does not support any server-side languages so our php files weren’t working properly hence having to store them via cPanel. We also encountered a challenge in prompting the results in French. For that, we used the xml:lang=“fr-fr” attribute as seen below.

7. There are a total of four applications, wouldn’t that be confusing? It would be better if all the functionalities can be done with a lesser amount of applications.
Response: We now only operate with two applications. One for opening and ending a voting question and one for results. These numbers will be owned by the radio stations, the app for opening and ending a voting question will also be used for a Yes vote from a participant and the results one will be used for a No vote from a participant. Below is the code used to identify the caller coming in, detecting whether it is a participant or the radio station.

```xml
<xml version="2.1">
 <var name="callerID" expr="session.callerid"/>
 <form>
 <block>
 <if cond="callerID == '31685947615'">
 <goto next="#langmenu"/>
 <elseif cond="callerID == '31641157016'">
 <goto next="#langmenu"/>
 <elseif cond="callerID == '31623697899'">
 <goto next="#langmenu"/>
 <else/>
 <goto next="http://webhosting.voxco.net/208129/www/yesVotes.xml"/>
 </if>

Additional feedback: Would prove extremely beneficial if the radio stations can also view the results on a web interface.

Response: Yes, a web interface has been developed for the radio stations to be able to open/close a voting question and view results. It is also provided in English and French, just like the voice application.
2. E3 Value models

Quantification of the Radio Station using only the web-interface

<table>
<thead>
<tr>
<th>Interface</th>
<th>Port</th>
<th>Transfers</th>
<th>Occurences</th>
<th>Valuation</th>
<th>Transferred</th>
</tr>
</thead>
<tbody>
<tr>
<td>{MONEY, broadcast service}</td>
<td>in: MONEY</td>
<td>MONEY</td>
<td>60</td>
<td>4,000</td>
<td>240,000</td>
</tr>
<tr>
<td></td>
<td>out: broadcast</td>
<td>(all transfers)</td>
<td>60</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>{audience, broadcast}</td>
<td>in: audience</td>
<td>(all transfers)</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>out: broadcast</td>
<td>(all transfers)</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>{opinion, engagement}</td>
<td>in: opinion</td>
<td>(all transfers)</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>out: engagement</td>
<td>(all transfers)</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>INVESTMENT</td>
<td></td>
<td></td>
<td></td>
<td>-100,000</td>
<td>140,000</td>
</tr>
</tbody>
</table>

Customers: local, advertiser, journalist
Know public opinion about binary question

OCCURRENCES: 60
(average of 2 votes per broadcast per day)
Quantification of the Radio Station using only IVR interface

<table>
<thead>
<tr>
<th>Interface</th>
<th>Port</th>
<th>Transfers</th>
<th>Occurences</th>
<th>Valuation</th>
<th>Transferred</th>
</tr>
</thead>
<tbody>
<tr>
<td>{MONEY, broadcast service}</td>
<td></td>
<td>240,000</td>
<td></td>
<td>240,000</td>
<td></td>
</tr>
<tr>
<td>in: MONEY</td>
<td>MONEY</td>
<td>60</td>
<td>4,000</td>
<td>240,000</td>
<td></td>
</tr>
<tr>
<td>out: broadcast service</td>
<td>(all transfers)</td>
<td>60</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>{phone call, MONEY}</td>
<td></td>
<td>-117,000</td>
<td></td>
<td>-117,000</td>
<td></td>
</tr>
<tr>
<td>in: phone call</td>
<td>(all transfers)</td>
<td>180</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>out: MONEY</td>
<td>MONEY</td>
<td>180</td>
<td>650</td>
<td>-117,000</td>
<td></td>
</tr>
<tr>
<td>{audience, broadcast}</td>
<td></td>
<td>0</td>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>in: audience</td>
<td>(all transfers)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>out: broadcast</td>
<td>(all transfers)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>{opinion, engagement}</td>
<td></td>
<td>0</td>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>in: opinion</td>
<td>(all transfers)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>out: engagement</td>
<td>(all transfers)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>INVESTMENT</td>
<td></td>
<td>-100,000</td>
<td></td>
<td>-100,000</td>
<td></td>
</tr>
</tbody>
</table>

INVESTMENT 23,000
3. MoSCoW

<table>
<thead>
<tr>
<th>Must Have</th>
<th>Should Have</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allow radio station to open a voting question</td>
<td>Allow radio station to open a voting question</td>
</tr>
<tr>
<td>Radio stations can hear result of the last voting question</td>
<td>View results on a website</td>
</tr>
<tr>
<td>Allow participants to bip vote yes</td>
<td>Provide a different language</td>
</tr>
<tr>
<td>Prevent duplicate votes</td>
<td>Allow users to use their own audio files for custom localisation</td>
</tr>
<tr>
<td></td>
<td>Simple verification(login) system</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Could Have</th>
<th>Won’t Have</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detect <em>voice</em> inputs</td>
<td>Radio stations can hear results of all voting questions</td>
</tr>
<tr>
<td>Storing the voting question</td>
<td>No interaction with system and participant(voter)</td>
</tr>
</tbody>
</table>

4. Prototype description: Textual and diagrammatic description of your project result.

Our application enables radio stations to open and close a voting session so that votes can be automatically stored and tracked. This can be done via voice application or web interface. They can also hear the results from the voice application as well as view the results on a web interface.

We previously had four applications, two for the radio stations to call (one to open/close a voting question, one to get results of the voting) and the other two for voting (yes and no). However, part of the feedback was that having multiple applications could cause lots of confusion, hence the decision to reduce them to two applications. The idea behind this decision is that the radio station can own both these numbers for the applications and use them both while simultaneously allowing them to be the numbers to store the incoming votes from participants.
Front-End and Back-End Implementation

When the radio station calls the open/close voting application, they will be met with an option of choosing whether they would like to proceed with English or French. Once they have made a decision, a menu will be provided with the option of either opening or closing a voting question by selecting “1” or “2”. When they select the desired action, the PHP from our backend will be triggered to either open or close the voting question. This will insert or update the column “Voting_Status” that is in the “VotingQ” table in our database that is hosted in the backend via cPanel. Below is a screenshot of the “VotingQ” table.

<table>
<thead>
<tr>
<th>Voting_Q_ID</th>
<th>Voting_Status</th>
<th>Yes_Votes</th>
<th>No_Votes</th>
<th>Start_Date</th>
<th>End_Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C</td>
<td>1</td>
<td>0</td>
<td>21/05/2022</td>
<td>21/05/2022</td>
</tr>
<tr>
<td>2</td>
<td>C</td>
<td>0</td>
<td>0</td>
<td>21/05/2022</td>
<td>21/05/2022</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>0</td>
<td>0</td>
<td>21/05/2022</td>
<td>21/05/2022</td>
</tr>
<tr>
<td>4</td>
<td>O</td>
<td>42</td>
<td>32</td>
<td>21/05/2022</td>
<td>NULL</td>
</tr>
</tbody>
</table>

The columns “Yes_Votes” and “No_Votes” from that table are used to extract the results for the radio station to hear or see.

When someone other than the radio station calls the open/close application, the number in the “Yes_Votes” column will increase. If they call the results application the number in the “No_Votes” column will increase. Besides the “VotingQ” table we also have a “Numbers” table. The “Numbers” table is only used to store encrypted phone numbers to prevent duplicate votes. Once a voting question is opened or closed, the numbers will be truncated from the table so that the owners of that number can vote again for the new voting question. Below is a screenshot of our “Numbers” table.

<table>
<thead>
<tr>
<th>Phone_Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>2612a10ac9b1780e70a8d7b2aba0fffa1</td>
</tr>
</tbody>
</table>

Our backend, that consists of PHP files and SQL statements within the PHP files, is where all the magic happens. Once the radio station inserts their choice, Voxeo will trigger the link to our PHP file which contains code that receives data passed from Voxeo and calls our database to then execute SQL commands to insert/update/read/delete the appropriate data received from Voxeo. Below is a four part screenshot of the code from Voxeo to the PHP that allows for a new voting question to be opened.

Declaring the variables (O for Open, C for close)

```php
<br>
</form>
<form id="form1">
  <p>Thank you for opening a new voting question! This program will now end.</p>
  <input type="hidden" name="status" value="O" />
  <input type="hidden" name="status1" value="O" />
</form>
```

Triggering the PHP link and passing our variable status.
In PHP, we insert the date the question was opened, check if there was a voting question open, and if so it will be closed. Then we insert the new voting question being opened and truncate any numbers in the “Numbers” table if there was a previous question with numbers stored.

The same concept is used for storing votes from participants. Below is the code for storing yes votes from voxeo to PHP.

```php
<?php
// Insert query
$sql1 = "UPDATE VotingQ SET EndDate = " . $date . " WHERE Voting_Status = '0'";
if ($conn->query($sql1) === TRUE) {
  echo "New record created successfully";
} else {
  echo "Error: " . $sql1 . "<br>" . $conn->error;
}

$sql2 = "TRUNCATE Table Numbers";
if ($conn->query($sql2) === TRUE) {
  echo "New record created successfully";
} else {
  echo "Error: " . $sql2 . "<br>" . $conn->error;
}
?>
```

In PHP, we insert the date the question was opened, check if there was a voting question open, and if so it will be closed. Then we insert the new voting question being opened and truncate any numbers in the “Numbers” table if there was a previous question with numbers stored.

The same concept is used for storing votes from participants. Below is the code for storing yes votes from voxeo to PHP.
Here, we are getting the callerID and storing the callerID to a variable. Then, we trigger the PHP file and pass the callerID to store the number to the “Numbers” table.

In the screenshot below, in line 14, the phone number has to be inserted in the “Numbers” table to update the “Yes_Votes” count in the “VotingQ” table. The “Phone_Number” in the “Numbers” table is the primary key of the table so it rejects storing the same number hence not updating the “Yes_Votes” count.

```
<?php
$addvauer = 1;
$callerID = $_POST['callerID'];
$username = "ict4dgroup2_VoiceApp";
$password = "?*=^&$@Mn-[]";
$database = "ict4dgroup2_Voice_app";
// Create connection using mysqli_connect function
$conn = mysqli_connect($servername, $username, $password, $database);
// Connection Check
if (!$conn) {
    die("Connection refused: ". $conn->connect_error);
}
$sql1 = "INSERT INTO Numbers (Phone_Number) VALUES (".md5($callerID)."');
if ($conn->query($sql1) === TRUE) {
    $sql2 = "UPDATE VotingQ SET Yes_Votes = (Yes_Votes + $addvauer) WHERE Voting_Status = '0';
} else {
    echo "Error: ". $sql1 . "<br>" . $conn->error;
}
if ($conn->query($sql2) === TRUE) {
    echo "New record created successfully";
} else {
    echo "Error: ". $sql2 . "<br>" . $conn->error;
}
?>
```

**Sequence Diagram of the results**

![Sequence Diagram](image)

The web application follows the same sequence as the voice app with the exception of voting.
5. Pointer to the Application code (Can be a URL or appendix)
https://github.com/Osayd66/VoiceApplication

6. Pointer on how to access the application
When accessing our link, make sure the “No protection” setting is turned off on your browser. To do that, go to Settings on your browser > Privacy and Security > Security > Under Safe Browsing check “No Protection”

Voice Application
For Radio Station (Anna):

End/Open voting question application
Video Link: https://group2.saadittoh.com/End_Start_Voting.mp4

Results Application:
Video Link: https://group2.saadittoh.com/Results.mp4

For Participants (Others):

Yes BipVote:
Video Link: https://group2.saadittoh.com/YesVote.mp4

No BipVote:
Video Link: https://group2.saadittoh.com/NoVote.mp4
For more information, follow the short usage scenario in the Appendices 3 section.

Web Application
https://group2.saadittoh.com/index.php
https://group2.saadittoh.com/index_french.php

7. Short Usage scenario
1. Call the start/end system, select english or french, login, open a voting question
2. Call the yes or no vote
3. Call the results system, select english or french, login and hear results
4. Call the start/end system, select english or french, login and close the question
5. Call the results system again to view the results after it has ended

Number to access ending and starting vote application (For Anna) and Number to vote YES (for other numbers): +31 20 8082848 then 9996122596 then login pin is 8542

Number to access results application (For Anna) and Number to vote NO (for other numbers): +31 20 8082848 then 9991494268 then login pin is 8542
8. Feedback questions.

The radio station:

1. Since the implementation of the voting system, has the application saved time and resources? E.g. counting votes manually, displaying results manually.
2. Since the implementation of the voting system, has the process of handling the application been improved? E.g. extracting results.
3. Since the implementation of the voting system, when talking in an approximate percentage, how many more votes are coming in?
4. Since the implementation of the voting system, how many more voting questions are you able to host in a week?
5. Since the implementation of the voting system, has the application resulted in more listeners?

The votee (listeners):

1. Since the implementation of the voting system, do you perceive the radio broadcast as more interactive?
2. Since the implementation of the voting system, how much more are you voting on a weekly basis?
3. Since the implementation of the new voting system, do you like the fact that the radio stations are able to host more voting questions during broadcasts?

9. Discussion of Scope and Fidelity.

All the must-have and should-have features (refer to the MoSCoW diagram) have been implemented. We aimed for a very useful and functional system while keeping it simple to use and not taxing through resources or cost. We could implement a feature that detects voice inputs but that would require a better voice recognition system which might be costly. Additionally, we could have the radio station store the voting question through the web interface for participants to call and hear the question in case they missed the broadcast, however that would mean that the participants will have to pay the cost of the call to hear the question.

10. Evaluation and reflection of old and new system

To evaluate and reflect on our bip-vote application we compare the old system with our new one.

<table>
<thead>
<tr>
<th>Old system</th>
<th>New system</th>
<th>New system compared to the old (+ positive - negative change)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manually count votes on paper</td>
<td>Automated counting</td>
<td>+ Time efficiency, + no human error</td>
</tr>
<tr>
<td>Manually check duplicates</td>
<td>Automated duplicate checking</td>
<td>+ Time efficiency, + no human error</td>
</tr>
<tr>
<td>Phone nr. is exposed</td>
<td>Phone nr. is encrypted</td>
<td>+ Security</td>
</tr>
<tr>
<td>No real interface</td>
<td>Voice, web interface</td>
<td>+ Availability, + Usability</td>
</tr>
<tr>
<td>Requires GSM phone</td>
<td>Kasadaka, devices with web browser or GSM phone</td>
<td>- More costs</td>
</tr>
<tr>
<td>GSM network, Radio</td>
<td>Local network, GSM network, Radio</td>
<td>+ Flexibility</td>
</tr>
</tbody>
</table>