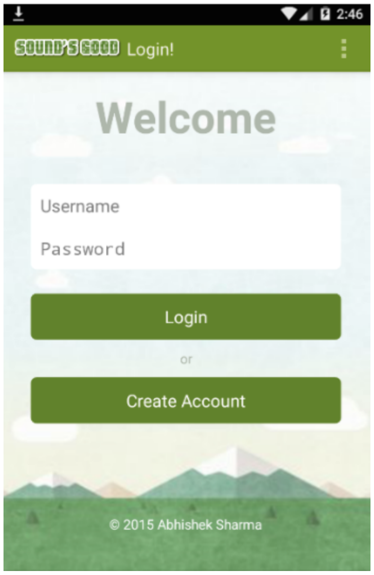
SOUND’S GOOD



It is an online social networking app. First of its type to be completely audio driven and controlled. It allows users to leave an audio comment on various posts and pictures. In this fast world, this application feature is very feasible. The whole dilemma of typing a comment will be bygone, all one has to do is speak, record and share or comment.

Objective of the project: The proposed app will be called ‘Sound’s Good’ and will be published to Play Store and after which anyone can make use of it. All users will be free to create account in this app and once the user is registered he/she can roam around the app and make use of it. Like Facebook or twitter, a user can post status, picture or an audio status. To which his friends can reply or share their thoughts using an audio comment.

Technical Specifications

Software - Will be developed on Android Studio supported by Mac OS X Yosemite
Hardware - All Hardware supporting Android.

JAVA RESOURCES

• ParseConnect.java

This is our Application class. As we have initialize Parse API on application creation, we need to create our own Application class named “ParseConnect.java” extending android.app.Application class. Here we have to specify the app key and client key, which we can take from our App Parse Account. We have to add a couple of permissions as well to the manifest.

We have to download a Parse SDK , add Parse library to our app. Add couple of dependencies to the gradle file and we are good to go. This java file basically provides an interface between our app and backend cloud service by parse.

- **LoginActivity.java**

This is the basic Login class where we ask user to enter Username and Password. Once authenticated user is taken to the Homescreen of our app. We have raised an intent to call the HomepageAcitivity. User can even select the menu option Reset Password, to do the obvious.

- **RegisterAcitivity.java**

Here User can signup for the app. User has to enter a Username, E-Mail, Password, Likes and Hate. We have created a new Parse user to save this information. We have to save this data in the background. user.signUpInBackground(new SignUpCallback()) method is allowed to save data on the cloud.
On our Parse App we see that, a User table is created with the specified info by the user. This table uses an objectId for each user. It also maintains a record of Email verification and createdAt which specifies the time account was created.

- **ResetPassword.java**

This class is used to carry a reset password activity for the user. User has to enter his/her registered Email info. And a link is sent to the user. The new RequestPasswordResetCallback() is a Parse service which is automated. It verifies and stores the new password to our Parse database automatically.

- **HomepageActivity.java**

This is the first screen of our app. This activity extends a ListActivity, because we want our page to appear accordingly. Here we have taken a List<ParseObject> mStatus, which appears in a list form for the basic layout. We created a Custom layout to make sure how exactly each status will appear with a logo on left, username, status.
ParseQuery<ParseObject> query is used over the Status table from parse which maiantains all the statuses from users. We use query.orderByDescending("createdAt"); so that recent status appears on the top. FindCallback<ParseObject>() method is used to query in the background. We make use of a custom StatusAdapter to fetch Statuses.
Author: Abhishek Sharma CS 9033 Android: Spring 2015
We have to click a status to move to StatusDetailView so we have also crated an onListItemClick method which takes us to StatusDetailView.

- **ProfileActivity.java**

Here we display current user name and email. We use getCurrentUser to get the current user. We have two options to upload a Profile Picture using picker from gallery or calling a camera activity. Intent.createChooser is used to pick image from gallery and new Intent (MediaStore.ACTION_IMAGE_CAPTURE) to call the camera from our phone. I am still working on saving this picture to Parse.

- **UdateStatusActivity.java**
Here we update a text status and put it to the HomeScreen List view. A method saveInBackground(new SaveCallback()) is used to save it in background. We also handle exceptions here by providing an alert dialog, if something goes wrong.

- **StatusAdapter.java**
This is our custom adapter to fetch data from the backend and display over our application. It extends a ArrayAdapter<ParseObject>. We have used a Parse query adapter for the other Audio Activity. We have taken the homepagecustomlayout for this adapter. A viewholder holds the username and status. We get the string user and string newStatus store it to String. We then fetch this to viewholder to display username and status on Homescreen.

- **StatusDetailView.java**
We have used ParseQuery<ParseObject> query = new ParseQuery<ParseObject>("Status"); to query the Status table. A query.getInBackground is used to fetch something from the background. A new GetCallback<ParseObject> fetches a particular data with the query. We retrieve the username, status and time createdAt to display.

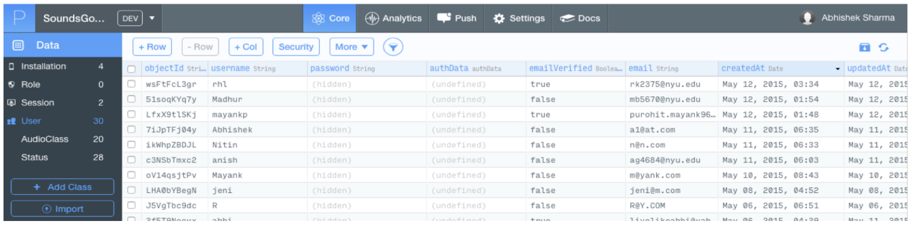
- **AudioActivity.java** This is basically the most important class of our application. Most of the functionality is based
on the tutorial about audio capture I presented in the class for the advance topic.
To quickly summarize we make use of a mediaRecorder object(). We specify various methods such as setAudioSource, setOutputFormat and setOutputFile for the obvious reasons. We have specified a dialog box, which prompts user to record a audio comment using a button. There is a 20s timer for the user to record. We have start and stop function. As soon as we click on Save button a Parsefile named “audiofile.mp3” to the current user. We don’t have to worry about the name as it is saved with a different objectId everytime. We have to convert this audio file into a Byte Stream because we cannot store it directly to the cloud.
We can click to the Goto Audio Wall button to view audio comments by various users.

- **AudioFilesList.java**
This is a simple a view for our audio wall. We have included a ListView to display all the recorded audio comments. Similar to the text status we use AudioFilelist_Adapter to fetch our audio comments. The only difference being it is a ParseQueryAdapter.

- **AudioFilelist_Adapter.java**
This is a ParseQueryAdapter which queries on the AudioClass table. This also specifies how this file will be displayed. There are two parts of this class. On the first we provide a custom look to our list view ordered by descending order.
On the second part we use a Media Player object. We make use of a Parse Object to fetch the audio file. We use a parse file to access the play command once the SOUNDBUZZ button is clicked. We also fetch and display Username and createdAt with this audio file.

- **Other Details:** There are various menu options in these files and toasts at some activities, I have not explained them in details here.

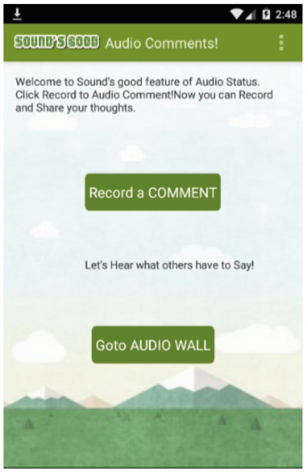
On our app we have used the push service, email verification, password-resetting services from Parse. On our Parse backend we have Session table for maintaining session, a User table for user data, an Audio Class table for audio comment record and Status table for text statuses.



Class	Count
Installation	4
Role	0
Session	2
User	30
AudioClass	20
Status	28

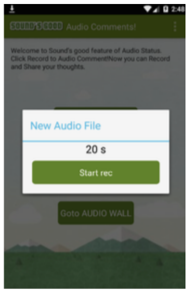
objectId Stri...	username String	password String	authData authData	emailVerified Boolea...	email String	createdAt Date	updatedAt Date
wsFtFcL3gr	rhl	(hidden)	(undefined)	true	rk2375@nyu.edu	May 12, 2015, 03:34	May 12, 2015
51soqKYq7y	Madhur	(hidden)	(undefined)	false	mb5670@nyu.edu	May 12, 2015, 01:54	May 12, 2015
LfxX9tlSKj	mayankp	(hidden)	(undefined)	true	purohit.mayank96...	May 12, 2015, 01:48	May 12, 2015
7iJpTFj04y	Abhishek	(hidden)	(undefined)	false	a1@at.com	May 11, 2015, 06:35	May 11, 2015
ikWhpZBDJL	Nitin	(hidden)	(undefined)	false	n@n.com	May 11, 2015, 06:33	May 11, 2015
c3NSbTmxc2	anish	(hidden)	(undefined)	false	ag4684@nyu.edu	May 11, 2015, 06:03	May 11, 2015
oV14qsjtPv	Mayank	(hidden)	(undefined)	false	m@yank.com	May 10, 2015, 08:43	May 10, 2015
LHA0bYBegN	jeni	(hidden)	(undefined)	false	jeni@m.com	May 08, 2015, 04:52	May 08, 2015
J5VgTbc9dc	R	(hidden)	(undefined)	false	R@Y.COM	May 06, 2015, 06:51	May 06, 2015
[illegible]	abhi	(hidden)	(undefined)	true	livelikeabhi@yah	May 06, 2015, 04:39	May 11, 2015

App Screen Shots

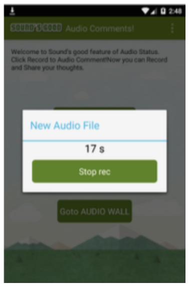


Welcome to Sound's good feature of Audio Status. Click Record to Audio Comment!Now you can Record and Share your thoughts.

Let's Hear what others have to Say!



Welcome to Sound's good feature of Audio Status. Click Record to Audio Comment!Now you can Record and Share your thoughts.



Welcome to Sound's good feature of Audio Status. Click Record to Audio Comment!Now you can Record and Share your thoughts.