

CrowdPlay

ITSMAP-01 Synopsis

Group 02

201404118	Anders W. Birkelund
201271001	Rune Rask
20080899	Khaled G. Edwan
201405166	Jonas R. Hartogsohn

Contents

Vision	3
Context	3
How it works	3
Early Design Overview	3
Component model.....	4

Vision

Our main goal is to get a deeper understanding of how several devices communicate in a practical application, using a web service. Furthermore, we want to explore the possibilities for using the API of an external service – e.g. Spotify, Google (Play Music), Youtube etc.

On a more basic level, we want to get experience in creating an app with a nice feel, e.g. using material design.

Context

Imagine that you are throwing a big party for all your friends. Everybody is having a nice time; the atmosphere is great and smooth tones are filling the room from your sweet B&O stereo. Still, you simply cannot relax! Why? Because every single time a song is over, someone wants to pick the next song and hook up their phone to your audio system. Your ears are crying as the gilded mini-jack plug enters one smartphone after the other – without the volume being turned down at all.

Not only is the noise terrible, but the songs being changed constantly is pretty annoying for everybody. This is where the *CrowdPlay*-app can help you out!

How it works

You open the app, choose a name and get the choice to be the *DJ* or to be the *Guest*.

DJ:

1. Choose a cool name for the party
2. Choose a few initial songs or an existing playlist to kick start the party
3. Share the unique identifier
4. START THE PARTY!

GUEST:

1. Hook up to the party using the unique identifier
2. Now you can:
 - a. Add a new song to the playlist
 - b. Up-vote an existing song in the playlist

The device logged in as DJ now plays the songs in the playlist from the top down. Songs move up the playlist, as they are up-voted.

Early Design Overview

We plan to use Firebase as our database, but also as our communication module between the guests and the DJ. The reason of this choice is that Firebase is a real-time database, which makes it possible to synchronize all the connected devices *live*.

The biggest risk in our proposed application is that we want to connect to an external API, which we do not have a lot experience with.

Component model

