



Application Development for Mobile and Ubiquitous Computing

Seminar Task Final Presentation

Group 17

Team: Gonçalo Mordido José Venâncio

The only way to **make music** with other people is:

- By knowing other people that can play other instruments.
- And then meeting them face-to-face.



Make music, together.



Depuração USB ligada

Swoop



Make music, together.

Username

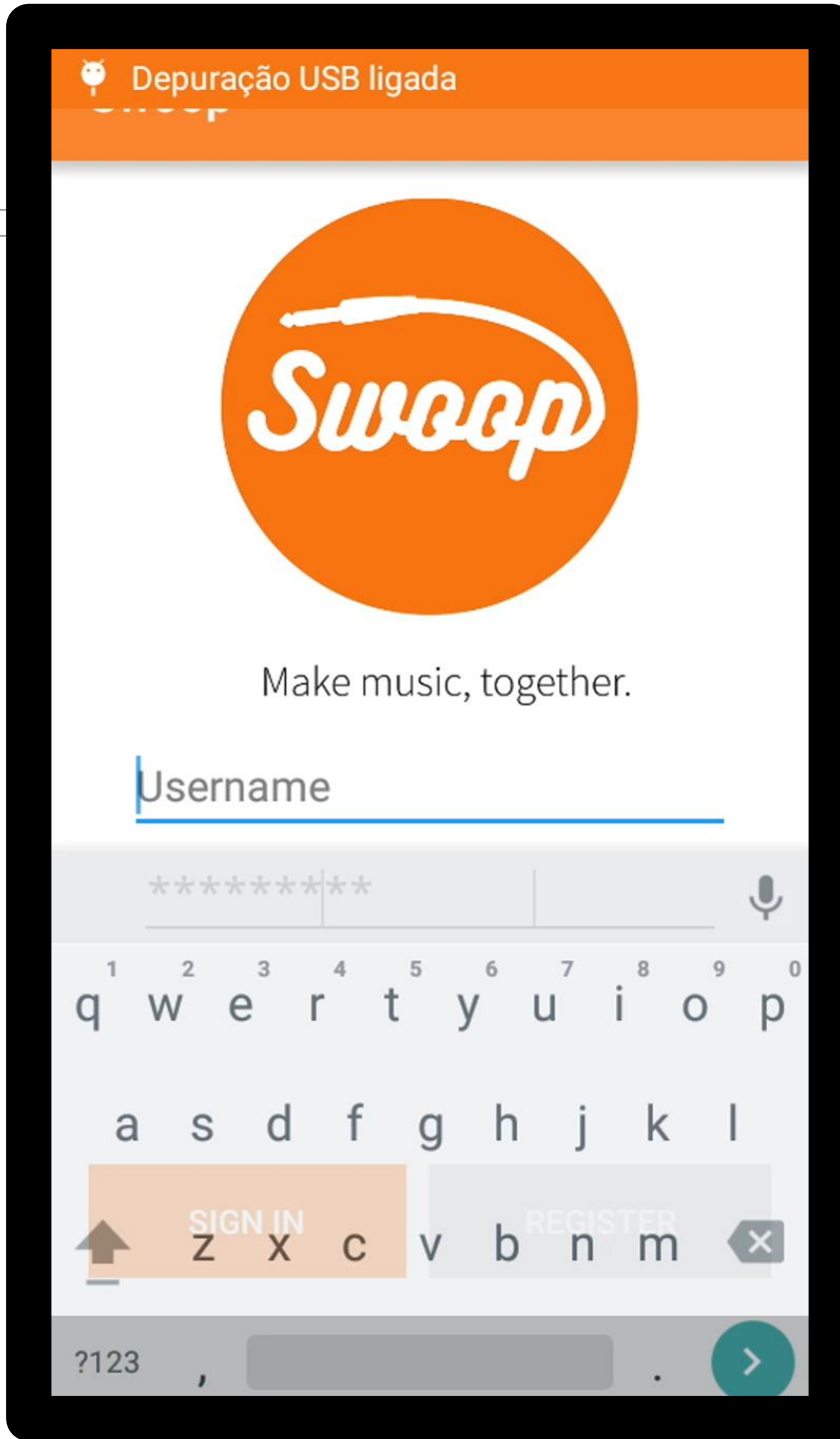
SIGN IN

REGISTER

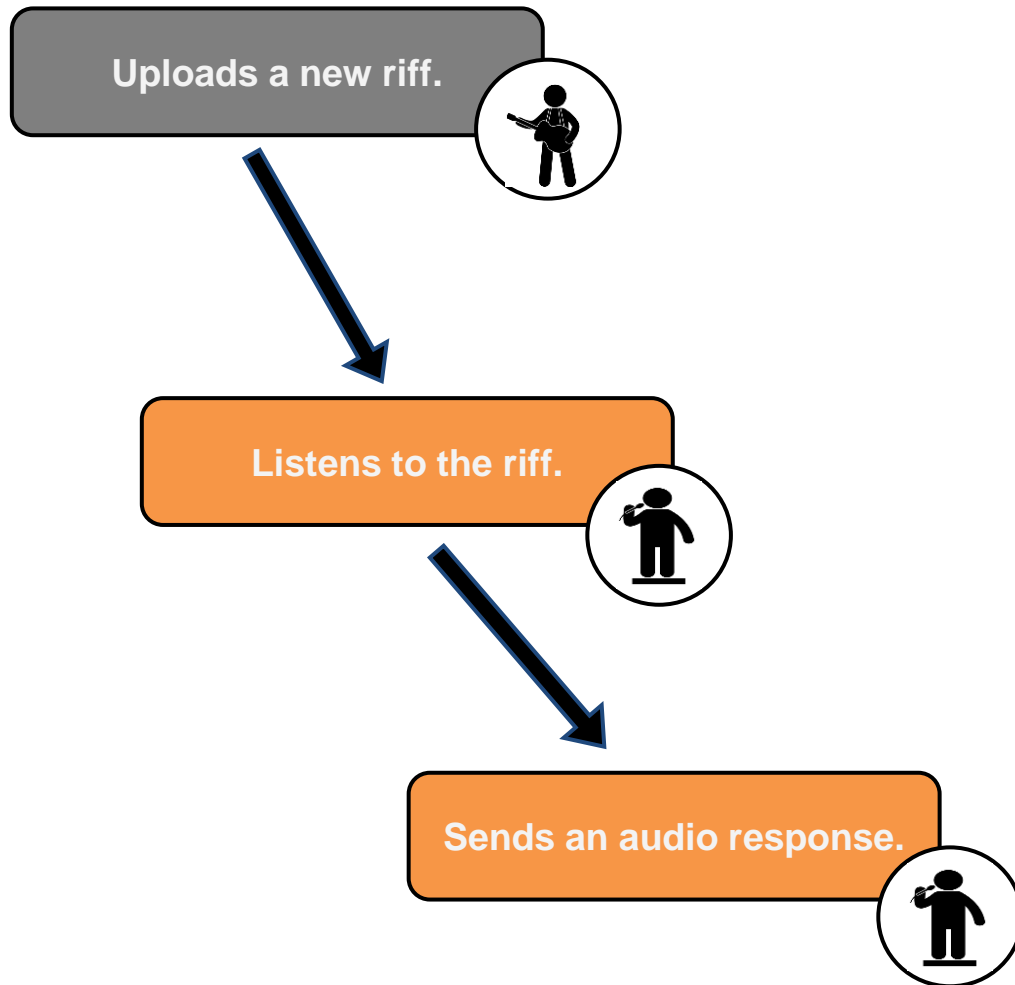
Register

Uploads a new riff.





Uploading a riff





Depuração USB ligada

Swoop

MY RIFFS

SHARED RIFFS

Riff Name

by gmordido • 1h ago



Samba

by gmordido • 10h ago



New Song

by gmordido • 2d ago



fcabrita

2d ago



jmvenancio

2d ago



laura

2d ago



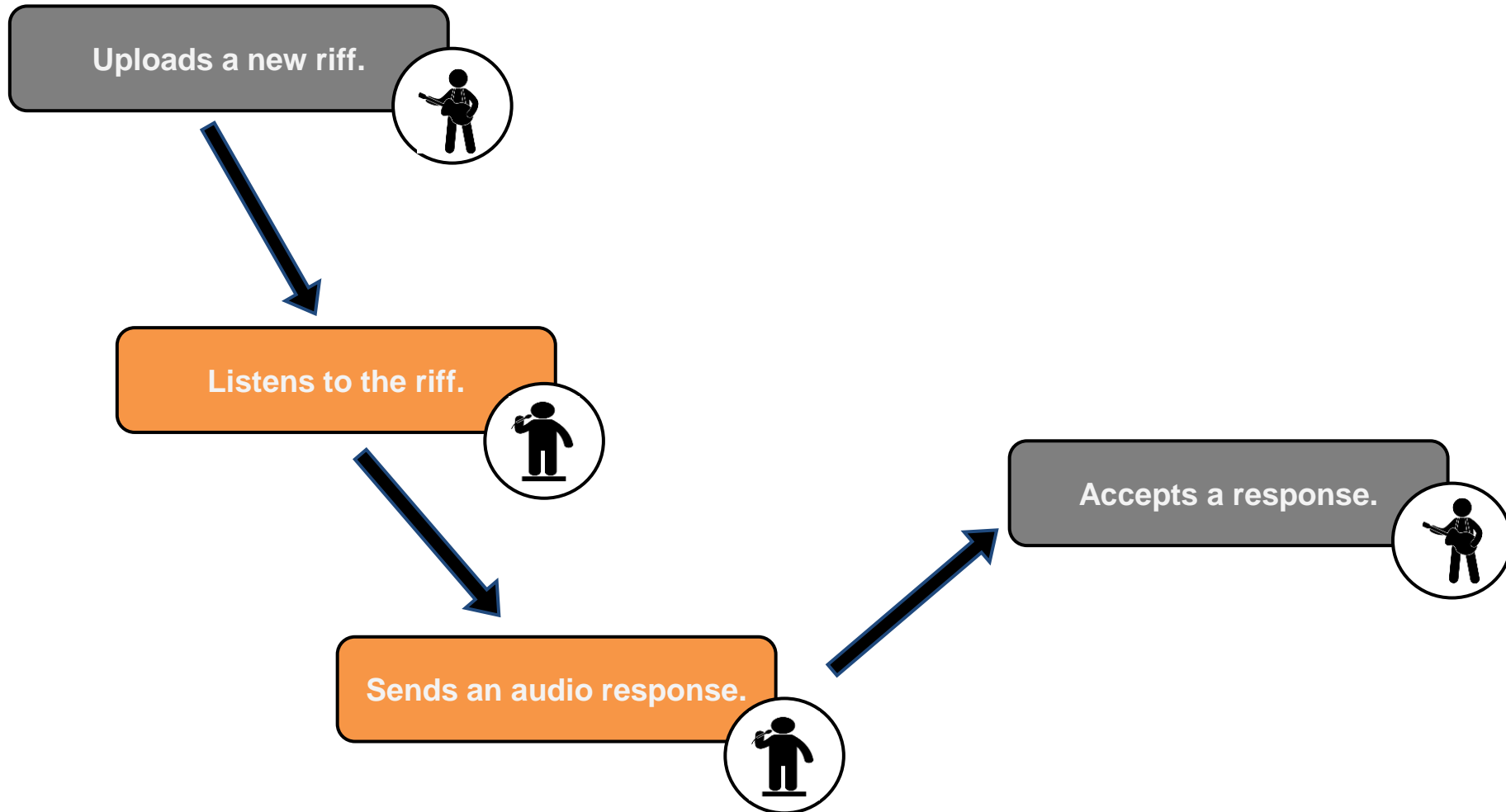
Rosie

by gmordido • 2d ago



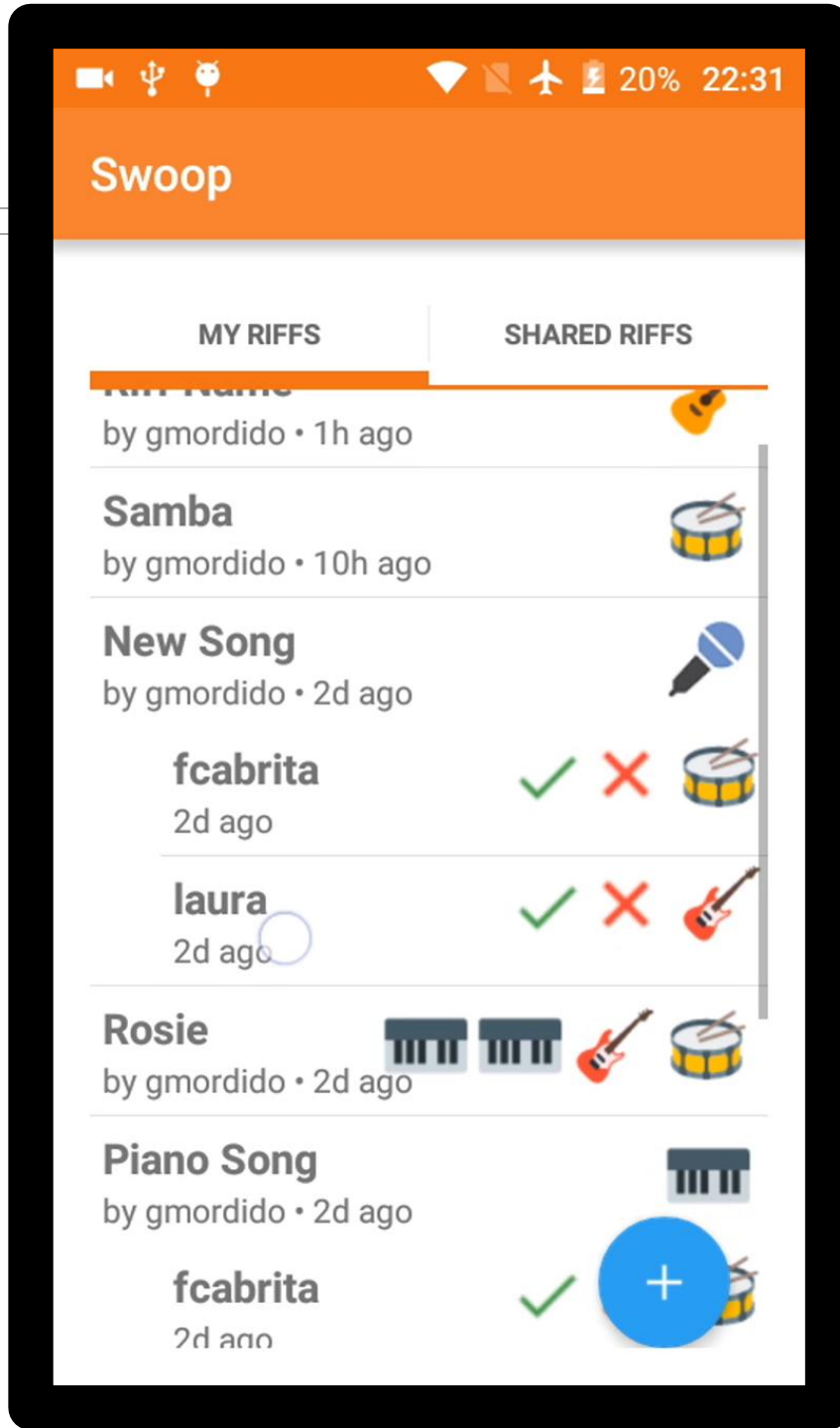
Piano Song

Listening and responding to a riff





Accepting a response



- Offline Challenge
provide some functionality when offline.
- Context:
Detect if a user has internet connection.
- Adaptation:
While submitting a riff or a response offline, store the audio file in cache.
When with internet connection again, the user can simply re-send the riff without the need to record it again.



🐱 Depuração USB ligada

Swoop

MY RIFFS

SHARED RIFFS

Riff Name

by gmordido • 1h ago



Samba

by gmordido • 11h ago



New Song

by gmordido • 2d ago



fcabrita

2d ago



laura

2d ago



Rosie

by gmordido • 2d ago



Piano Song

by gmordido • 2d ago



imvenancio

Offline Challenge

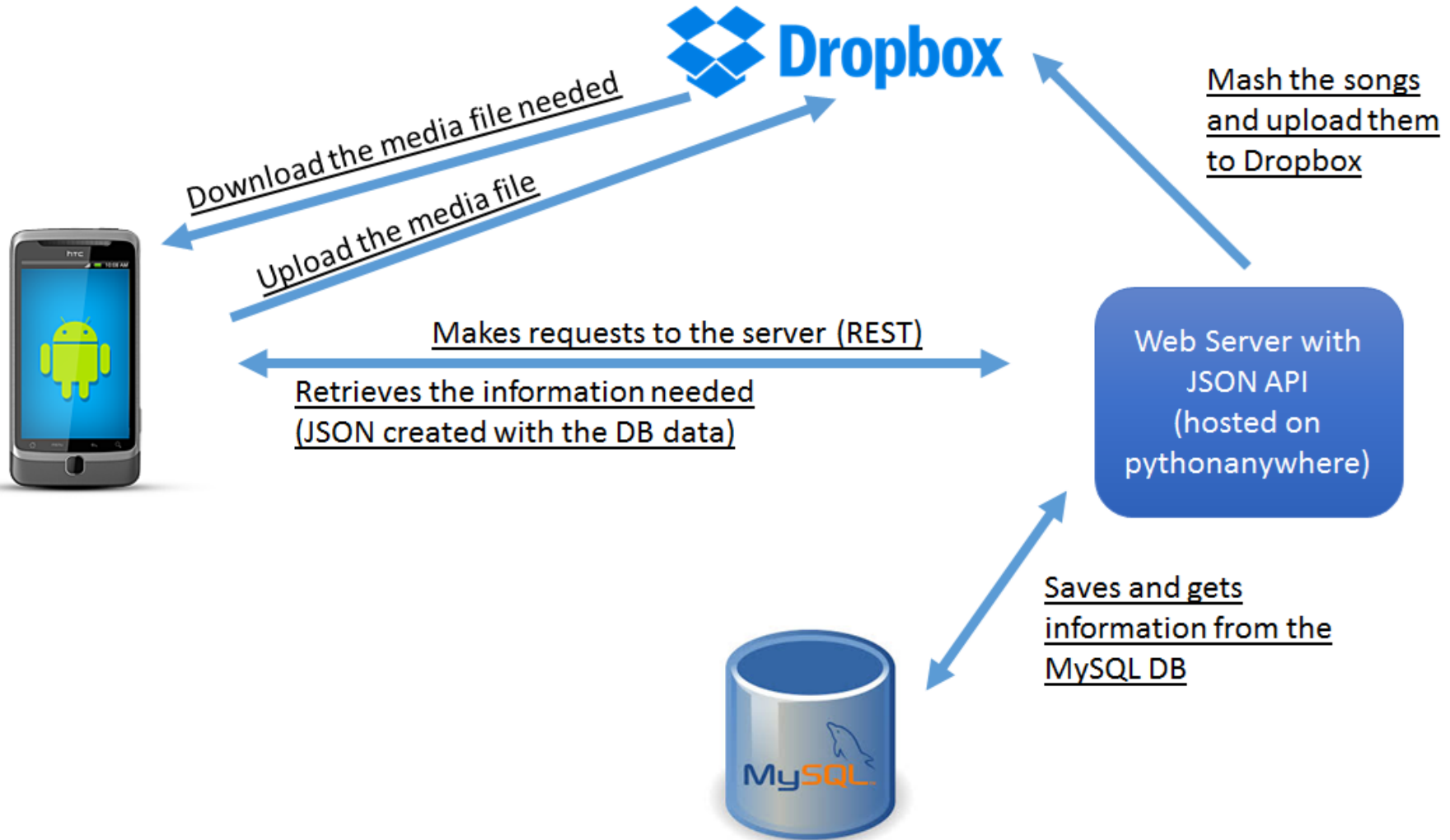
- Usability challenge
 - Changes the audio output of the application depending on the context.
- Context:
 - Detects if a user has headphones connected to the device.
- Adaptation:
 - If there are no headphones plugged in, the audio retrieved by the application goes only through the earpiece speaker, and not the overall speakers of the phone (placed near the mic).

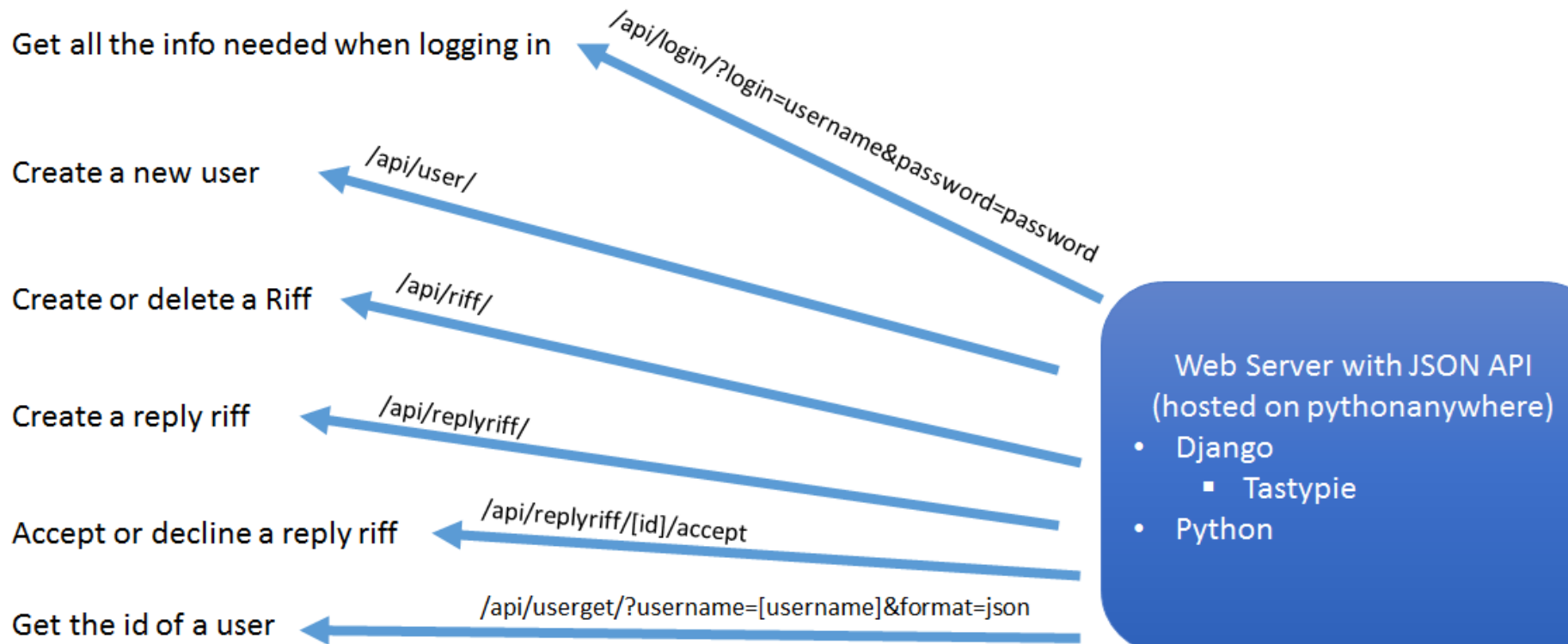
- Connectivity challenge
 - Changes the quality of the downloaded riff depending on the type of the internet connection.
- Context:
 - Detects which type of internet connection the user has.
- Adaptation:
 - Whenever a user submits a riff a version with a lower bitrate will be created.
 - When downloading a riff using:
 - Mobile data, a lower quality of the riff is downloaded.
 - WiFi, the previously uploaded quality is downloaded instead.

- Offline Challenge
 - To detect if a user has internet connection, we use the method `getNetworkInfo()` available in the classe `android.Connectivity`, and the `isConnect()` method available in `android.NetworkInfo`.
 - In order to save the audio file the user has recorded, we simply save it in the external memory (sd card) and then remove it after sending it to the external server.

- Usability challenge
 - To detect if a user has headphones connected to the device, we use the `isWiredHeadSetOn()` method of the class `android.AudioManager`.
 - To achieve the adaptation described above, we set the `AudioManager` to the handset mode (using the `setMode(AudioManager.MODE_IN_COMMUNICATION)` method), and then we disable the other speakers of the phone (`setSpeakerPhoneOn(false)`).

- Connectivity challenge
 - To detect which type of internet connection the user has, we use the `getType()` method provided by `android.NetworkInfo` and check if it equals to `Wifi` or `Mobile` (using the constants `ConnectivityManager.TYPE_WIFI` and `ConnectivityManager.TYPE_MOBILE` of the library `android.net.ConnectivityManager`).
 - The process of decreasing the quality of a sound is made on our backend using `pydub`. We simply lower the bitrate of the sound (setting it to 50).





- Client (Android)
 - android.media.MediaPlayer - to play mp3
 - android.media.MediaRecorder - to record sound files
 - squareup.okhttp - to connect to the server
 - Dropbox Core API
- Server
 - MySQL Database.
 - Python
 - Uploads audio files to the Storage Server (Dropbox)
 - Pydub library (deal with sound files)
- Storage Server (Dropbox)
 - Contains all the audio files of the application

File compatibility between pydub API and Android:

- When pydub tried to use a .mp3 file uploaded by the application, the format was incompatible. Due to that there was the need to upload it in .mp4 and the conversion to .mp3 takes place at the server.

Problems with SwipeRefreshLayout inside a scroll view:

- The Refresh button provided by `android.support.v4.widget` had several bugs when used inside another scrolling component. The solution encountered was enabling it only when the scroll views are on the top element and disable it otherwise.

Android Recorder latency delay:

- When recording on a device, we experienced a not-constant delay on different devices. This happens due the fact that it depends strongly on the processing speed that each device takes to start the recorder.

Android AudioManager.setMode delay:

- Changing the mode of the audio output is a process that may take not-constant time to be executed (depending on the phone).