

ICTD SOLUTIONS FOR DECENTRALIZATION OF GOVERNANCE

*A thesis submitted in partial fulfillment
of the requirements for the degree of*

MASTER OF TECHNOLOGY

in

Computer Science & Engineering

by

SURBHI JAIN

2014MCS2803

PREETI RANI

2014MCS2131

Under the guidance of

Dr. Aaditeshwar Seth



**Department of Computer Science and Engineering,
Indian Institute of Technology Delhi.
January 2016.**

Certificate

This is to certify that the thesis titled **ICTD Solutions for Decentralization of Governance** being submitted by **SURBHI JAIN, PREETI RANI** for the award of **Master of Technology** in Computer Science & Engineering is a record of bona fide work carried out by them under my guidance and supervision at the **Department of Computer Science & Engineering**. The work presented in this thesis has not been submitted elsewhere either in part or full, for the award of any other degree or diploma.

Dr. Aaditeshwar Seth
Department of Computer Science and Engineering
Indian Institute of Technology, Delhi

Abstract

The objective of this project is to give the power of governance in the hands of rural India by the means of technology [4]. When information will be retrieved as well as generated by the local people, it will provide a quick platform to the people in greivance redressal as well as the information generation. Human Access Points (HAPs) of the village will be provided with the mobile applications [5] which acts as a bridge in establishing communication with the local community. HAPs can launch suveys, view active surveys of GramVaani, can broadcast any urgent voice message or text message through the Gramvaai IVR setup. NGOs are provided with web portal to notify HAPs. Web portal provides communication interface between NGOs/ District personnels/ Officials and HAPs. It provides functionalities for admin registration, notifying HAPs, launching surveys, viewing survey reponses and so on.

Acknowledgments

We would like to express our heartiest gratitude to our supervisor Dr. Aaditeshwar Seth for guiding this work with utmost interest, patience, care and scientific rigor. We thank them for setting high standards, giving us freedom to explore multiple facets of the problem and teaching us value of analytical thinking and hard work. We want to express our sincere thanks for providing technical guidance and support during difficult times and whose suggestions went a long way in making this work a reality. We also want to thank Vinod Maurya, Dinesh Kapoor, entire GV Team for their invaluable technical support throughout the project. We would also like to thank our family and friends for their love and support.

**Surbhi Jain
Preeti Rani**

Contents

1	Introduction	1
1.1	Literature Review and Related Work	1
1.2	On Ground Surveys	2
1.3	Survey's Conclusion	4
2	Platform and Technologies	5
2.1	Application Server / Web Server	5
2.2	Application Design	5
3	NGOs' Web Portal	6
3.1	Registration	6
3.2	Functionalities given to NGO People	6
4	Local Governance Application	8
4.1	What is Gologo?	8
4.2	Functionalities	8
4.3	Options on Menu Bar	14
4.4	Usability	15
4.5	Challenges and Suggestions	16
5	Gram Vaani APIs Communication	19
5.1	Launch and View Surveys	19
5.2	Contact Groups Management	20
5.3	Audio Broadcast	20
5.4	Message Broadcast	21

6 Results	22
6.1 Screenshots of the Application Interface	22
6.2 Screenshot of the GramVaani Web Instance	24
6.3 Screenshots of the Web Portal	24
7 Conclusion	39
8 Epilogue	40
8.1 Challenges	40
8.2 Future Work	40
9 Google Cloud Messaging	41
9.1 What is GCM?	41
9.2 Characteristics of GCM	41
9.3 System Architecture	42
9.4 Key Concepts of GCM	42
9.4.1 GCM Components	42
9.4.2 GCM Credentials	43
9.5 LifeCycle Flow	43
Bibliography	45

List of Figures

6.1	Pin Authentication	22
6.2	Message for Pin Authentication	23
6.3	Pin Recovery	24
6.4	Use Cases	25
6.5	Record Audio	26
6.6	Options After Recording Audio	27
6.7	Speak/Type Message	28
6.8	List of Active Surveys	29
6.9	View a Particular Survey	30
6.10	Choose Target Contacts	31
6.11	GV Contact Groups	32
6.12	Phone Contacts	33
6.13	GramVaani Web Instance	34
6.14	GramVaani Soochna Sanchar Home Page	34
6.15	NGO's Login Page	35
6.16	Use Cases of NGO Personnel	35
6.17	Admin's Registration Form	36
6.18	Launch Survey in a District	36
6.19	View Active Surveys	37
6.20	View Active Survey Responses	37
6.21	View Active Survey Questions	38
9.1	GCM System Architecture	42

Chapter 1

Introduction

Information and communication technologies for development in decentralization refers to more better information and communication for further development of a society. ICT4D also requires an understanding of local community development, poverty, agriculture, healthcare, employment and basic education. This makes ICT4D appropriate technology and if it is shared openly open source appropriate technology. Therefore to setup ICTD in a society where people are not equipped with technologies and information communication mediums, there is a need of an application which act as a platform for the human key points of the society which enables them to spread information among local people of the society. Decentralization as explained [7] helps in easy monitoring of the community issues.

1.1 Literature Review and Related Work

The following papers were reviewed while developing the concept.

- Emergent Practices Around CGNet Swara, A Voice Forum for Citizen Journalism in Rural India, ICTD12 [6] : This paper talks about the initiative of CGNet Swara, which is a project similar to JMR active in Chhattisgarh. The authors explain the deployment of the system, and their experiences. It also delves into qualitative and quantitative analysis of the data coming in, of the callers, topic about which stories were reported among other things.
- Designing Mobile Interfaces for Novice and Low-Literacy Users, ACM Transactions on Computer-Human Interaction. 2011 [5]: This study explores different interfaces for low-literacy and novice mobile users. The authors conducted two studies comparing text-based interfaces to other different alternatives such as, one: automatic solutions including

graphics, spoken dialog and text-free user interfaces and second: a live human operator. Based on these studies and interviews conducted with the subjects, the authors cite results regarding the comfort of novice users with the different mobile interface components. They also lay down certain design recommendations while designing mobile user interfaces for such users.

1.2 On Ground Surveys

The following surveys are conducted on ground.

1. **Survey of Munirka Village :** Munirka is an urban area in South West Delhi, located near Jawaharlal Nehru University (JNU) and Indian Institute of Technology Delhi (IIT Delhi) Campuses. Munirka is a village where development has started in early 1990s. The area is mostly dominated by the jaat community. We entered in a dealers shop and asked about the village life, sarpanch and marginalized group in munirka. He himself owned a big shop and carried 2 phones. Moreover, we found that the sarpanch of munirka himself lives in Vasant Kunj and rarely visits his constituency. That was very disappointing part as he was not at all involved in solving the problems faced by his people. He also gave us contact number of vice-sarpanch Bharat Singh who lives in a nearby street in munirka. He said Munirka is no longer a village and the area was well developed where everyone owns a smart phone and living a standard life. All shop owners were well equipped with basic amenities. We also talked to some local shop owners on the main street. The place was still following the village third-tier Panchayati Raj System but the place cant follow the support the model of human access points.
2. **Survey of Vasant Vihar :** Our next visit was towards vasant vihar. Vasant Vihar is an exclusive neighborhood located in the South West Delhi district of National Capital Territory of Delhi. We had a visit to coolie camp situated in the same place. It is a slum area where people

were living in adverse conditions. There was no proper sanitation facilities and no hygienic conditions. More than 3000 jhuggi-jhopdiyas, some pakke makaans were agglomerated in such a small area. The irony is Vasant Vihar is one of the most expensive residential areas in the world (source Wikipedia) and it still has such slum areas.

Well, we asked various questions to the residents of that place regarding the availability of basic amenities. Whether they are able to avail the benefits of various schemes, are able to solve their problems at community level, or by the involvement of the MLA of their area Parmila Tokas. They said for the issues of water and electricity availability, they approach to the MLAs office to put their problems. Sometimes, Officials or people from some department or ministry come to take surveys for various statistics related to literacy rate, population count, number of schools, toiletries etc but problems of local people are not addressed. They take numbers and put it in records. One of the person standing near the retail shop said that they make no efforts after the surveys, just take the figures that too improper and report to higher authorities.

People were reluctant in answering the questions and were uninterested in sharing the stories and experiences they face and encounter.

1.3 Survey's Conclusion

It is analysed from all the surveys that people who live in the underdeveloped areas like rural areas and the slums do not have a platform where they can get all the information about the basic needs which they require in their lifestyle. Information about administration, education, health schemes, employment, agriculture, gas cylinder bookings, land acquisitions can be correctly communicated to them which will help them on fulfilling their needs and eventually develop and making their lifestyle better. But there are certain problems in the implementation of the method.

- People do not have even small knowledge of operating the mobile phones whether basic mobile phones or smart phones. Condition is even worse with women. Though youth and earning member of the house have phones but still they do not know more than dialing and receiving the call thus intense training is required for the proper implementation of the approach.
- People are reluctant in sharing the information to the people outside their community. Or they tell data which is partially or completely false. Thus to collect data through telephonic surveys, Trust needs to be built that it is for their welfare only.
- If started by implementing the approach for all the areas mentioned above, it is unlikely that it will be implemented properly for all the fields and a large dataset of information is required and need to be maintained. Thus the approach should be started by implementing for 1 or 2 areas initially.

Chapter 2

Platform and Technologies

2.1 Application Server / Web Server

- Implemented using Ruby on Rails web framework [3]
- Database is designed using MySQL
- GCM Server

2.2 Application Design

- Android Platform with Android Studio 1.1.0 [2]
- Database is designed using SQLite.
- Using Volley Libraries for asynchronous task communication to the network server [1]
- Android 5.0.1 Minimum SDK build tool and Platform tool with APIs level 21 or above support
- GCM Client

Chapter 3

NGOs' Web Portal

3.1 Registration

The registration of the app users is done through the website by the NGO by filling the required information fields in the registration form. The fields will be the name, age, gender, occupation, address, district, city, mobile-phone number, residence number, username (unique) and password. NGO people will also register themselves through the website and authenticated using NGOs Email Address by sending an automated link on mail just after registration. They will be authenticated as the trusted NGO users.6.17

3.2 Functionalities given to NGO People

1. **Registration of HAPs of a particular village :** They can register the contact number of the admins of a particular village. Registered admins of the village then can use the application for achieving the other functionalities. NGOs will train HAPs of the village by the on-ground training and knowing the feedback from other localite.6.17

2. **Launching of Surveys :** NGOs can launch surveys from the website by selecting from the recent surveys of Gramvaani which will be displayed on the website. Secondly, NGOs can upload their own surveys to be conducted and they can choose the village where they want to conduct the survey so that the admins of those villages will be notified for the surveys through the application. List of surveys will be sent to the android app user as a notification. Admin can then select the survey(one at a time) and the contacts/groups from one or more of the three options described above and he can launch the survey. App

server will send the id of the survey to the GV server along with the contacts. GV server will give the IVR calls to the contacts and collect the responses which will be updated to the admin who launched the survey and on the website for the NGO personnel to view and use the information of the survey. ??

3. **Viewing of Survey results :** Surveys results will be displayed after fetching them from the gramvaani.
4. **Viewing of details of admin :** NGOs are authenticated to control the access permissions of the app users.

Chapter 4

Local Governance Application

4.1 What is Gologo?

Gologo is an android operating system based local governance application which will provide the functionalities to volunteers for sending infotainment news and announcements to the local residents of the community. Users (Volunteers) of the application will be able to broadcast news among localites in an easy and quick way. The system provides platform to send quick audio and message updates to the relevant users. Volunteers will receive alerts regarding ongoing surveys from the authorized block/ district officials (through web portal) on their application to launch surveys in their community. They can also track responses of the people for the surveys which are launched. Consequentially, System helps in delegating governance responsibilities to the community people in an effective and relevant manner.

4.2 Functionalities

The android application which is provided to the human access points/ Volunteers of the society has the following features.

- **One Time Login :** After registration through the web portal [web portal chapter link], volunteer receives a unique pin number of 6 digits on his registered mobile number. After installing the application, user has to do one time login. On first time login in gologo, the volunteer is authenticated for the application with his registered contact number and received PIN. On right credentials, volunteer can directly use all the functionalities as verification will be done only once. 6.1

Also, on first time login. GCM registration on GCM server will be done and user receives a unique registration ID. This registration ID will be

used to send alerts to the user. Also, all requests to the app server involves the authentication through the obtained registration ID. The uniqueness of the registration ID will help in authenticating the user via taking registration ID as key. [GCM chapter link].

- **Broadcast Announcements :** After authentication from the server into the android app, the user can select the option of Broadcast Audio on the home screen. Under this option, the app user gets a recording screen through which he/she can record an audio message or any instant broadcast. Then, contacts picker screen [link of figure of contact options] appears through which he can select the either of the following options :

- Concerned multiple gram vaani groups
- Local contacts saved in phone
- Mobile Vaani callers between a particular duration

After selecting a particular option, he chooses the target people and clicks on send people. After Clicking on send button, a request to the app server is made to send the audio. The message will be then sent to the contacts through Gram Vaani voice calls. Application user will receive an alert message through the GCM notification when message gets played to the target audience. Either of the two options given below can be selected to send the audio

Recorded audio message will be saved in the mobile vaani instance as content so that people can later listen to it when they give calls to the IVR.

- **Survey Dial Outs :** Volunteer acts as a bridge between the NGOs/District officials etc to make the surveys available to the end people. Results will be notified to both the admins through application and NGO personnel through the website.

Volunteer chooses this option to launch the survey in his community. A request will be sent to the application server having the authenticated

gcm id to get a list of ongoing recent surveys from the gram vaani server. Each listed survey will have the following options:

1. View Survey : Volunteer can view the questions of a particular survey. When volunteer selects a survey, the unique survey id along with the authenticated credentials will be sent to the application server. Application user gets a list of text questions.
2. Launch Survey : Volunteer can launch a particular survey in his community via two ways.
 - (a) **On receiving GCM notification** : NGO/ District official are provided with the functionality on the web portal to send gcm notifications to the registered users [link of survey image on portal]. List of surveys will be displayed to the NGO personnel. He will select a particular survey and the location where survey is to launched. Depending upon the selected location, portal displays a list of registered volunteers. He will select the volunteer and click on send which will send a gcm notification alert to the respective volunteer on his application. Application user clicks on notification to launch that survey by sending a list of relevant contacts to the application server.
 - (b) **On pressing launch button on application** : By clicking on launch button, volunteer is provided with the contact options where he will choose the list of contacts from the given options. A request with survey id and list of contacts will be sent to the application server for the dial outs.
3. View Survey responses : This functionality is provided to view the status of launched survey. Volunteer can track the number of people who responded the survey calls. Application also lists the count of responses corresponding to each question. In survey dial outs, it is possible that receiver wont respond to all questions and cut the call in between. In that scenario, volunteer also gets the statistics of responses per question. It will help him in acknowledging the survey status. He can also send the message alerts to

the community regarding survey reminders.

- **Send text Alerts** - Similar to the audio message, the user can send a text message to the selected contacts. The user selects the Send message button from the home screen, selects the contacts/groups from one or more of the three options described above and sends the message which then makes a request to the app server to send the message to the selected contacts through Gramvaani server. Templates are registered with the gram vaani where volunteer will feed only the values related to a particular template. It helps in informing end users instantly regarding the news of surveys, government schemes or future camps nearby the locality area. Following templates are used to send text message alerts to the target people.

1. **Template 1. Announcement of surveys** - Text alert of survey to be launched in the community will be sent with the specified date to inform people regarding the survey launch. Application user selects the date of launching using date picker. Selected value will be fed in the template.

Input Params: date

Message Template (English) : Dear all, a Survey will be launched in your village on date `|date|` by Mobilevaani. Please submit your responses sincerely. Team Mobile Vaani.

Message Template (Hindi) : Priye nivaasiyon, mobile vaani dwara ek sarvekshan dinaank `|date|` ko aapke gaanv mein shuru kiya jaega. Kripaya apni pratikriya jaroор prastut karen. Team Mobile Vaani.

2. **Template 2. Announcement of Camps (with timings)** - Text alert of upcoming camp to be organised in the community will be sent with the specified parameters to inform people. Application user selects the parameters from the drop down and insert the values in edit text date of launching using date picker. Selected value will be fed in the template.

Input Params: campname, startdate, enddate, starttime, endtime, venue

Message Template (English) : `[campname]` Camps will be organized at `[venue]` from date `[startdate]` to `[enddate]` with timings from `[starttime]` to `[endtime]`. Kindly participate in the camp to take its maximum benefits. Team Mobile vaani.

Message Template (Hindi) : Priye nivaasiyon, `[campname]` camp `[venue]` sthaan par dinaank `[startdate]` se `[enddate]` ko samay `[starttime]` se `[endtime]` tak aayojit kiya jaega. Kripaya camp mein bhaag lekar iska adhiktam laabh uthae. Team Mobile vaani.

3. Template 3. Announcement of Camps(without timings)

- Text alert of upcoming camp to be organised in the community will be sent with the specified parameters to inform people. Application user selects the parameters from the drop down options and insert the values in edit text. Selected values will be fed in the template.

Input Params: campname, startdate, enddate, venue

Message Template (English) - `[campname]` Camps will be organized at `[venue]` from date `[startdate]` to `[enddate]`. Kindly participate in the camp to take its maximum benefits. From Team Mobilevaani.

Message Template (Hindi) - Priye nivaasiyon, `[campname]` camp `[venue]` sthaan par dinaank `[startdate]` se `[enddate]` ko aayojit kiya jaega. Kripaya camp mein bhaag lekar iska adhiktam laabh uthaen. Team mobile vaani.

4. Template 4. Announcement of Govt. Schemes -

Input Params: schemename, date, beneficiaryname Message Template (English) - Dear all, Government has launched `[schemename]` Scheme on date `[date]` for `[beneficiaryname]`. For more information, contact the volunteers and take its maximum benefits. Team Mobilevaani.

Message Template (Hindi) - Priye nivaasiyon, sarkaar dwara `[schemename]` yojana dinaank `[date]` ko `[beneficiaryname]` ke lie shuru ki gai hai. Adhik jankari ke lie volunteers se sampark karen aur iska adhiktam laabh le. Team mobile vaani.

- **Add a group** - Volunteers are given this functionality for the management of contact groups. Multiple contact groups are made and managed as per the groups of the community. It helps in easy dissemination of information to the relevant audience. App user will be asked to speak group name where his voice will fill the text view. When user clicks on create button, new contact group with that name will be created in the corresponding application instance. [refer to contact group image] [put sequence diagram of info flow].

For example, if some polio booth camp is to be organised in the community, then only the relevant people who have children of age 3-5 should be informed instead of unnecessary sending dial outs to all people.

Similarly, Survey regarding death due to pregnancy will be sent to homes having women. Dial out will be made to only those contacts/contact groups who falls in this category.

Global surveys can be sent across all groups which are relevant to all people of the community.

- **Add a contact** - The app users can add a new contact. The user selects the option of Add a contact on the home screen of application. In this option, he can fill in all the details of a new contact of his community having name, contact number, gender, date of birth, contact groups and location URI. Volunteer need not to type the name of the contact. He can simply speak the name of the person by clicking on speaker image and it will be converted into text. Phone will be a ten digit mobile number. Gender is a drop down menu containing F and M as the options. Date of Birth can be chosen using date picker. One contact can be added in multiple contact groups. Contact groups are fetched from the gram vaani instance. Application user gets a check list having all existing contact groups where he can select one or more groups to add contact in it. Locations are also fetched from the gram vaani database where each location contains block, district and state. [figure reference]

Volunteer fills all the details from the options into the text boxes and spinners. When he clicks on create, a dialog box asking for confirmation

pops up. He can either click on edit to change any detail or on confirm to add the new contact.

When volunteer clicks on confirm, a request having json data will be sent to the gram vaani server from the application server to update its contacts through APIs. In json data, location URI corresponding to each location will be sent. Also, block, district and state will be extracted from the displayed locations and sent separately. Similarly, ids of the contact groups corresponding to each contact group name will be sent. In a way, json data will be sent to the server in terms of key value pairs. [sequence diagram of add contact]

After confirmation, the app server sends added contact response. Thus, contacts will be synced from and to the Gram vaani server.

4.3 Options on Menu Bar

Some quick options are provided on the application which are available across all activities. Application user can navigate across these options. Following options are explained :

1. **My Profile** - A volunteer can view all his details which are entered when he was registered through the portal. His details contains his user type, name, registered number, date of birth, gender, block, district and state.
2. **Home** - The user can directly go to the home screen from any screen.
3. **Application Information** - This option enlists the usability of each option on home screen with a brief description. It helps the application user to know about the functioning of each option on menu.
4. **Gram Vaani Website** - This option redirects the volunteer to the portal of the gram vaani from the application.
5. **View Recordings** - Volunteer can view all the recordings he did directly from the application.

6. **Share Application** - Volunteer can share the link of .apk file of the application among the people using various options like Whatsapp, Gmail, Facebook etc. It sends the message along with the link to spread the word.

4.4 Usability

Application can be used extensively to achieve various benefits. Volunteers will receive on ground training for the usability of the application. People will be taught regarding its usage and advantages. They will learn how these community people can be benefitted by it and how information can be disseminated among people by using it effectively. It will bring changes among the lives of poor people by equipping them with the tool of information. Their basic phones will be used to respond to the surveys, to view recent news, to receive text alerts, to listen audio announcements, to browse through the content by dialing to the mobile vaani. Following uses are listed below :

1. Volunteers can use the application to receive alerts sent by the NGO/ Block/ District officials.
2. Volunteers can use the application to send text alerts related to any recent news, entertainment shows, ongoing infotainment media etc.
3. Information related to government schemes can be disseminated among the people.
4. Information related to community camps to organise in the society can be broadcasted through text alerts.
5. Instant audio messages to the relevant audience can be sent as dial outs by selecting concerned contact groups.
6. Contact groups can be managed which helps in easy handling of large target community. Groups like Teachers, Farmers, Women, Children, Zamindars, Middle Man, Wholesalers etc can be created and maintained by adding new contacts among them properly.

7. One to one addressing is possible by sending dial outs to individual contact numbers which helps in gathering more responses from the people.
8. Survey responses will increase by separate dial outs which gives more realistic survey results and accordingly, measures will be taken by the government.

4.5 Challenges and Suggestions

The following are some of the challenges identified in the smooth run of the process. The ones listed below are primarily from the point of view of the application and target people where in spite of rigorous training and best interest and efforts of the community representatives, their contribution may be unworthy in helping the community people.

1. Poor internet connectivity:

The internet connectivity in the areas where the app is operated is often not good. Even though many a times there is network available on the phone, internet connectivity speeds can be very low. Due to no internet connectivity or slower speeds, there might be delay in the time when the text updates or dial outs are received at the community level. Also, large audio announcements in MBs will take more time to upload on the server and requires continuous net connectivity. On flaky networks, it will result in repeated failures while uploading. A continuous net connectivity will help in smooth uploading of the audio files. Rest, all functionalities require change of text data which will work in poor connectivity areas or when app user comes in affinity of the internet, data can be exchanged over the network.

2. Long Power Cuts

A number of times, there are long power cuts due to which the Android phones cannot be easily charged again. Imagine a scenario

where a community representative has to delegate some information to the community, but due to inability to charge the phone his phone got switched off before the updates can be sent. In such a case, updates wont be received at the server till the phone is switched on again and gets proper internet connectivity. In order to deal with this, the community representatives were taught about simple steps to manage their battery. For example, keeping the phone brightness low, switching off apps which connect to the internet and regularly charging their phones whenever they have electricity. These are small precautions that can be taken to avoid a situation like above but it can still be a cause of delay in sending information updates or dial outs to the community people.

3. Novice Users

Since the community representatives are novice smartphone users, a rigorous training is mandatory in which dealing with phone problems along with application usability is taught. Problems such as Message Memory Full or Phone Memory Full perplex them and they are not able to handle them. The way the phone functions is that you are not able to access anything on the phone until some memory is freed. In order to counter such situations, basic guidelines were given to the community representatives in written on how to handle such situations. But this can only be done for a few specific common problems. This is necessary as when these community representatives take their phones to a local shop to be checked, at times they delete certain applications. For instance, community representative may not be able to contribute for his community due to these issues in spite of best of intentions.

- **Contacts Selection :** Contacts to which the audio message is to be sent, can be selected in various ways as follows:6.10
 1. Contacts can be fetched from the GramVaani database through API call.
 2. Contacts can be selected from the local phone address book.

3. The audio can be directly sent to the contacts who accessed mobilevaani in the last week/month and similarly other parameters (without fetching them).

For example : The request to directly send the moderated audio to the contacts who called into mobilevaani in the past week/month

- **Send a text message :** Similar to the audio message, the user can send a text message to the selected contacts. The user selects the Send message button from the home screen, selects the contacts/groups from one or more of the three options described above and sends the message which then makes a request to the app server to send the message to the selected contacts through Gramvaani server.6.7
- **Add/Edit a contact** The app users can add a new contact or edit a contact(which is already in the Gramvaani contact list). The user selects the option of Add a contact on the apps home screen. In this option, he can fill in all the information of a new contact which is Name, Contact number, District, Age, Gender and submits the form. The user selects the option of Edit a contact on the apps home screen. In this option he selects a contact from the fetched contact list from GramVaani and can edit any of its field and submits the form. Both of the above options request the GramVaani server to update its contacts through APIs. After updating its contacts, the app server sends Added contact or Edited contact response. Thus, Contacts will be synced from and to the GV contact list. 6.11 6.12
- **Launch Survey :** Surveys to be conducted (as per NGO selects- explained later) will be listed on the tab Launch Survey where admin can select the survey to be launched along with the specified users or all phone users of the village. Admin acts as a bridge between the NGOs/District officials etc to make the surveys available to the end people. Results will be notified to both the admins through application and NGO personnel through the website. 6.8 6.9

Chapter 5

Gram Vaani APIs Communication

Following GV APIs are used to achieve the functionalities given to the end users and NGO personnel.

5.1 Launch and View Surveys

- Get Active Surveys : To get the surveys of a particular app_instance
API URL : /survey_survey/
- Get Survey Questions : To get the form of a survey, which contains the questions.
API URL : /form_question/
- Get Responses : To get the responses of a survey.
API URL : /survey_record/cdr_records/
- Create Questions : To create questions for a particular form. The responses to the questions can be of three type VoiceMCQQuantitative
API URL : /form_question/add/
- Upload Audio : To upload audio prompts, which can be used later as a questionprompt
- Create Survey : After creating the form_questions, the survey needed to be saved. This will make the survey active by default.
API URL : /survey_survey/save/

5.2 Contact Groups Management

- Get Contact ListsGroups : Fetches all the contact groups (lists) maintained by the user.
API URL : /callerinfo_contact_list/
- Get Contacts : To get the contacts of a particular contact list
API URL : /callerinfo_contact/
- Create New Contact List : To create a new contact list in GV instance
API URL : /callerinfo_contact_list/
- Create Contact in Single or Multiple Contact Groups : To create a new contact and adding it to multiple contact_lists
API URL : /callerinfo_contact/

5.3 Audio Broadcast

- Create New Item : It helps in creating a new item on the GV server. Creating an item instance on mnews returns instance id which will be used to upload audio on the GV server.
API URL : mnews_news /create_new_item/
- Upload Audio : MNEWS API will be called to upload an audio and make it as a news item's recording.
API URL : mnews_newsupload_audio
- Uploading on Default Channel : To upload audio by getting the default channel of the app instance.
API URL : /media_prompt_audio/upload_audio/

5.4 Message Broadcast

- Sending Message : Message will be sent in the form of template with various template parameters along with the target contact list.

API URL : sms_message/send/

- Check Message Status : sms_message with GET request along with app instance ID can be used to check the status of message delivery for a particular caller ID.

API URL : /vapp/api/v1/sms_message/id

Note - Base URL : <http://internal.gramvaani.org:8080/vapp/api/v1>

Chapter 6

Results

6.1 Screenshots of the Application Interface

1. First time app Users will be authenticated via PIN received through the message on NGO Registration.

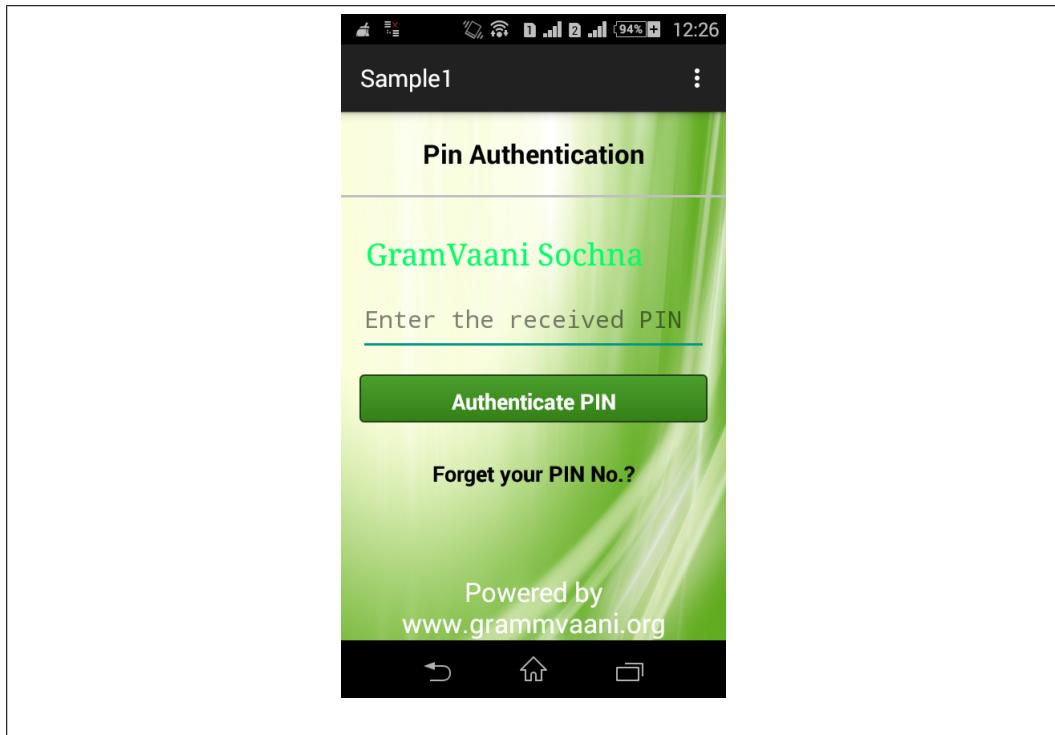


Figure 6.1: Pin Authentication

2. Online User Registration by the NGO sends a message on registered number for authentication.
3. User can retrieve PIN by clicking on Forget PIN option.
4. User will be provided with the following options.

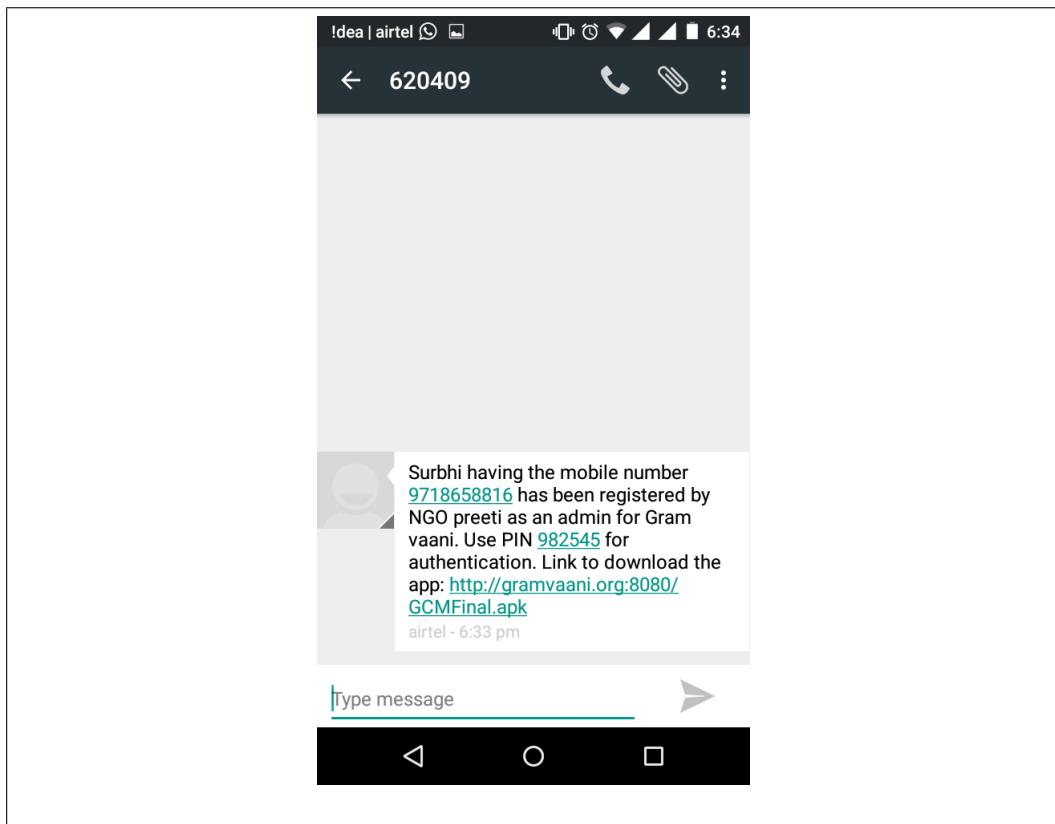


Figure 6.2: Message for Pin Authentication

5. User can record audio by choosing audio format.
6. User can choose to either save or discard an audio message.
7. User can speak a message or type the message to send.
8. User can view all the active survey names of the GramVaani Server.
9. User can view a particular survey with all survey questions on selection.
10. User can choose the target people among his phone callers, GV callers, Mobile Vaani Recent Callers and GV groups.
11. Multiple GV contact groups can be chosen as the target people.
12. Multiple phone contacts can be chosen as the target people.

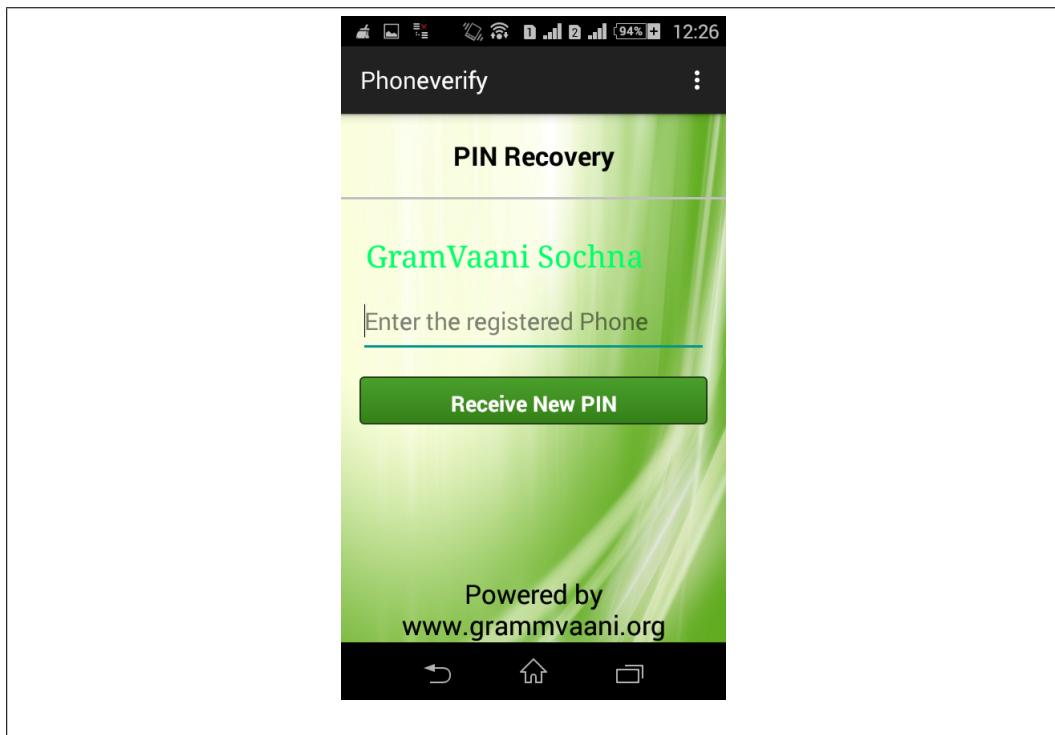


Figure 6.3: Pin Recovery

6.2 Screenshot of the GramVaani Web Instance

1. GramVaani Graphical Web Portal Instance

6.3 Screenshots of the Web Portal

1. Home Page of the Web Portal for NGO's Registration and other functionalities
2. NGO's Login Page
3. Use cases provided to the NGO users
4. Web Online form for Admins Registration and Authentication

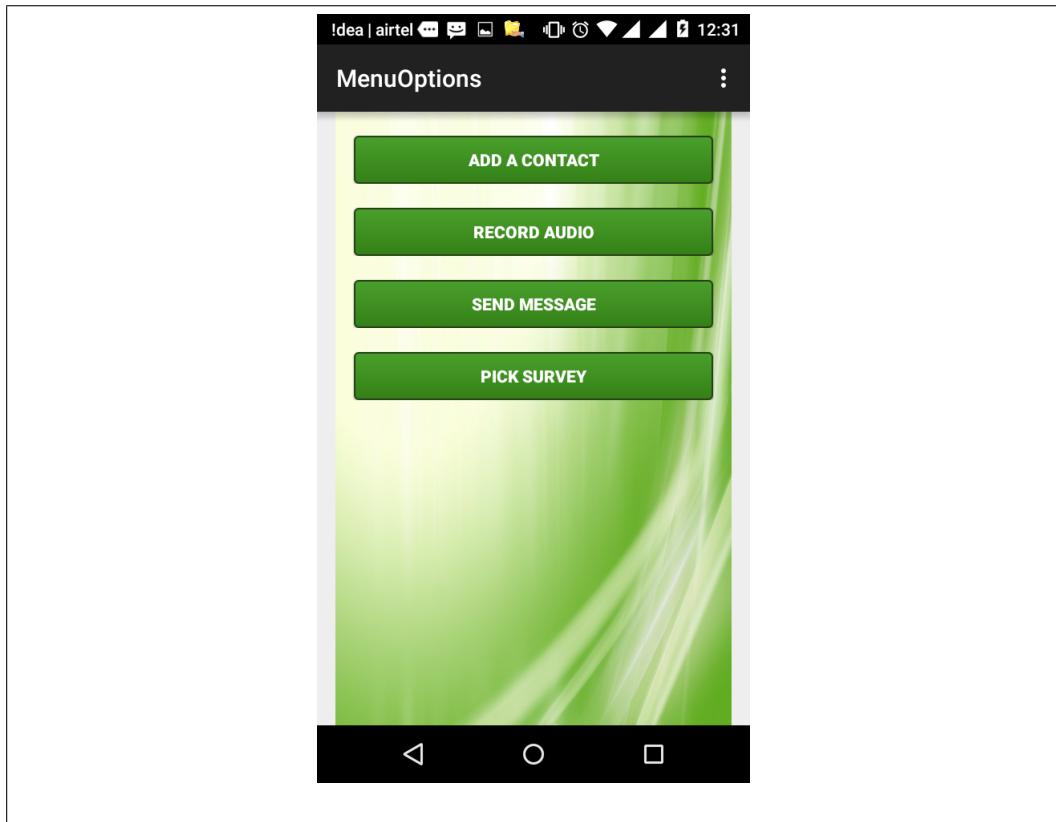


Figure 6.4: Use Cases

5. NGO Personnel can launch a particular survey for a particular district.
6. NGO Personnel can view all the active surveys along with viewing current responses and survey questions.
7. NGO Personnel can view the responses of the active survey.
8. NGO Personnel can view the text questions of the active survey.

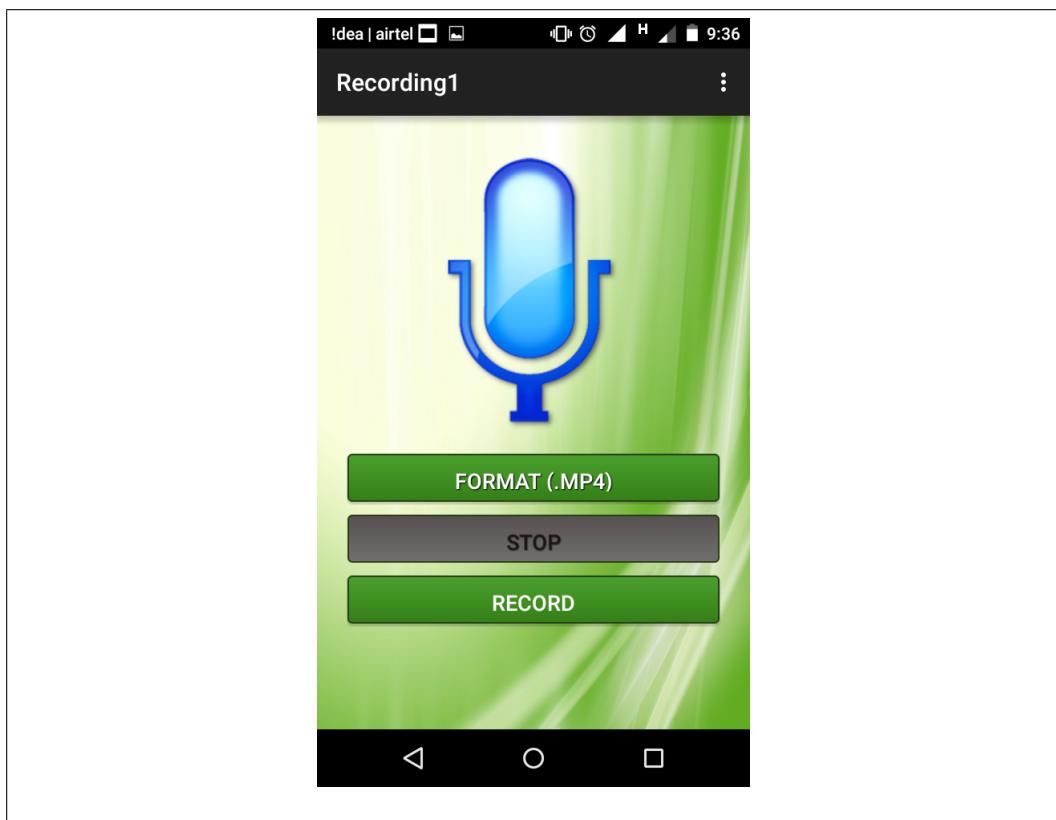


Figure 6.5: Record Audio

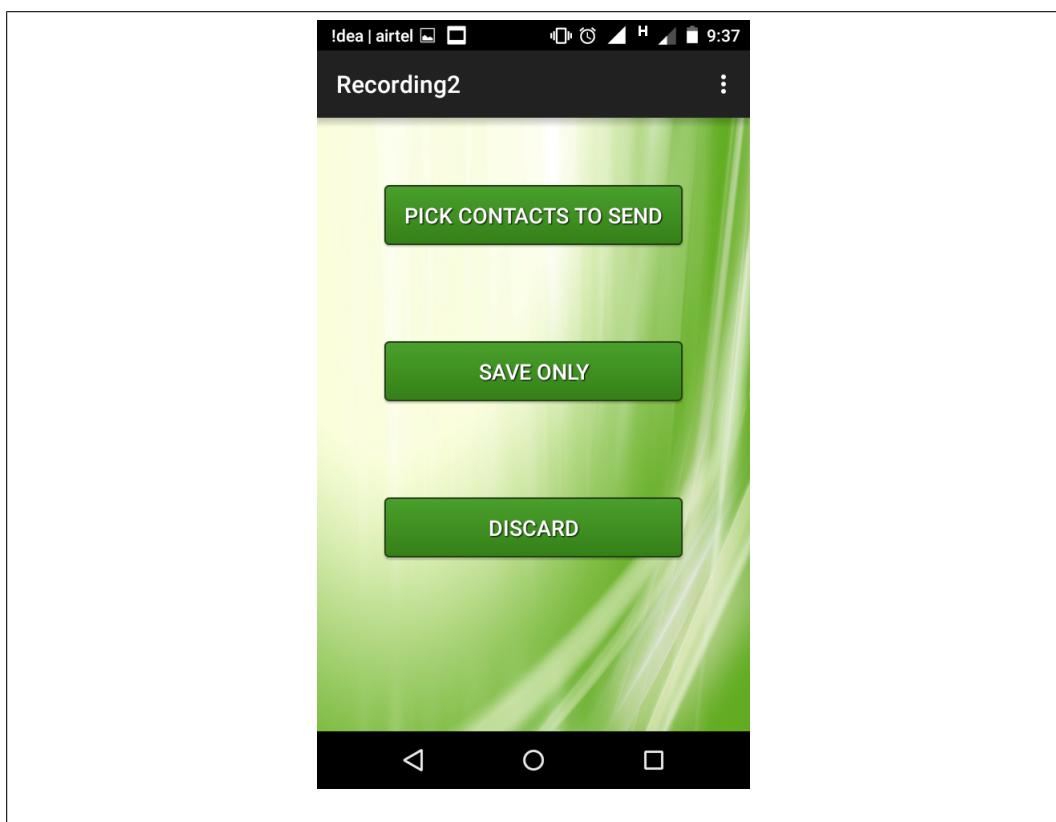


Figure 6.6: Options After Recording Audio

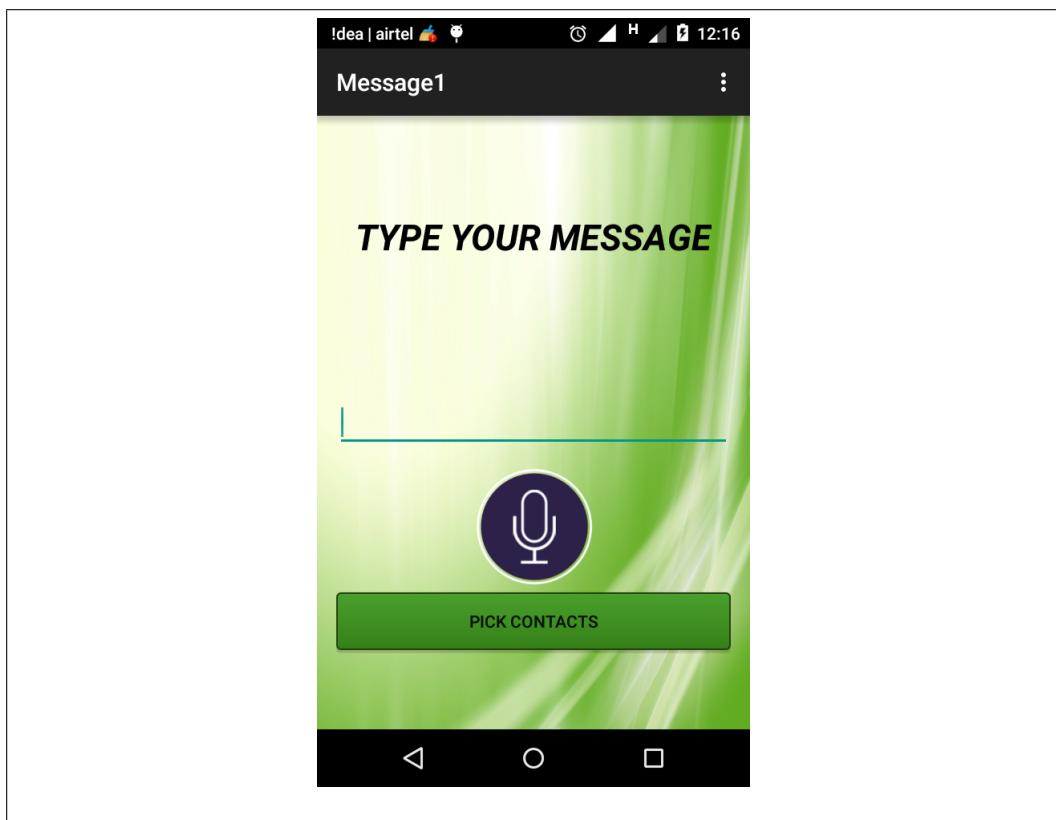


Figure 6.7: Speak/Type Message

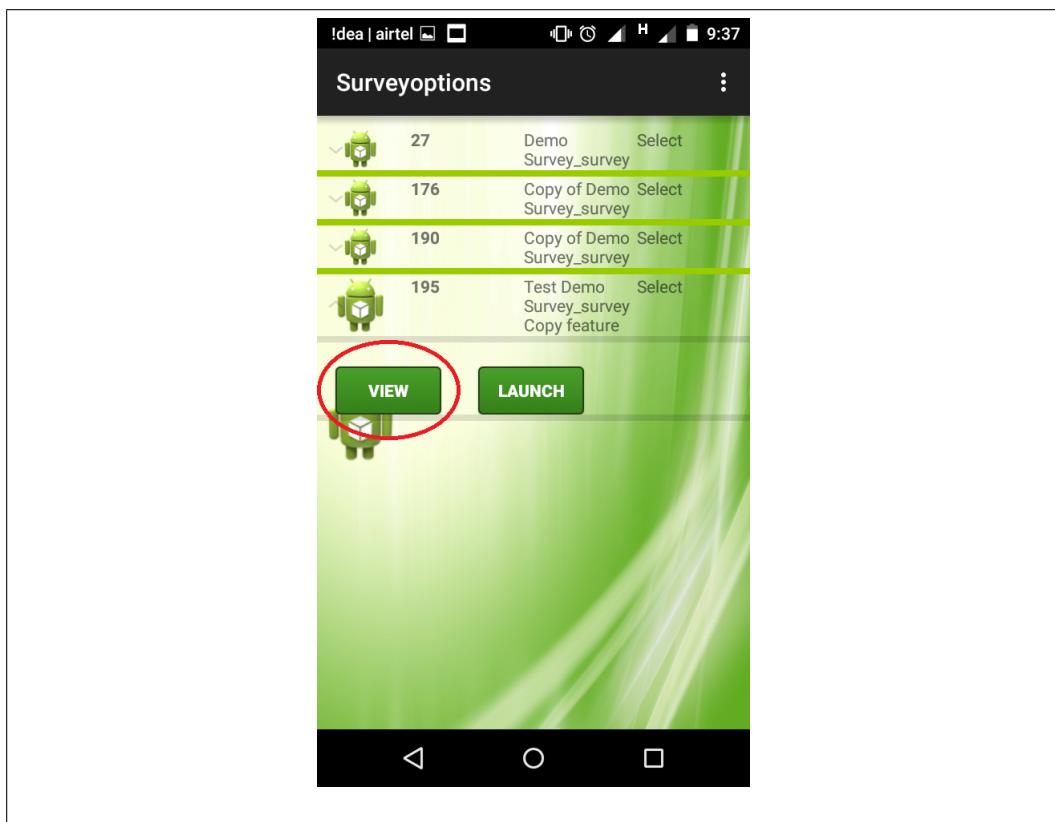


Figure 6.8: List of Active Surveys

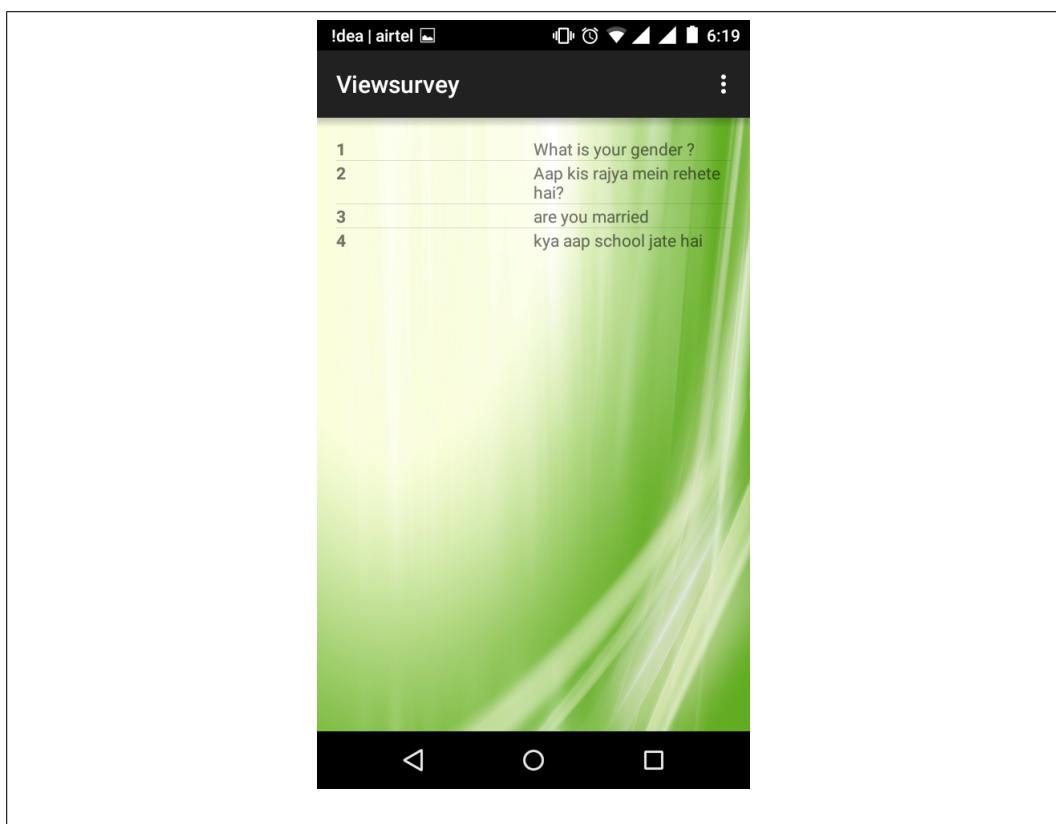


Figure 6.9: View a Particular Survey

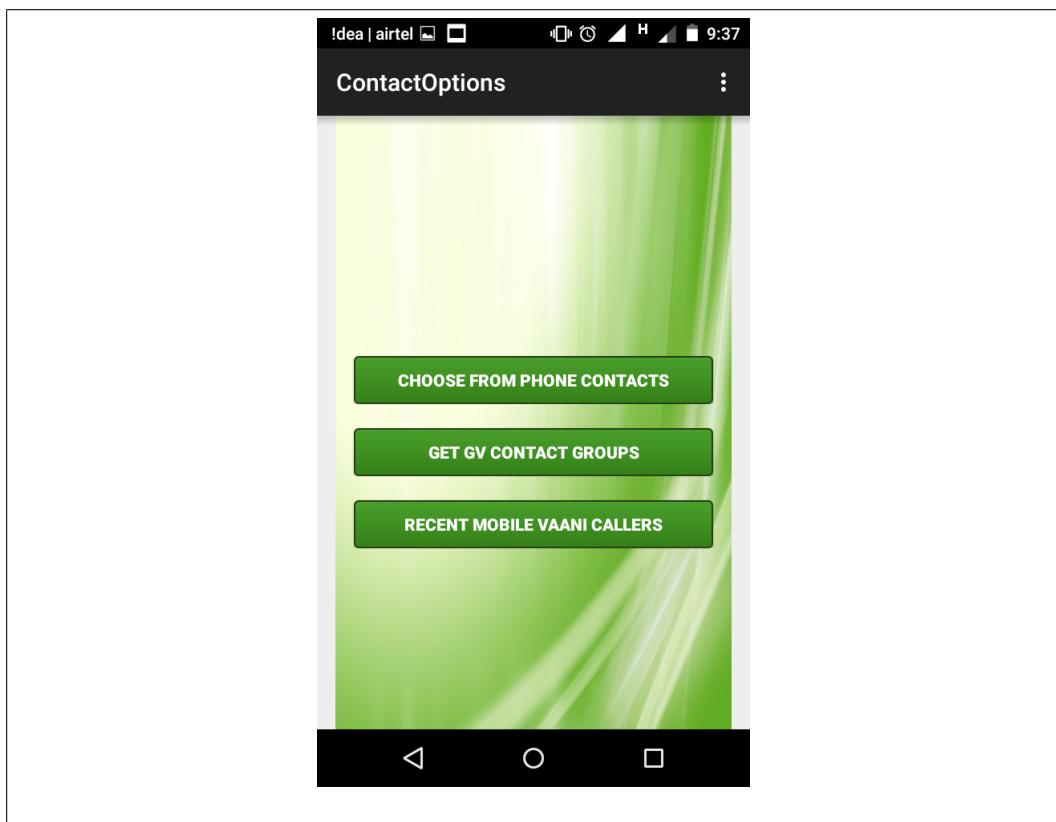


Figure 6.10: Choose Target Contacts

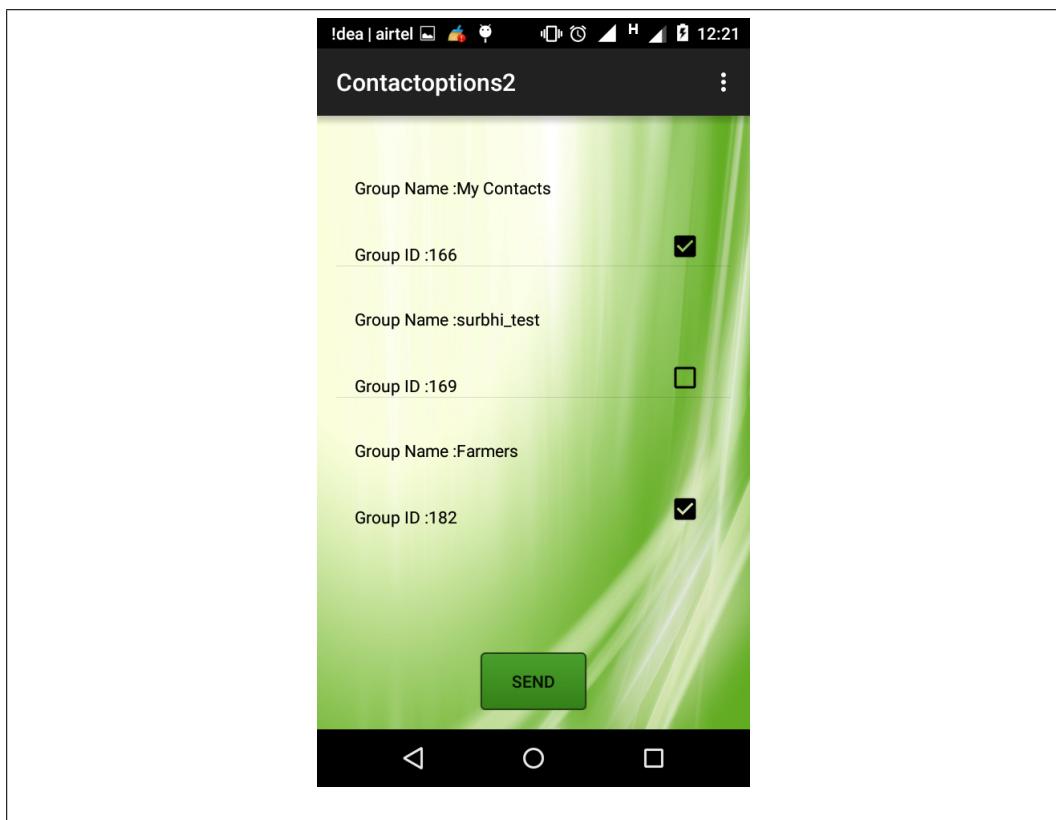


Figure 6.11: GV Contact Groups

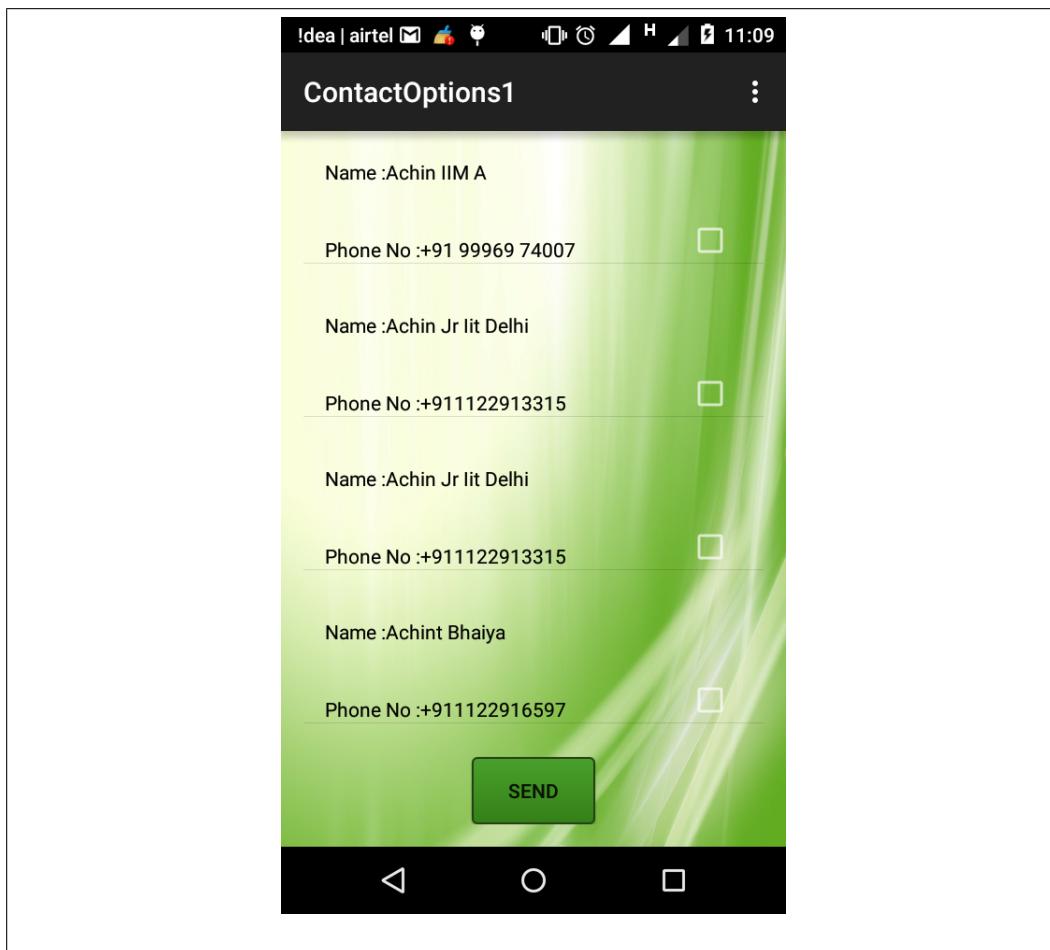


Figure 6.12: Phone Contacts

6.3 Screenshots of the Web Portal

34

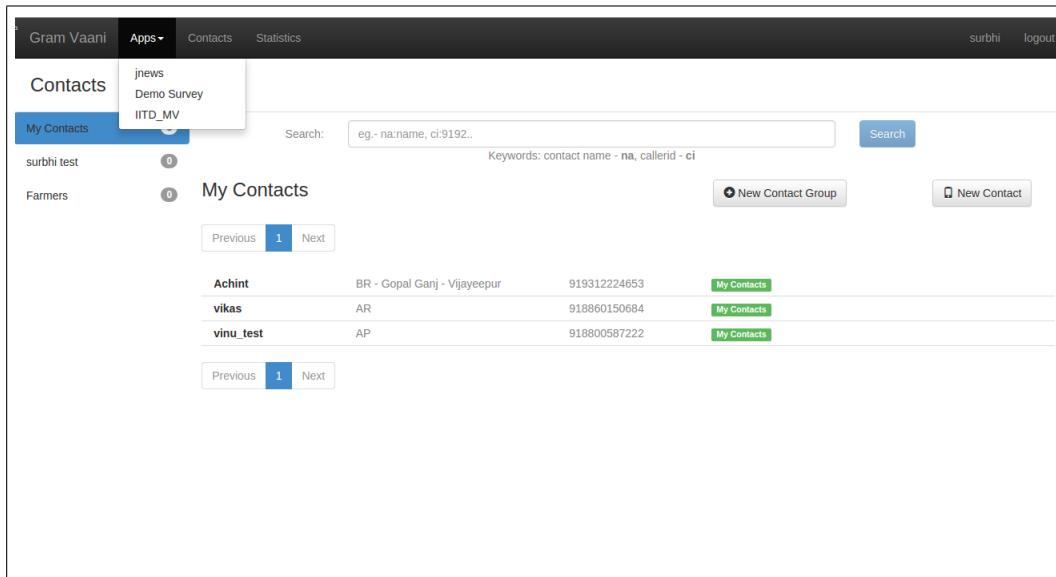


Figure 6.13: GramVaani Web Instance

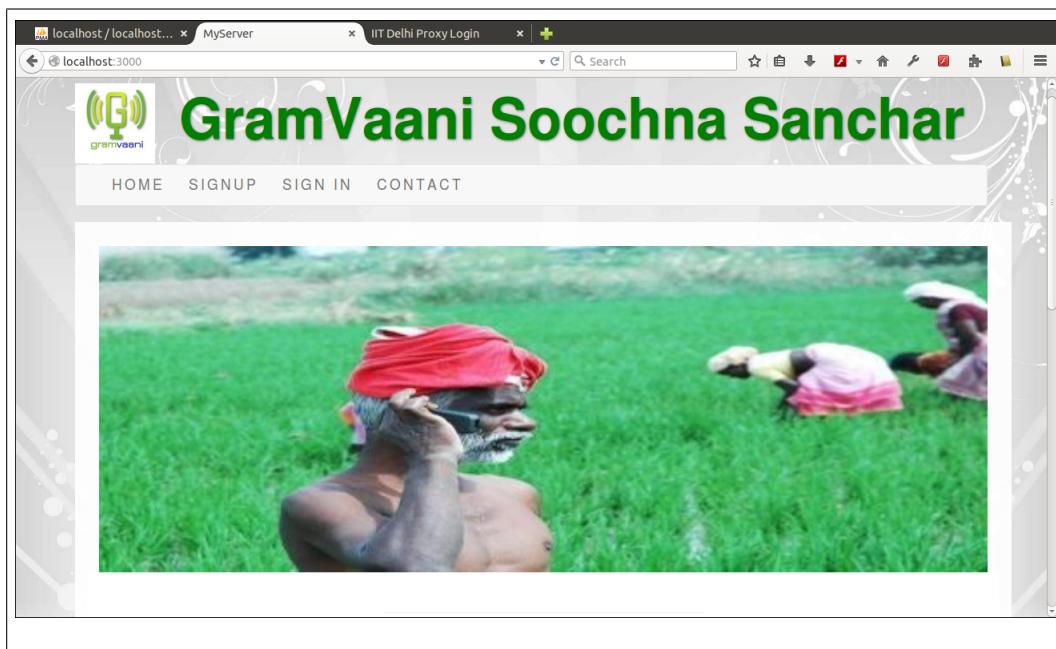


Figure 6.14: GramVaani Soochna Sanchar Home Page

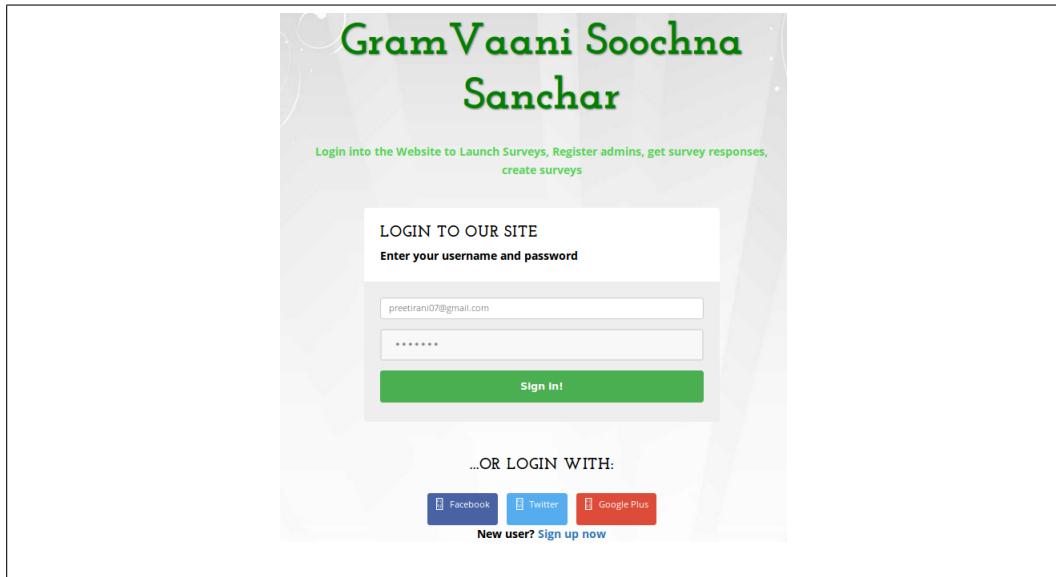


Figure 6.15: NGO's Login Page



Figure 6.16: Use Cases of NGO Personnel

GramVaani Soochna Sanchar

Register admin to make them use to android app to launch the surveys, audios, messages, create contacts in th Gramvaani instance of the local users

REGISTER ADMIN FORM

Enter Details

Name	<input type="text"/>
Contact	<input type="text"/>
Gender	<input type="text"/>
Age	<input type="text"/>
city	<input type="text"/>
State	<input type="text"/>
District	<input type="text"/>

[Save Admin](#) [Back](#)

Figure 6.17: Admin's Registration Form

Launch surveys

Select District
Bokaro

SELECT	FORM ID	SURVEY ID	NAME
<input checked="" type="checkbox"/>	70	27	Demo Survey_survey
<input type="checkbox"/>	349	176	Copy of Demo Survey_survey
<input checked="" type="checkbox"/>	367	190	Copy of Demo Survey_survey
<input type="checkbox"/>	372	195	Test Demo Survey_survey Copy feature

[Launch Survey](#)

```
--> !ruby/hash:ActionController::Parameters
      controller: survey
      action: create
```

Edit: Controller | View (47.156ms) | Partials (0) | Stylesheets (0) | Javascripts (0)
Show: Assigns (1) | Session (4) | Cookies (2) | Params (2) | Filters | Routes | Env | Queries (0) DB (0.000ms) | Log (0)

Figure 6.18: Launch Survey in a District

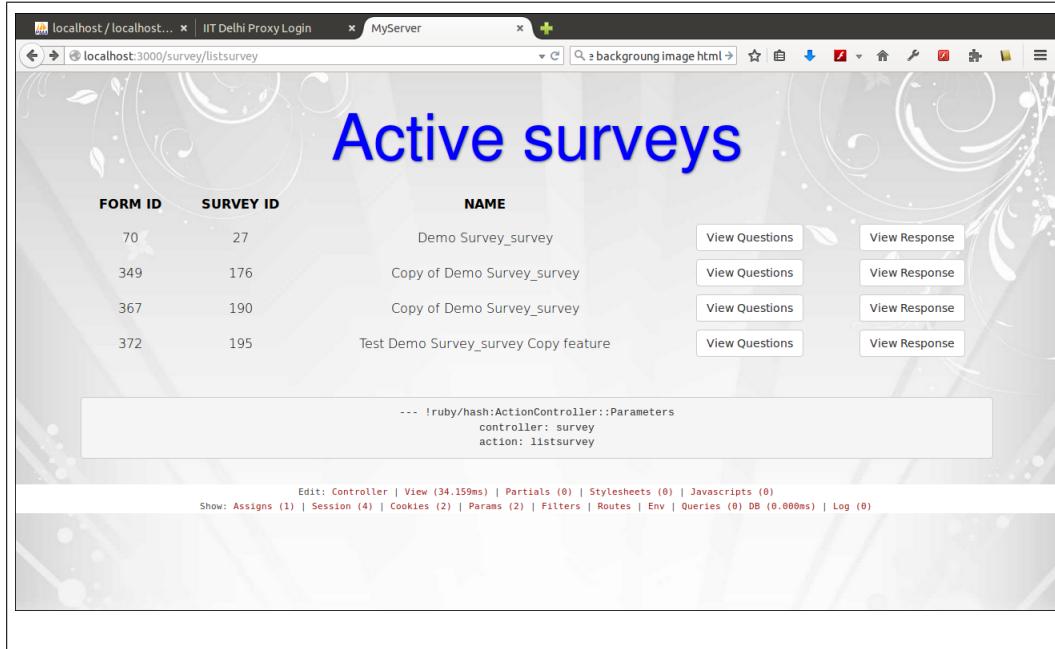


Figure 6.19: View Active Surveys

The screenshot shows a web browser window titled 'MyServer' displaying a list of survey responses. The title 'Survey Responses' is prominently displayed at the top. Below it is a table with columns: CALLER ID, CDR ID, ID, QUESTION ID, SURVEY NAME, and TIME. Each row represents a survey response with its corresponding details.

SURVEY ID = 27					
CALLER ID	CDR ID	ID	QUESTION ID	SURVEY NAME	TIME
918800587222	2198667	141413	83	Demo Survey	2015-12-31T18:28:28
918800587222	2198667	141414	51	Demo Survey	2015-12-31T18:29:53
918800587222	2198667	141415	77	Demo Survey	2015-12-31T18:30:26
918800587222	2198644	141405	83	Demo Survey	2015-12-21T18:46:28
918800587222	2198644	141406	51	Demo Survey	2015-12-21T18:47:53
918800587222	2198644	141407	77	Demo Survey	2015-12-21T18:48:26
918800587222	2198644	141408	1417	Demo Survey	2015-12-21T18:48:43
918800587222	2198636	141390	83	Demo Survey	2015-12-16T14:19:37
918800587222	2198636	141391	51	Demo Survey	2015-12-16T14:19:45
918800587222	2198636	141397	1421	Demo Survey	2015-12-16T14:21:24
918800587222	2198636	141396	1420	Demo Survey	2015-12-16T14:21:11

Figure 6.20: View Active Survey Responses

Survey Questions		
FORM ID = 70		
TEXT	TYPE	OTHER
What is your monthly house hold income	Multiple Choice	Choices 10,000, 10,000-20,000, 20,000
What do you think about this voice demo	Voice Reponse	Max Duration 60
What is your age ?	Quantitative	Max Digits 9
are you married	Multiple Choice	Choices yes, no
are you married	Multiple Choice	Choices yes, no
"are you married"	Multiple Choice	Choices yes, no
new question	Multiple Choice	Choices yes, no
new question	Multiple Choice	Choices 1,2,3,4
"how old are you"	Multiple Choice	Choices yes,no
"how old are you"	Voice Reponse	Max Duration 10
"how old are you"	Quantitative	Max Digits 10

Figure 6.21: View Active Survey Questions

Chapter 7

Conclusion

1. On ground training is mandatory before launching any scheme, giving any benefits, introducing ICTD media among people, deploying any technology.
2. Manual intervention and involvements are the key elements in introducing big changes and turning heads of the people.
3. The local knowledge of village is very important prior introducing any new model in that place.
4. Necessity of responsible people in various regulatory authorities, commission departments, panchayats, Government officers, NGOs workers, ASHA workers, school teachers.
5. People should themselves come forward to seek solutions and seek information and registering complaints.
6. Mobile phones users are many and they can be given on ground training for making the human access points and local villagers known with the problem and the technologies.
7. Main issues and problems are specific to the villages, to the regions. They need to be identified and then application can be used and make a great contribution in actually helping people by various means.
8. For each department, there are separate commissions and agencies working under them, they can be directly put in link with the problem.

Chapter 8

Epilogue

8.1 Challenges

Deployment may face the below listed challenges.

- Multi-Lingual support
- Designing in coherence with end user capabilities
- Capturing user requirements
- Content Management and Content Moderation
- Unawareness of HAPs regarding community problems.
- Data Security
- Scalability

8.2 Future Work

- Introducing flexibility in user interface by minimizing user input
- Sending push notifications and current status of scheduled task to app users through GV call-back APIs.
- User Feedback and Real time Application Deployment
- Dispatching information among NGO personnel and Application users

Chapter 9

Google Cloud Messaging

9.1 What is GCM?

Google Cloud Messaging (GCM) is a free service that helps Android developers to send data from servers to their Android applications, and upstream messages back to the cloud from the users device. This can be a lightweight message telling the Android app that there is a new data to be fetched from the server or it can be a message containing up to 4kb of payload data. The GCM service handles all the aspects of queuing of messages and delivery to the target Android application running on the target device.

9.2 Characteristics of GCM

1. Allows 3rd-party application servers to send messages.
2. Using GCM Cloud Connection Server, one can receive upstream messages from the users device.
3. Android application doesnt need to be running on a device to receive messages. When the message arrives, system will wake up the Android application via Intent broadcast, as long as the application is set up with the proper broadcast receiver and permissions. Gologo uses WakefulBroadcastReceiver to awake the device when notofication arrives.
4. Built-in user interface or other handling for message data is not available, GCM simply passes raw message data straight to the Android application that has full control of how to handle it. For instance, survey alerts are sent and handled at the application side.
5. Requires devices running Android 2.2 or higher with Google Play Store app installed or an emulator running Android 2.2 with Google APIs.

9.3 System Architecture

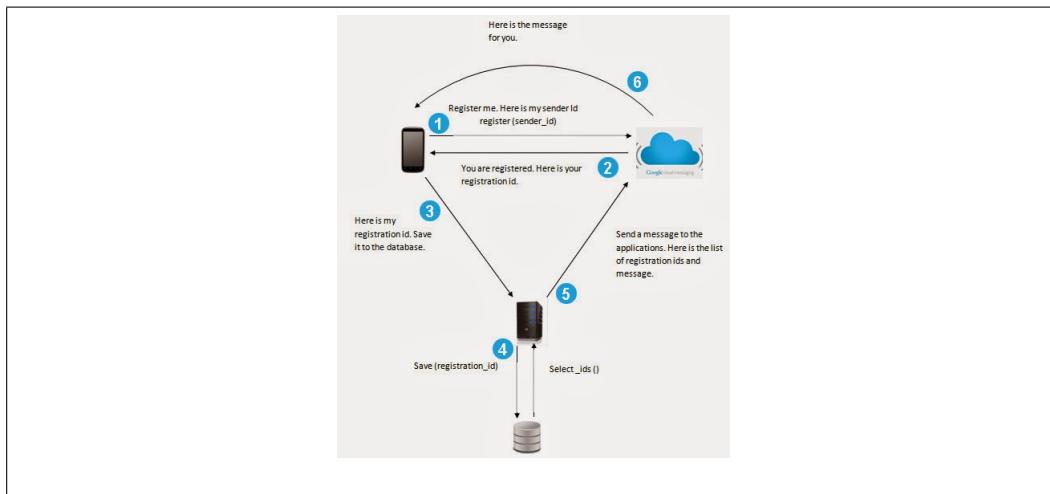


Figure 9.1: GCM System Architecture

9.4 Key Concepts of GCM

9.4.1 GCM Components

- Client App** It is a GCM-enabled Android application running on a device. This must be 2.2+ Android OS device with Google Play Store installed, and it must have at least one logged in Google account if the device is running a version lower than Android 4.0.4.
- 3rd party Application Server** An application server that you write as part of implementing GCM. The 3rd-party application server sends data via the GCM connection server to Android application on the device.
- GCM Connection Server** These are Google-provided servers involved in taking messages from the 3rd-party application server and sending them to the device.

9.4.2 GCM Credentials

1. **Sender ID** The sender ID is used in the registration process to identify a 3rd-party application server that is permitted to send messages to the device.
2. **Application ID** The Android application that is registering to receive messages.
3. **Registration ID** An ID issued by the GCM servers to the Android application that allows it to receive messages. Once the Android application has the registration ID, it sends it to the 3rd-party application server, which uses it to identify each device that has registered to receive messages for a given Android application.
4. **Sender Auth Token** An API key that is saved on the 3rd-party application server that gives the application server authorized access to Google services. The API key is included in the header of POST requests that send messages.

9.5 LifeCycle Flow

1. **Enable GCM** Android application running on a mobile device registers to receive messages.
 2. **Send A Message** A 3rd-party application server sends messages to the device.
 3. **Receive A Message** Android application receives a message from GCM server.
-
1. **Enable GCM** To use the messaging service on Android application for the first time, it needs to call the GoogleCloudMessaging method register(). The register() method returns a registration ID which should be stored by our Android application for later use.

2. **Send A Message :** Here is the sequence of events that occurs when the application server sends a message. Message is sent to GCM servers by the application server. Google enqueues and stores the message in-case the device is offline. When the device is online, Google sends the message to the device. On the device, the system broadcasts the message to the specified Android application via Intent broadcast with proper permissions, so that only the targeted Android application gets the message. This wakes the Android application up. The Android application does not need to be running beforehand to receive the message. The Android application processes the message. If the Android application is doing non-trivial processing, you may want to grab a PowerManager WakeLock and do any processing in a service.
3. **Receive A Message :** This is the sequence of events that occurs when an Android application installed on a mobile device receives a message. The system receives the incoming message and extracts the raw key/value pairs from the message payload, if any. The system passes the key/value pairs to the targeted Android application in a com.google.android.c2dm.intent.RECEIVE Intent as a set of extras. The Android application extracts the raw data from the com.google.android.c2dm.intent.RECEIVE Intent by key and processes the data.

Bibliography

- [1] Android working with volley library. <http://www.androidhive.info/2014/05/android-working-with-volley-library-1/>.
- [2] Introduction to android development with android studio - tutorial. <http://www.vogella.com/tutorials/Android/article.html>. (Visited on 01/05/2016).
- [3] Ruby on rails guides. <http://guides.rubyonrails.org/>.
- [4] G Shabbir Cheema and Dennis A Rondinelli. *Decentralizing governance: emerging concepts and practices*. Brookings Institution Press, 2007.
- [5] Indrani Medhi, Somani Patnaik, Emma Brunskill, SN Gautama, William Thies, and Kentaro Toyama. Designing mobile interfaces for novice and low-literacy users. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 18(1):2, 2011.
- [6] Preeti Mudliar, Jonathan Donner, and William Thies. Emergent practices around cgnets swara, voice forum for citizen journalism in rural india. In *Proceedings of the Fifth International Conference on Information and Communication Technologies and Development*, pages 159–168. ACM, 2012.
- [7] Diego D Navarra and Tony Cornford. Ict, innovation and public management: Governance, models and alternatives for egovernment infrastructures. *ECIS 2005 Proceedings*, page 121, 2005.