**Slide 1**

Choonz create, rate and debate music with mates! brought to you by the Groopees.

**Slide 2**

Choonz allows users to create, upload or share music playlists from their favourite streaming sources. Create a new playlist and apply descriptive ‘tags’ to help categorise the playlist. Other users will then be able to rate and review your playlist/s and will be recommended similar popular playlists.

Our primary user, Susan is a music lover, she likes to create playlists to share with her friends. Is proud of her eclectic music taste and wants a way to share it with a wider audience.

She has some key requirements of the website which we will now demonstrate

**Slide 3**

Upon entering the website, she is presented with a range of top-rated playlists, she scrolls through each section to see what kind of ratings options there are.

The Awesome-sauce playlist catches her eye, so she checks it out.

**Slide 4**

Inside she can see the songs listed with links to listen on spotify as well as similar playlist tags, if she really likes this playlist, she can even check out the creator’s profile, Pete might just be her next favourite DJ.

**Slide 5**

She has seen enough and wants to join; she pops in her username email and password and is ready to go. Although if she had github twitter or spotify she could have logged in that way too.

**Slide 6**

She rates Pete’s playlists 5 stars and drops a comment, but she thinks she can do better

**Slide 7**

she decides to have a go at creating a playlist.

**Slide 8**

She adds a name, description and adds tags to her playlist

**Slide 9**

Once created she can edit her playlists adding songs manually or import them from an existing spotify playlist

**Slide 10**

She adds her favourite song before doing searching spotify for some hip hop

She remembers she already has a great playlist so imports with spotify,

**Slide 11**

Her playlist is now live and can be see on the front page of choonz under recent playlists

**Slide 12**

Having uploaded a playlist, she now wishes to listen to new music, she enters all playlists, she can filter by playlists or by name

**Slide 13**

Searching for playlists is great but she really wants to keep track of her ratings and be recommended something based on what she likes, entering her profile she finds all her rated playlists, favourite tags and a selection of personalised recommendations.

**Slide 14**

Checking out ‘view my stats’ we are presented with a variety of data and graphs relating to ratings and activity

**Design Spec**

As part of our design specification we were able to complete most of our objectives. The final to objectives could be included in future iterations

**Future Expansions**

Other Possible expansions could in include an Artist’s pages

And a ticket master api which would allow for easy monetization of the site

Other talking points include exporting playlists

As we don’t own the music we can’t offer a download however porting between rival music steaming services could be done if the scope of this project was increased.

Users may also wish to engage in more discussion as such future iterations could also have forum posts.

**App Dev**

Specification:

The final application varied both in scope and design from the original Design Specification document. For example, it was originally thought that Artists would play a larger role in the site, but in the building of the application it was found that they were not nearly as significant as expected.

Development Challenges:

As a group, our collective prior experience with JavaScript, jQuery, AJAX and CSS was close to nil. Thus, all aspects of the design and implementation of the application were learned in development. Libraries such as Bootstrap were used for some functionality, as spending time to create a unique design for basic components and interactions seemed like re-inventing the wheel.

Database and Model changes:

While the model was updated a couple of times, the edits were mainly superficial as the original design spec had been reasoned through fairly well. For example, nearly all objects in the model now have a slug for easier identification – something that was overlooked in the original spec. As mentioned above, the Artist object had a number of attributes removed, however, the main object relationships remained unchanged (see Model Updates).

**ER Diagrams**

Upon implementation of this project the decision was made to remove playlist images to reduce user requirements when creating a new playlist.

fundamental changes to our model resulted in a reduced need for artists fields.

Artists no longer had a like relationship and rather than playing songs simply owned them.

Consequently, another minor change to our relationship model was many songs being included rather than liked by many playlists

**URL**

Links remained generally the same as the specification with only minor omissions relating to artist pages

**System Architecture**

Likewise, you system architecture remained the same with minor changes with github api being selected for log in over facebook and python and sql search replacing haystack and whoosh.

Choonz create, rate and debate music with mates!

Thank you