**Problems that arose during development.**

During our development faze we accounted some problems in developing the stop functionality we encountered that when using the API stop function the program resets its buffer this will then not allow for the API to start playing the song from beginning. We found a way around this problem by using the pause function and the seek to function as displayed below.

*stop.setOnClickListener(new View.OnClickListener() {*

*@Override*

*public void onClick(View v) {*

*//because using mp.stop() makes the music stop playing without possibility to restart it*

*mp.pause();*

*mp.seekTo(0);*

*Toast.makeText(MainActivity.this, "music stopped", Toast.LENGTH\_SHORT).show();*

*}*

*});*

We accounted another problem with the language barrier as the API uses the Latin alphabet and does not have the letter æ-ø-å this gives problems wen adding songs.

We tried when setting the text size to use *String s=”<font size=”25/”>text</font>* but this did not work so we decided to use this line instead *t1.setTextSize(TypedValue.COMPLEX\_UNIT\_SP, 25);*

During the seek bar development we discovered that it wasn´t moving because we forgot to add a handler to handle the thread of continuingly update the seek bar according to the songs current position we set it to update every 1ms to make it run smoothly.

*public void seekbar\_update()*

*{*

*if(mp.isPlaying())*

*{ sb.setProgress(mp.getCurrentPosition());*

*Runnable r=new Runnable()*

*{*

*@Override*

*public void run()*

*{*

*seekbar\_update();*

*}*

*};*

*new Handler().postDelayed(r,1);*

*} }*

Another problem we ran into was using the media player to places both in the on create method and the seek bar method this gave us a lot of truble but it turned out we declared the media player as a field and the initialized it in the on create method but the reason it wasn´t working was because we initialized it to places in the code