Dean Hetherington

H00013640

BSc Computer Science

StreamBot: A public display interface for music selection

April, 2014

Dissertation Supervisor: Judy Robertson

Second Reader: Lilia Georgieva

Declaration

I, Dean Hetherington confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed:

Date: 28/04/14

Acknowledgements

I would like to gratefully acknowledge my wonderful supervisor, Judy Robertson whose support enabled me to achieve the goal of finishing this project and greatly improve my writing ability.

I would also like to thank Sarah Donnachie, who gave me inspiration and put up with reading a lot of drafts.

Abstract

In this paper, I describe my current research in the field of interactive large public displays within a social environment. The core of my research is based on multiple user interface concepts and comparing them in a way to find which is most effective. StreamBot was created to test this. A system that can receive concurrent song requests through Twitter, process and compile them into a playlist that is played/shown on a public display. Research also included how effective and usable such a system would be. As a result of my research I found that StreamBot is a very usable system and scored highly in System Usability, regardless of the interface and design patterns used.

CONTENTS

1. Aims and Objectives

1.1. Introduction

While there is a body of evidence to suggest that larger display technologies are becoming increasingly popular within social environments and social situations (G. Schmidt, 2006), it remains unclear what content and interaction methods are the most effective use for them. For example, making a playlist of music on your own is as easy as loading up YouTube/Spotify and clicking add to playlist: but what if this was to be a more collaborative effort? It is more difficult for practical reasons to have everyone within an open space adding to a playlist, as the users may end up crowded around the one input device to input their personal preferences and then it is not always clear if you have been successful or not.

How can information be shown in such a way that interested onlookers can actively engage with a public display and receive feedback in next to real time? Using a mobile phone and Twitter as an input method for almost real-time input could be a viable solution to collaborative input in social situations. In this paper, **StreamBot** will attempt to test this method: a collaborative input web application to create video playlists for social situations. This study's research goals are to:

- To provide future researchers with recommendations regarding how to display and interact with a shared music player interface on a public display
- To design a visually appealing and enticing user interface for displaying a shared music interface on a public display.
- To explore in detail how usable Twitter is as an input method in the context of a shared public display

In order to reach these goals, a research question was considered (explained in greater detail in section 4.2)

- What factors encourage/discourage a potential user from using a public display for a shared music application?

Additionally, further data was gathered from users to find secondary but relevant research:

- What are users' preferences regarding layout for StreamBot on a public display?
- How can Twitter be used as a real-time input method and how does this compare to *a keyboard and/or mouse?*

2. Literature Review

2.1 Why Twitter?

Twitter is characterised as 'microblogging' or 'tweeting'. Posts are very short entries (140 characters or less) or updates on a blog or social networking site, typically via a mobile phone. Compared to blogging, tweeting fulfils a need for an even faster mode of communication.

By encouraging shorter posts, twitter lowers users' requirement of time and thought investment for content generation. Another important attribute is how often updates are made: on average, an active blogger may update their blog once every few days (T.Finin & B. Tseng, 2007). On the other hand a twitter user may post several updates in a single day.

This speed and ease of use was the main factor in why Twitter was chosen as StreamBot's main input method. A user can make a song request just as easily as they can post a new tweet. Ultimately this is a positive with regards to this project. Requests to the application can be made quickly and without much thought.

Secondly, 80% of UK users access Twitter through their mobile device. Combined with the fact that 3 in 5 Twitter mobile users are on Twitter whilst watching TV (Nielsen, 2013), it makes Twitter an ideal platform for quick and portable input to a public display.

Furthermore, anybody can create a Twitter account and use it from a variety of devices (e.g. phone, laptop and SMS. Rather than using one static interface that was forced onto all users, Twitter allows users to use their own devices as the input method. Bringing familiarity and less of a learning curve than one overall, unified and unfamiliar input method.

In order to understand Twitter and appreciate why Twitter has gained so much popularity, this study will look into why people use Twitter and what they use it for. T. Finin & B. Tseng (2007) attempted to analyse twitter users' reasons for tweeting. Popular Twitter users were split into nodes and other less popular users into edges connecting each node. By analysing the content of each tweet in the dataset, it was determined that the main reasons why people use twitter are:

- Chatter - Most posts on Twitter talk about what people are currently doing. This is the largest and most common user of Twitter.
- Conversations - users comment to one another using an @ symbol followed by the directed username. About one eighth of all posts in the collection contain a conversation.
- Sharing information - About 13% of all the posts in the collection contain some URL in them. Due to the small character limit a URL shortening service like bit.ly is frequently used to make this feature feasible.
- Reporting news - Many users report latest news or comment about current events on Twitter.
- Information Seeker - An information seeker is a person who might post rarely, but follows other users regularly.

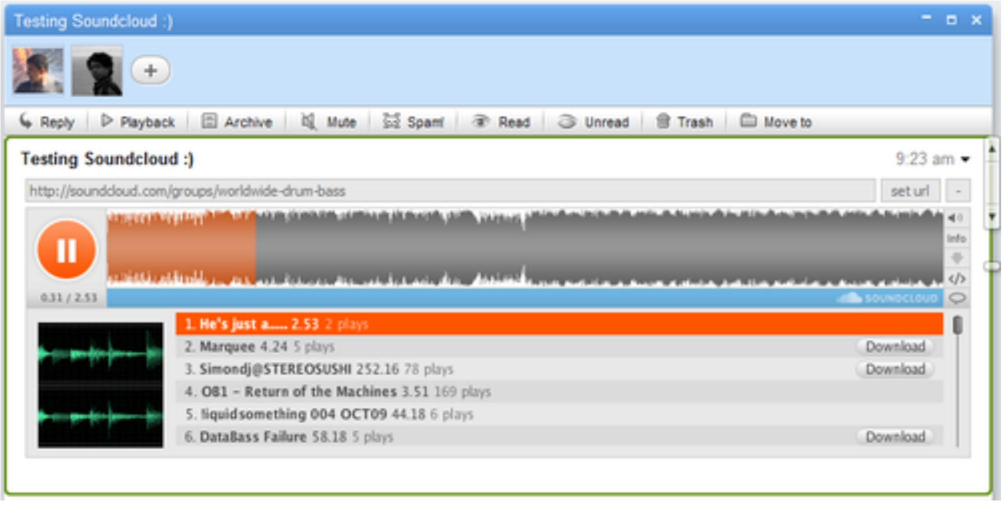
It is worth noting that this study makes no mention of Twitter as an input source for another application.

However, at a higher abstracted level - mobile and Internet connected devices can still be used as input devices, M.Kuhn, R. Wattenhofer and S. Welten (2010) iterate that many displays are already equipped with touch screens, allowing content to be created ad-hoc as users walk up to the display. There are many situations where using the mobile phone as an alternative interaction technology might be more suitable as an input. For example, a phone provides additional artistic freedom by using, e.g. the camera, to augment posts with images or videos. Secondly, posts could be created on-the-go, e.g. as users commute on the train knowing that they would pass a display later on their journey. Thirdly, inputting personal information on public displays, such as an email address or telephone number, raises privacy issues due to lurkers and shoulder surfers. With the mobile phone, such sensitive data can be entered in a secure manner.

Twitter is a good choice for StreamBot's input method because it is mobile and allows concurrent users to interact and input their choices. Users are not tied to their current location. This is critical as its intended use is within a public space where users are usually moving. Essentially, the use of Twitter means that anyone with a device that is capable of tweeting (by SMS, App, Web) can use StreamBot.

2.2 Music Interfaces

There are a variety of ways to display a song that is currently being played within a music interface. One way is for a song to be played in a two or three-dimensional format, where the audio is mapped onto some timeline that can easily be skipped: this is considered to be a traditional method (see Soundcloud for example).



Testing Soundcloud :)

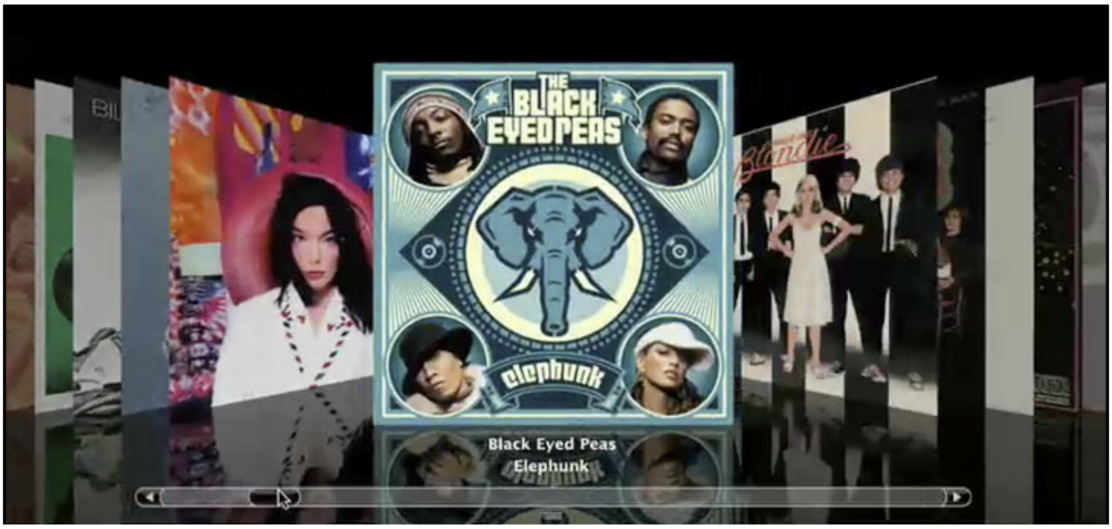
9:23 am

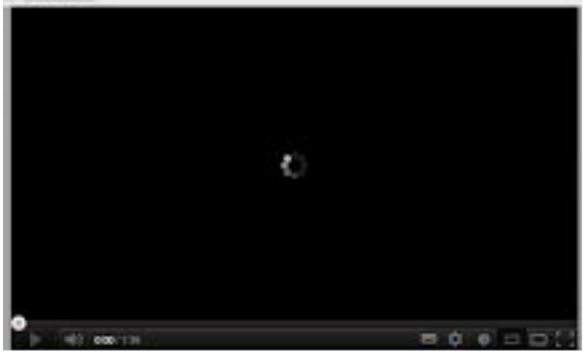
http://soundcloud.com/groups/worldwide-drum-bass

0.31 / 2.53

1. He's just a..... 2.53 2 plays
2. Marquee 4.24 5 plays
3. Simondj@STEREOSUSHI 252.16 78 plays
4. O81 – Return of the Machines 3.51 169 plays
5. liquidsomething 004 OCT09 44.18 6 plays
6. DataBass Failure 58.18 5 plays

There are however other methods such as Sony Ericsson's commercial SensMe interface which maps songs along two axes, mood and tempo. Apple's cover-flow interface replaces the textual album lists with album covers. The sheer popularity of this approach shows that album covers are a useful and familiar visual hint to retrieve and recognise music.





StreamBot will use Youtube videos and the Youtube player as a source of music so active music elements within the application will be limited to a more traditional 2D timeline for audio (as required by the player). However, this does not limit the interface around it for the rest of the application.

Following Apple’s success and proven use of visual art, there will be heavy emphasis on song artwork (see section 3.2.1).

Given that music applications are exceptionally popular and useful elements from these have been discussed above, it is worth delving deeper to look at similar music delivery platforms to StreamBot: such as Spotify, Last.fm and Pandora. M.Kuhn, R. Wattenhofer and S. Welten (2010) emphasise that music recommendation platforms are becoming ever more popular and useful for music discovery. They created various layouts for a mobile music application to contrast and compare.

The application was published on the Android Market, the App Store for Android. At the first start-up, the user was asked for permission to log anonymous usage data. Overly short log files were removed, as interest in the usage of ‘regular users’ was considered to be more important. The data gathered from the log files was then analysed to find the most effective way to display a music interface.

Their research shows that just using user feedback from the audio played within the application is severely limited compared to using implicit and explicit methods of measuring user feedback. A lot more data can be found when the user goes through a questionnaire or when user behaviour is further monitored. This is confirmed by the success of the aforementioned online services.

While automated analysis may be useful in some situations for non-subjective testing, it proves to be difficult in application for real-world settings. Music is targeted at people and the perception of music is inherently subjective. Thus, any testing for a music application that is based around the content or the music playing rather than the interface will not provide the most useful results. It can be difficult to find similar music solely through automated analysis. The best results come from user feedback.

On that basis, StreamBot's interface was not evaluated through automated user metrics. Instead, real-life usage was simulated and users were monitored and asked to fill out System Usability scales and feedback surveys to analyse StreamBot's interface. (See section 4.3)

2.3 Enticing Use and User Input

While there may be a variety of ways to display a music interface, we also have to consider how this may be displayed on a larger public display. There is an important difference between a user interface displayed on a user's personal small phone and a larger scale display that they have to be encouraged to use in a social space. For this reason it is helpful to examine how to persuade users to interact with a larger screen.

The main challenge of a public display is getting the user to engage with it in the first place: to approach it and begin interaction. This is normally broken down into three separate phases:

1. Noticing the display
2. Understand the display is interactive
3. Be persuaded to become active users

H. Kukka, H. Oja and V. Kostakos (2013) looked into three separate elements of how a user could be engaged to interact with a large public display: colour/greyscale, animation/static and icon/text.

Using eight different combinations of colour, motion and graphics, tests were carried out on eight interactive displays on a university campus for eight days. At midnight every night, the displays were swapped for each location so every display had shown every test.

They found from a smaller testing group that using skeuomorphic elements such as a hand icon touching the screen or a virtual keyboard may encourage interaction but it was generally found that real-world physical objects perform better at promoting engagement. A keyboard attached to a screen is more likely to encourage interaction than a virtual one displayed on-screen. Following recommendations from literature (section 2.4) yellow objects on a blue background were used for their experiment.

For the experiment, the screens were monitored from afar. Random passers by could interact with the screen on a variety of levels. Briefly looking at the screen, touching the screen and leaving, or actively engaging with the screen and completing the short questionnaire that would come up. The effectiveness of the engagement pattern was based on how much interaction the users had with the interface.

It was found that displays with colour were more effective when the text was animated, while greyscale was more effective when the text was static. Displays that used icons were equally effective regardless of colour but static icons were better than animated.

Findings show that text is more effective in enticing interaction than icons, colour is more effective than greyscale, and static elements on screen are more effective than animated for attracting possible users attention. Furthermore, the study identified a distinct behavioural pattern: display avoidance. Display avoidance refers to people actively looking away from a display even when they are at arms' length from it, mainly in order to avoid information overload.

Based upon these findings, these successful elements will be used as the basis for StreamBot's interface. (Section 2.4). Display avoidance is also worth noting but may not be a problem since people are likely to remain around a public display (and less likely to avoid it) within a social situation like a party.

Also worth noting that when enticing user input, the Honeypot effect has been observed and described by Brignull et al. (2003) in the context of the 'Opinionizer' public display that was shown during a party. Whenever a crowd of people had already gathered around the display, this crowd seemed to attract a lot of attention and other people were much more likely to also attend the display. This can only be a positive for StreamBot as its intended use is in a social setting such as a house party.

2.4 Design for Public Screens

When designing graphics that need to both attract attention and be legible at the same time, L. MacDonald (1999) recommends using colour combinations with high contrast such as black or blue on yellow, or vice versa.

H. Kukka, H. Oja and V. Kostakos (2013) also created a study of 233 people from 22 different countries, it was found that 35% of female respondents and 57% of male respondents (i.e. 42% of all respondents) named blue as their ‘favourite’ colour. Marketing research shows that blue is often associated with trust and security, and used by formal restaurants to create a calming and relaxing atmosphere. Yellow is associated with optimism and youth, and is often used to catch attention of window shoppers. Following this research, high contrast colours (black/white, blue/green) were used prominently within StreamBot’s design.

To further understand designing for public screens, we must also consider looking at a higher level of design, the design space: the overall and entire range of independent design choices. Research from J. Müller, F. Alt, D. Michelis et al. (2010) wrote that design space on public screens could be split up into multiple categories:

- Posters - With the digital advertising column, the display implicitly reacts to passers-by by showing flowers wherever people stand, using the modality of body position.
- Window - Hole-in-space serves as a window to another place. Here, the effect of the display depends entirely on the audience that is present at the other end. For a passer-by, the other audience may follow him with their eyes, which is also a form of implicit interaction.
- Mirror - Magical mirrors shows an augmented mirror image of passers-by. The mirror model directly supports implicit interaction, as any passer-by will be reflected on the display. As the passer-by looks at the display, attracted by the movement, their curiosity may be raised by the augmentations on the mirror image.
- Overlay - The Jumping Frog for the Everywhere projector shows a frog on some surface in the environment. Implicit interaction is supported when a user accidentally steps nearby a frog displayed on the floor and the frog jumps away.

StreamBot falls in the category of a poster. Other designs and layouts will be used (see section 3.2.1) but essentially, all variations will also fall under poster. Playing and queued songs will be shown with a list of user requests reacting as new songs come in. Informal testing was carried out within a group of five potential users and it was found that any other sort of design space would cause too much wasted space on screen to get the user's attention. It is also questionable as to whether designs categories such as an overlay would work within a social setting with many people.

2.5 Sharing Songs Socially

One of the main challenges inherent within this project is that large public displays tend not to give a feeling of ownership or personality to their viewers in the same way that a personal computer might do.

Mobile devices, particularly mobile phones, are completely personal devices that usually hold personal user information such as contact records and user profiles. Unlike the inherent public exposure of large public displays, interaction with such displays using mobile phones can bring a certain level of privacy and intimacy to the act of interaction.

J. Seeburger, M. Foth and D. Tjondronegoro (2012) observed that in general, interactions with large public displays can be either overt or covert. In an overt interaction, it is explicit who is interacting with the large display, how they are doing it, and what modifications they are making to the content on the display. However, in a covert interaction, the user interacting with a large public display is not completely visible to other spectators or bystanders. A covert interaction can be considered as a way of hiding the identity of the actor from the audience watching the display, thus bringing more privacy.

While spending time in social situations, people may use their electronic devices as a way to “cocoon” to create their own personal space (S. Komulainen & N. Oyj, 2010). Whether purposefully or not, this can lead to avoiding direct contact with surrounding strangers. Even without a mobile phone reception, people tend to use their devices for different purposes such as playing games or listening to music.

Instead of secluding oneself from the surrounding environment, portable music devices can also be used to connect – or simply associate – with other people in space utilising music as a common ground for interaction.

K. Liu and R. Reimer (2008) created ‘The Social Playlist’ as a mobile application enabling friends to create a user-generated radio based on selected songs from different data sources. Based on current mood or activity, user selected music is transferred to a server that generates a music playlist which is streamed to a group of friends. The research conducted has shown that the selection and sharing of songs can create anchor points for follow-up social interactions.

J. Seeburger, M. Foth and D. Tjondronegoro followed up on their initial research and created an app called Capital Music. It visualises songs that are currently played in the vicinity of users, rather than a visualisation of users themselves.

The mobile application was developed for iPhone, iPad and iPod Touch devices using the Apple iOS software development kit.

The trial took place in a university meeting room and participants were asked to bring their own devices containing their personal music library. Participants were briefed on how Capital Music works, in particular how to set a nickname and how to access the music library. Afterwards the participants were asked to use the application.

One participant was concerned about the public's opinion of his song choice: "I don't want that people know what I listened to previously. So I listened to this song and it was really bad and no one likes it and people start hating me for it. So now I'm listening to this [new] one and also want to change my name."

After using the application, participants were asked if they felt comfortable sharing their currently played songs with collocated people. All participants felt comfortable. Three different groups of users emerged. The first category of users was "all for sharing music" or in general "like sharing things." The second category consists of users who are comfortable sharing song selections due to the anonymity of the application. One participant explained that, "all they get from me is a nickname and my song selection. I don't feel like I'm giving too much information away." The last category of users consists of people who felt comfortable with limitations on sharing. These limitations are either related to their own song selections in terms of how "embarrassing your music is" or if the song has been illegally downloaded before the official release date.

Participants were asked how they felt about other people seeing what they are listening to and if that influenced their song selection. A group of participants explained that other songs currently played in the same space would influence their own song choice. “Some of the things that were showing up in electronic were more in my taste so I found myself navigating in my own library towards electronic. Even though in the main I tend to be more on the other side, more into the rock side of music.”

The anonymity of Capital Music was mostly perceived as beneficial for mobile mediated interactions in public spaces. “You don’t have to sign up or use personal information. That’s quite good.” Another participant explains that “the more information I have to give the less inclined I would be to interact with the application.” However, some participants were slightly concerned about anonymous interaction in public places.

In general, study participants enjoyed the anonymity of the mobile mediated interactions but security mechanisms such as blocking users should be implemented in the next iteration of Capital Music.

More research and testing within user groups will have to be done to find out whether users that make song choices are viewable for everyone or not. Given this inconclusive research, it was useful to include a question about preferences regarding anonymity in StreamBot’s user feedback session.

3. Design and Development

StreamBot was built using an iterative and incremental prototype development cycle. Features were built upon and evolved as development went on. Design choices that were made evolved and developed as the application became more feature-complete.

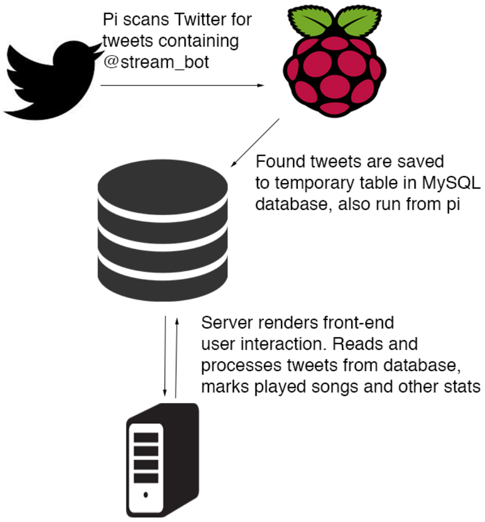
3.1 Technical Design

StreamBot's backend and processing is written largely in PHP, implementing third party PHP frameworks for searching Twitter and finding suitable music requests on YouTube.

The whole system runs from a virtual private server and a Raspberry Pi. The server hosts static files such as the front-end of the system and everything that the user will see and interact with. The Raspberry Pi hosts the database and the back-end for the system.

The input system of StreamBot comprises of two background processes running indefinitely within a command line interface. Based upon the Phirehose framework, the processes search for and monitor incoming tweets to the user @stream_bot. One process finds new tweets and saves them to a table within a MySQL database; the other process parses requests within the table, creating a readable format for the YouTube API to use.

The user interface and layout is written in HTML5, CSS3 and JavaScript (jQuery). Using AJAX requests for immediate feedback, a JavaScript function within each user's browser queries the database every three seconds and updates the page accordingly. When a new request is found within the database, the string is searched for on YouTube using Google's client API framework. When a video ID is returned it is appended to the database row of the searched video. Requested songs are displayed on screen as a generic thumbnail until they are fully processed. As they are displayed, their video ID is used to find the artwork as an embeddable image and that replaces the generic thumbnail when found.


Pi scans Twitter for tweets containing @stream_bot
Found tweets are saved to temporary table in MySQL database, also run from pi
Server renders front-end user interaction. Reads and processes tweets from database, marks played songs and other stats

When a song is the first row in the database, a new YouTube player instance is created using JavaScript with the attached video ID. As a video ends, it destroys the current player instance and moves the song to another database (removing it from requests). The function then calls itself, using the new first row in the database and moving to the next song.

It is important to note that after developing a technical design, research found that Alt, Shirazi, Kubitza, Schmidt (2013) investigated the potential of using a mobile phone to interact in this new form of communication medium by creating a digital public notice area, (Digifieds1).

Their display client used AJAX to create an interactive UI capable of attracting and enticing people through immediate feedback. HTML5 and CSS were used to layout the content and a browser in kiosk mode to run the client. Using asynchronous HTTP requests, their display client periodically polled for data changes. If there was any new content, the user interface updated. The default update rate was 30 seconds. The internal browser cache minimises the data traffic and was used for media documents. The browser’s local storage API saves the classified’s data in JSON format even between browser sessions or in network-loss situations. StreamBot’s design was very similar to this and chosen as the final method because it has been shown to work within successful studies.

System Requirements

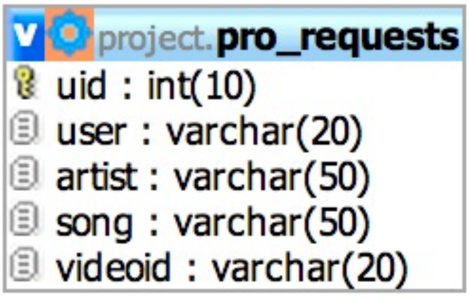
For backend processing of tweets, StreamBot requires a server or local machine running a minimum of PHP 5.1. The frontend user interface is displayed within an Internet browser. Most modern browsers capable of rendering HTML5 and CSS3 are supported by StreamBot. These include Chrome, Firefox 4+, Safari and Internet Explorer 10+.

Technical Development

StreamBot's development was based around core, release and stretch goals (see Appendix. 7.1). As of April 2014, StreamBot contains all core and release features.

Under the original development plan, the whole StreamBot system was hosted on a single virtual server (2.2 GHz, 512MB of RAM). Due the constantly updating nature of the database and the need for immediate user feedback, the system was under continual stress and prone to crashing. Database development was moved over to a Raspberry Pi (800MHz, 512MB RAM), spreading the compute load across two systems and ensuring a far greater uptime.

The original database schema was first designed as a proof of concept and only stored what user requested a song and the song they requested.



project.**pro_requests**

uid : int(10)

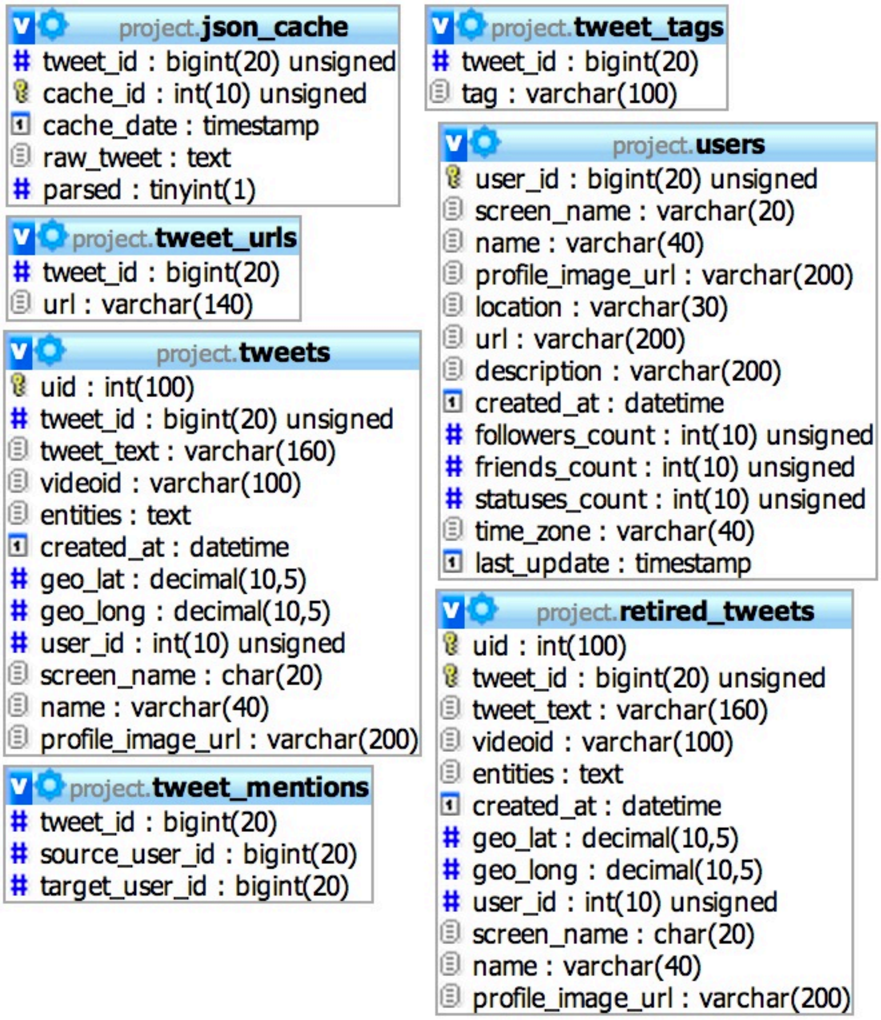
user : varchar(20)

artist : varchar(50)

song : varchar(50)

videoid : varchar(20)

Iterating over code and adding core features forced the database schema to be updated to accommodate for new features and methods of analysis.



```
project.json_cache
# tweet_id : bigint(20) unsigned
cache_id : int(10) unsigned
cache_date : timestamp
raw_tweet : text
# parsed : tinyint(1)

project.tweet_urls
# tweet_id : bigint(20)
url : varchar(140)

project.tweets
uid : int(100)
# tweet_id : bigint(20) unsigned
tweet_text : varchar(160)
videoid : varchar(100)
entities : text
created_at : datetime
# geo_lat : decimal(10,5)
# geo_long : decimal(10,5)
# user_id : int(10) unsigned
screen_name : char(20)
name : varchar(40)
profile_image_url : varchar(200)

project.tweet_mentions
# tweet_id : bigint(20)
# source_user_id : bigint(20)
# target_user_id : bigint(20)

project.tweet_tags
# tweet_id : bigint(20)
tag : varchar(100)

project.users
user_id : bigint(20) unsigned
screen_name : varchar(20)
name : varchar(40)
profile_image_url : varchar(200)
location : varchar(30)
url : varchar(200)
description : varchar(200)
created_at : datetime
# followers_count : int(10) unsigned
# friends_count : int(10) unsigned
# statuses_count : int(10) unsigned
time_zone : varchar(40)
last_update : timestamp

project.retired_tweets
uid : int(100)
tweet_id : bigint(20) unsigned
tweet_text : varchar(160)
videoid : varchar(100)
entities : text
created_at : datetime
# geo_lat : decimal(10,5)
# geo_long : decimal(10,5)
# user_id : int(10) unsigned
screen_name : char(20)
name : varchar(40)
profile_image_url : varchar(200)
```

Following research from M.Kuhn, R. Wattenhofer and S. Welten (2010), a monitoring system was implemented that will ignore infrequent use and edge cases where users may not act as intended.

Exceptional behaviour will be logged but will not be included in what is defined as 'normal' user behaviour. To do this, StreamBot's database was redesigned and rebuilt to store far more metadata. Allowing for analysis of quantitative data such as tweets per minute and the ability to implement future features (see Appendix 7.1). Furthermore, the ability to store URLs was added given that: “13% of all the posts in the collection contain some URL in them.” (T. Finin & B. Tseng, 2007)

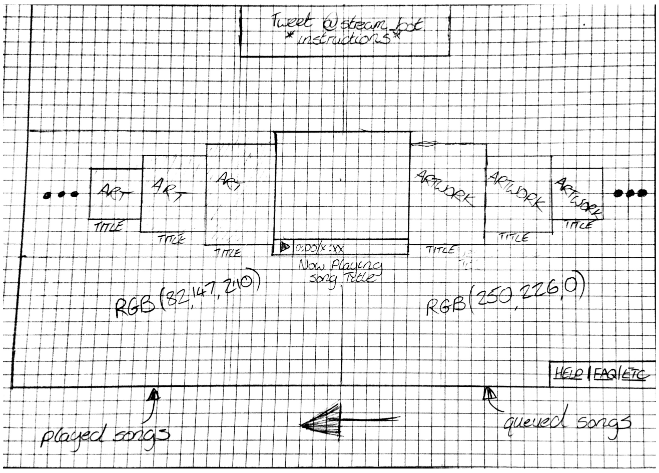
During one stage in development, Twitter updated their API authentication protocol. The new protocol blocked StreamBot’s application keys and stopped the ability to tweet to the application. This was quickly rectified by requesting new application keys and updating code to use the new authentication method.

User Interface Design

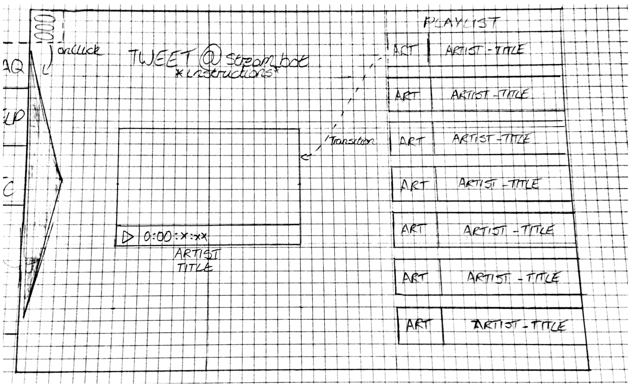
Due to the chosen research methodology (see section 4.3), StreamBot needed to be tested with multiple interfaces. To create these, research was carried out to compare and contrast various design patterns. These were chosen by using informal user evaluation and academic literature:

- Carousel - Allows better utilization of page real estate while providing an engaging experience when presenting pictorial objects. This pattern should be used when a set of objects are closely tied with imagery (in this case, album art) and is useful for maximising screen real estate as elements on a big screen must be large and visible in a public setting.
- Page Grids - A standardized grid providing a consistent and cohesive user experience with flexibility to incorporate common and dynamic page elements. The main advantage of this design pattern is familiarity, a large quantity of websites are designed like this. Page grids also known to be clean and consistent.
- Display Collection - enables objects to be displayed with similar attributes, should be used when showing a collection. The pattern should be used when showing a group of similar objects (in this case, requested songs).

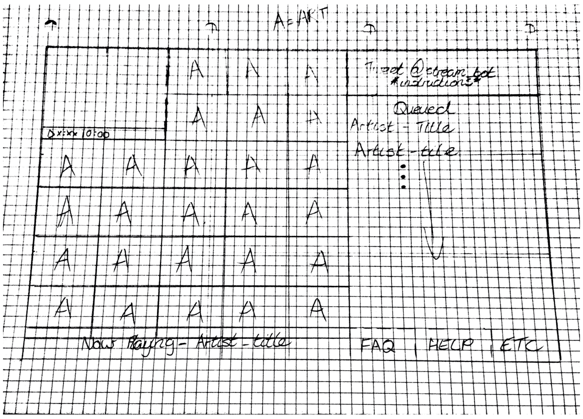
These patterns were chosen because of their variety. Carousels allow for a large and simple interface but forces users to primarily focus on the centre element. Page grids provide familiarity to users but may not necessarily be optimized for a larger screen. A display collection can show a large collection of songs simultaneously but individual elements may appear too small. Multiple wireframe mock-ups from the chosen design patterns were created and are shown below:



1) Carousel - Major emphasis on artwork and previous/next songs. Little text and heavy emphasis on artwork and currently playing song.

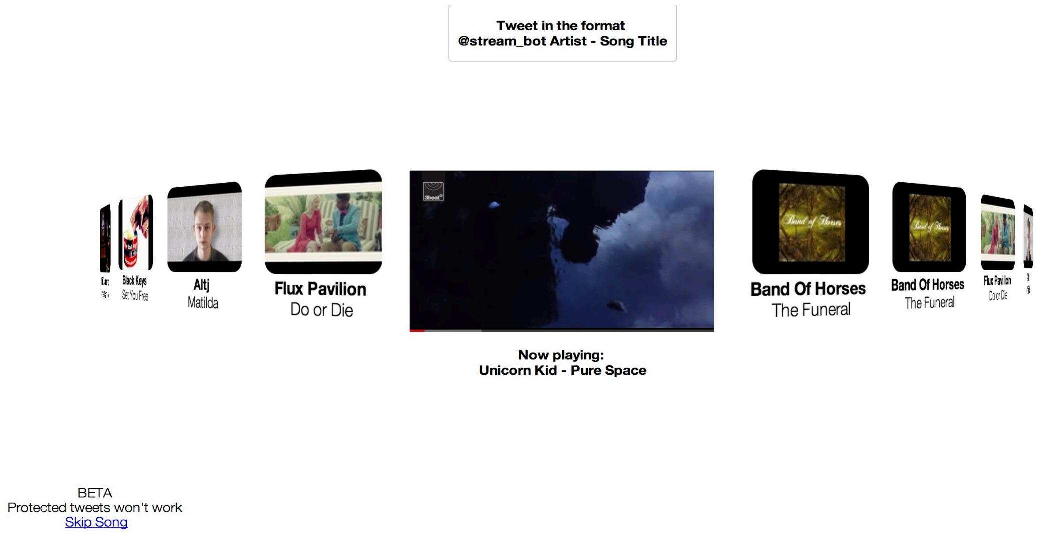


2) Page Grid - A sparse layout focussing primarily on what is currently playing. Gives a more traditional playlist.



3) Display Collection - A layout that mixes elements from layout 1 and 2. Combines playlist format from layout 2 and heavy use of graphics as in layout 1. Lots of little previews of artwork.

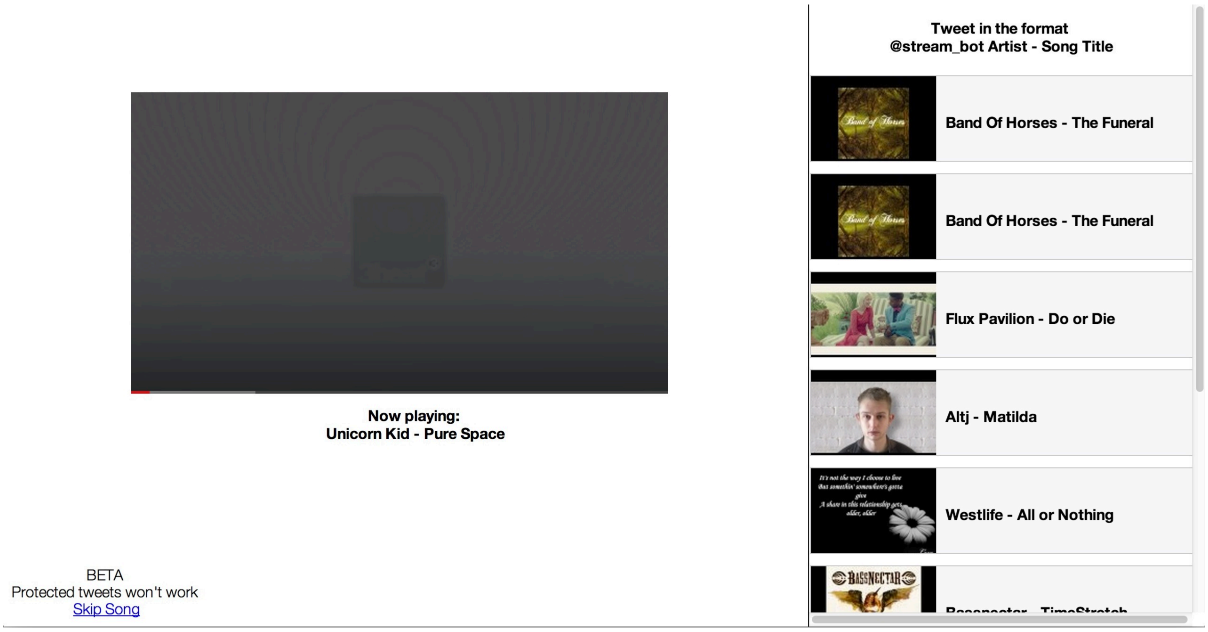
The wireframes were then created with HTML and CSS and then implemented with the back-end functionality of StreamBot:


Tweet in the format
@stream_bot Artist - Song Title

Now playing:
Unicorn Kid - Pure Space

BETA
Protected tweets won't work
Skip Song

Layout 1



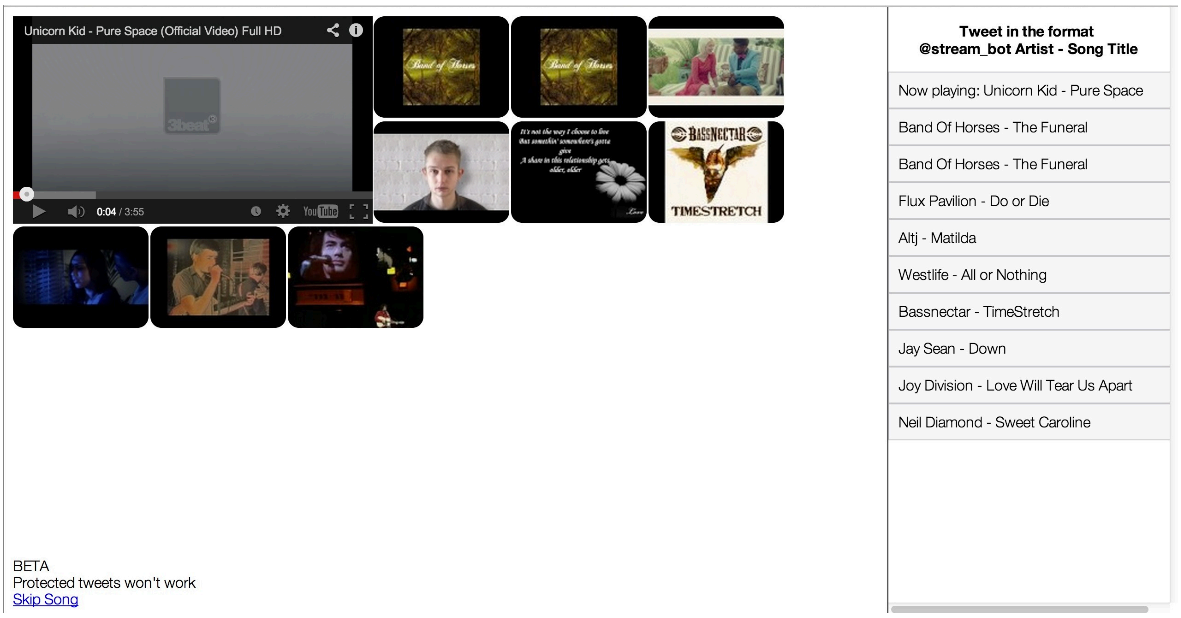
Tweet in the format
@stream_bot Artist - Song Title

- Band Of Horses - The Funeral
- Band Of Horses - The Funeral
- Flux Pavilion - Do or Die
- Altj - Matilda
- Westlife - All or Nothing
- Bassnectar - TimeStretch

Now playing:
Unicorn Kid - Pure Space

BETA
Protected tweets won't work
Skip Song

Layout 2



Tweet in the format
@stream_bot Artist - Song Title

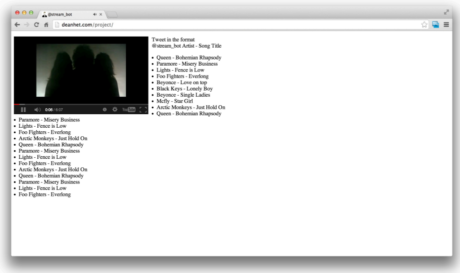
- Now playing: Unicorn Kid - Pure Space
- Band Of Horses - The Funeral
- Band Of Horses - The Funeral
- Flux Pavilion - Do or Die
- Altj - Matilda
- Westlife - All or Nothing
- Bassnectar - TimeStretch
- Jay Sean - Down
- Joy Division - Love Will Tear Us Apart
- Neil Diamond - Sweet Caroline

BETA
Protected tweets won't work
Skip Song

Layout 3

Pilot Testing

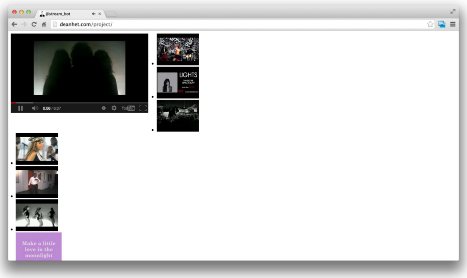
StreamBot used an iterative development process, interfaces were developed using informal user feedback and real life usage. Elements of each interface changed and adapted to how users interacted with the system. An example of an iteration is shown below:

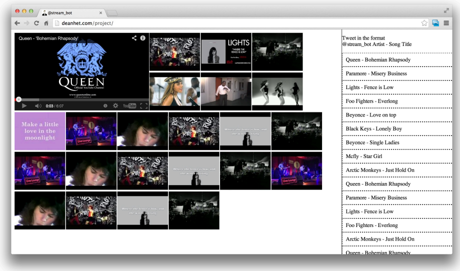


Tweet in the format
@stream_bot Artist - Song Title

- Queen - Bohemian Rhapsody
- Paramore - Misery Business
- Lights - Fence is Low
- Foo Fighters - Everlong
- Beyonce - Love on top
- Black Keys - Lonely Boy
- Beyonce - Single Ladies
- Mcfly - Star Girl
- Arctic Monkeys - Just Hold On
- Queen - Bohemian Rhapsody

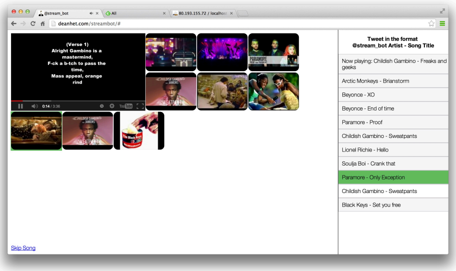
- Paramore - Misery Business
- Lights - Fence is Low
- Foo Fighters - Everlong
- Arctic Monkeys - Just Hold On
- Queen - Bohemian Rhapsody
- Paramore - Misery Business
- Lights - Fence is Low
- Foo Fighters - Everlong
- Arctic Monkeys - Just Hold On
- Queen - Bohemian Rhapsody
- Paramore - Misery Business
- Lights - Fence is Low
- Foo Fighters - Everlong




@stream_bot
deanhet.com/project/
Queen - Bohemian Rhapsody
QUEEN
LIGHTS
Make a little love in the moonlight

Tweet in the format
@stream_bot Artist - Song Title

- Queen - Bohemian Rhapsody
- Paramore - Misery Business
- Lights - Fence is Low
- Foo Fighters - Everlong
- Beyonce - Love on top
- Black Keys - Lonely Boy
- Beyonce - Single Ladies
- Mcfly - Star Girl
- Arctic Monkeys - Just Hold On
- Queen - Bohemian Rhapsody
- Paramore - Misery Business
- Lights - Fence is Low
- Foo Fighters - Everlong
- Arctic Monkeys - Just Hold On
- Queen - Bohemian Rhapsody



(Verse 1)
Alright Gambino is a mastermind,
F-ck a b-tch to pass the time,
Mass appeal, orange rind

Tweet in the format
@stream_bot Artist - Song Title

- Now playing: Childish Gambino - Freaks and geeks
- Arctic Monkeys - Brianstorm
- Beyonce - XO
- Beyonce - End of time
- Paramore - Proof
- Childish Gambino - Sweatpants
- Lionel Richie - Hello
- Soulja Boi - Crank that
- Paramore - Only Exception
- Childish Gambino - Sweatpants
- Black Keys - Set you free

Due to the nature of the study, only two interfaces were needed to compare. As layout 2 and 3 both shared a similar layout of content on the left and side-pane on the right, one was selected. To make this choice both interfaces were posted on Twitter, informally asking users to choose which interface they would be more likely to use. From 102 followers, 8 people responded. All said that they would rather use layout 2 than layout 3. As a result of this feedback, layouts 1 and 2 were the ones included in the study and comparisons.

3 Research

Introduction

The literature review covered a major part of the research by identifying different trends, methods and styles for displaying information on a public display.

When considering the aims and objectives for this project, a focused and narrow research methodology was chosen rather than an all-encompassing study and vague result. To do this, three questions were asked that would be answered using both qualitative and quantitative data gathered from the experiment.

Research Questions

The research questions aimed to cover three areas of using a public display within a social space:

- *What are users' preferences regarding layout for StreamBot on a public display?*

The aim was to find useful information regarding what a user wants to see on a public screen in a social space. The focus was solely on the layout for the user interface, as it is the most forward facing for the user. StreamBot was tested with multiple variations and gained extensive user evaluation for each variation.

The second question continues with, once the user has a preferred interface, what can affect their use of the application:

- *What factors encourage/discourage a potential user from using a public display for a shared music application?*

Based upon findings from pilot testing, the preferred interfaces were further developed. The study then measured factors within different interfaces that would encourage a user to interact with a shared music application.

Other factors were also noted following research, for example, external factors such as the Honeypot Effect could have been measured. How quickly the user receives visual feedback and how other users use a shared music application could also have affected overall use. This study aimed to find and measure different factors that can affect how a shared music application is used within a social space.

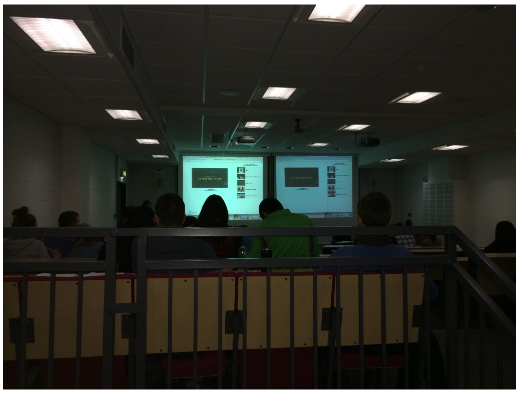
The final question aimed to use Twitter as an alternative input method:

- *How can Twitter be used as a real-time input method and how does this compare to a keyboard and/or mouse?*

The aim here was to build and use as the main input method, a system where users can message an automated Twitter account and see their requests automatically processed from their tweet. Although this question does not address any particular goals of this study, it was valuable research in finding how effective this is within a social space compared to traditional input methods such as a keyboard and mouse.

Research Methodology

The research methodology chosen for this section of research was in the form of a standardised system usability scale (Brooke, 1996) and a short questionnaire. This allowed quantitative and qualitative to be collected and collated. The nature of this dissertation constrains the available research methods - as the systems intended use is within a busy and public space - testing is limited to large groups and Twitter users.

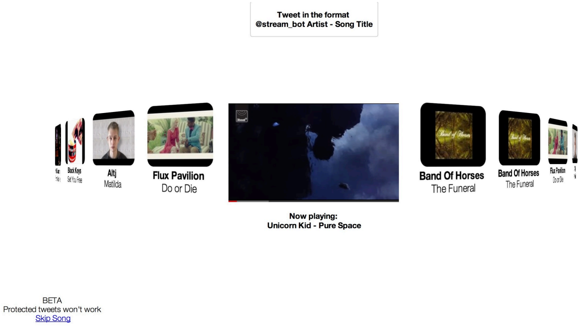


This study used a between-subjects design. There was one independent variable: the layout and design of the interface. The dependent variable was a system usability scale, defined using a practiced and in use method.

There were 25 participants who evaluated StreamBot (16 of which owned an active Twitter account). The participants were students of a university interaction design class during the third week of February 2014. Participants were between the ages of 20 and 29. 17 male, 7 female, 1 undefined and all were free from any obvious physical or sensory impairment that may affect their ability to measure the effectiveness of the interfaces. Participants took part in the study willingly and the aims and purpose of the study was explained to them prior to their participation.

Measurement of tweets per minute was calculated using attached recorded times from the tweets used to request songs. Each interface was shown on a large projected screen in a small lecture room. Printed System Usability Scales were printed and participants were asked to fill them out using a pen or pencil.

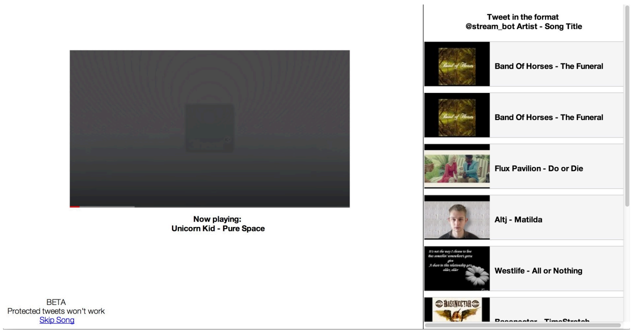
From 2:30 to 3:15 p.m. the experimenter ran two 15-minute tests (with a 15 minute gap in-between each test). Users were shown one interface design of StreamBot and, if they so decided, were free to use it and request a song. After 15 minutes, participants were asked to fill out a System Usability Scale regarding the interface they had just seen.
After a 15-minute break, participants were shown another variant of the StreamBot interface and asked to fill out another System Usability Scale along with additional qualitative questions regarding the whole StreamBot system and not just its interfaces. Users were not told how to operate or navigate StreamBot in each case.


Tweet in the format @stream_bot Artist - Song Title

Now playing: Unicorn Kid - Pure Space

BETA
Protected tweets won't work
Skip Song

Interface 1


Tweet in the format @stream_bot Artist - Song Title

Now playing: Unicorn Kid - Pure Space

BETA
Protected tweets won't work
Skip Song

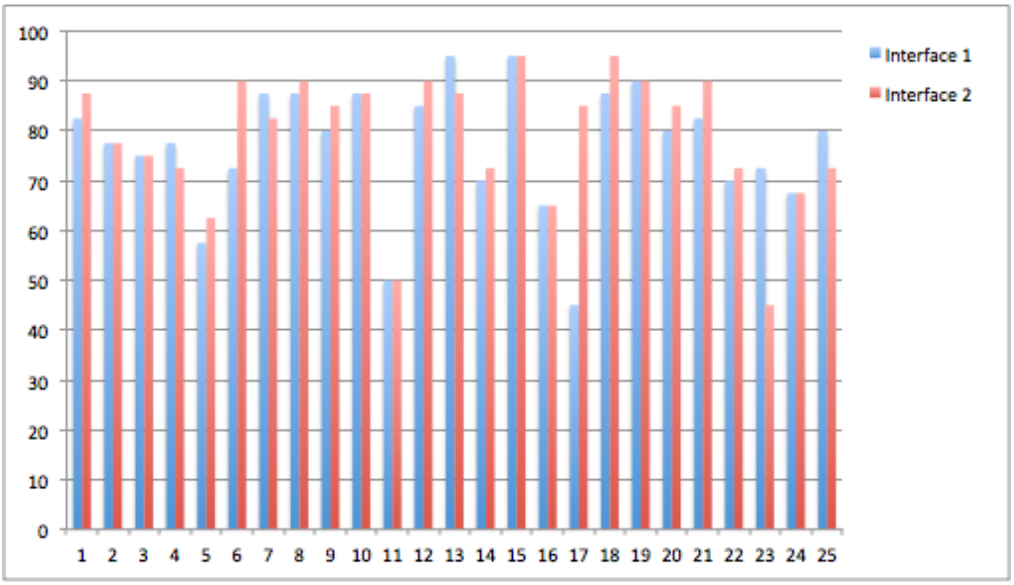
Band Of Horses - The Funeral
Band Of Horses - The Funeral
Flux Pavilion - Do or Die
Altj - Matilda
Westlife - All or Nothing

Interface 2

Research Results

In order to process results from the System Usability Scale, each question's score contribution had to be calculated, ranging from 0 to 4 (Field, Hole, 2003). For positively worded items (odd numbered questions), the score contribution is the scale position minus 1 (x - 1). For negatively worded items (even numbered questions), the score contribution is 5 minus the scale position (5 - x). To then get the overall score, the sum of score contributions is multiplied by 2.5. Making scores range from 0 to 100 in 2.5 point increments and easier to compare to other studies and global values.

Post-hoc tests revealed that, overall, average System Usability Scores fell into the 78th

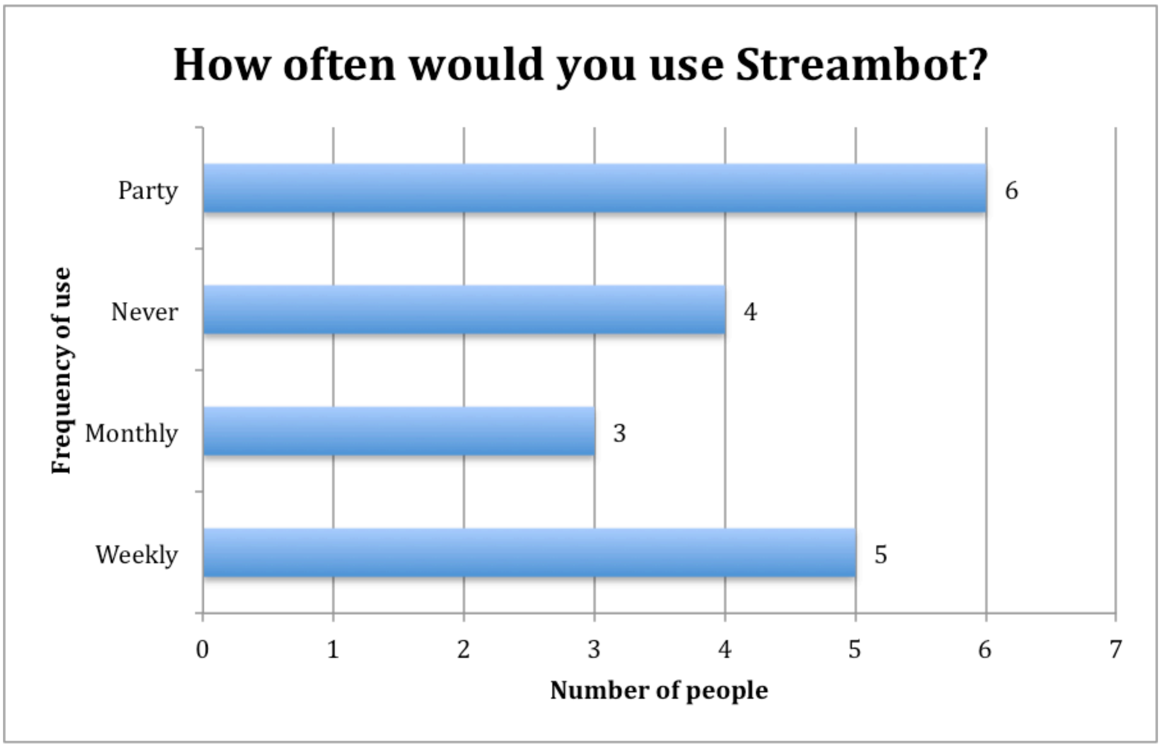


(Score - 76.8) and 85th (Score - 78.9) percentiles of the curved grading scale of SUS Scores (Sauro, Lewis, 2012). Both scores are also above the global mean of 68, showing that participants of the study found both interfaces adequate.

The above figure shows System Usability Scores for each user (the higher the score, the better). Inspection of this suggests that there is very little variance between scores from the two interfaces. A repeated measures t-test (scores from first interface and scores from second interface) was performed on the data. There was no significant effect found *(Correlation = .636, p < 0.001, r = 0.19)*.

	Mean score	**Mean (SD)**
Interface 1	76.8	12.78
Interface 2	78.9	13.29
(Maximum score 100)		

When participants were asked how often they would use either variant of StreamBot, the majority would consider use at a party (its intended primary use) while others would use it on a less frequent weekly or monthly basis. It should be mentioned that 100% of those asked who answered that they would never use StreamBot do not own a Twitter account and thus, would have no use for this system.



During the 15-minute testing period of interface 1, 6 songs were requested (0.4 tweets per minute). 1 input error was detected by the system (and gracefully handled) in which a request was submitted in an incorrect format (no hyphen between Artist and Title of song).

Interface 2 received 7 requests within the same time frame (0.46 tweets per minute). Again, 1 input error was caught in which the user had input plain text rather than a song choice (“hello”).

Participants were asked that where possible, to provide three positive and three negative aspects that they thought of each interface. Twelve responses indicated that interface 1 displayed positive aspects because it was simple, easy and had a plain interface. Six responses within the same study listed the same reasons as a negative aspect. Four evaluations mentioned that videos/elements on screen were too small.

When asked to evaluate interface 2, five answers showed that the ability to see more videos/details on screen was a positive. Correspondingly, seven responses approved of bigger video sizes.

Similar to interface 1, four responses listed the interface being plain/easy/simple as a negative. Two responses indicated that integration with Twitter was a positive aspect while five also listed it as a negative.

Participants were asked whether it was more important to show or hide who requested a song. Out of nineteen answers, eleven said that they would prefer song choices to remain anonymous, raising concerns of revealing embarrassing song choices. Four said that it should be optional depending on the event and four said found it important to always show who requested a song, as it would discourage spamming attacks.

Finally, participants were asked how they found Twitter as an input method in comparison to more traditional methods such as keyboard and mouse. Responses included "Nice idea", "Prefer mouse", "Easy to use, excellent", "You need Twitter" and "Cool, can use many devices at the same time." Comments were largely positive, praising StreamBot's ease of use and ability to accommodate multiple users simultaneously. Five comments from seventeen were critical; one user wrote that they preferred using a mouse whilst the other four remarked that it was a negative that Twitter was a requirement to use StreamBot. Although StreamBot received much praise in this part of the questionnaire, answers were rarely relevant to the question. Many appreciated the ease of use, simplicity and ability but only seven answers gave any sort of comparison between input methods.

From the seven participants who provided a comparison, all said that using Twitter is easier, simpler or more inclusive.

Results Discussion

It is possible that scores from the two conditions will correlate (because the data in each condition come from the same people and so there should be some consistency in their responses). In this case there was a fairly moderate correlation coefficient (r = .636), this is highly statistically significant as p < .001. With just a 2-point difference between means and a standard deviation of 12.8 and 13.3 we can say there is a significant overlap between results.

Interface 2 was more likely to receive positive scores (t = -.944) than interface 1. However there is a 35% (Sig. = .355) probability of this being purely by chance. This is well above Fisher's criterion of .05, showing this particular result to be negligible.

There were only 25 participants in total so this test only had the power to detect a very strong effect. Around 4% (r = 0.189) of participants were affected by using the different interfaces. Although this effect is highly statistically significant, the size of the effect is very small and so represents a trivial finding.

We can therefore say that in terms of System Usability Scores, there was no meaningful difference between interfaces.

Rate of song requests between interfaces show many similarities. Six requests and one input error for interface 1 compared to seven requests and one input error for interface 2. Due to one interface being shown after another, it is possible that participants may have carried knowledge of how to use the system on to the second test. The t value shows this that this is indeed the case. Between error rates and tweet requests per minute, there is a very little margin of difference between the interfaces when discussing what encourages user participation.

Mixed qualitative results show that opinions on StreamBot's various interfaces are highly subjective: some users like the simplicity, and others do not.

However, four participants did mention that they thought the on-screen elements were too small with no users referencing this as a positive point. StreamBot could therefore benefit from a larger video element.

When using interface 2, seven users indicated that they saw the larger videos shown in this display as a positive aspect of their user experience. Furthermore, five users highlighted the ability to see more details as a positive aspect of interface 2. As corresponds with the results from interface 1, users value seeing more on screen. Contrary to that, four users still found the simplicity, or plainness, of the interface to be a negative.

With regards to the use of Twitter as an input method, two users felt that this was a positive aspect of StreamBot, while five indicated that this was a negative aspect. It is important to note however that from the five users who saw this as negative, four did not have a Twitter account: these users could have been biased against using Twitter more generally.

The same opinions being shown as both positives and negatives for each interface gives an undetermined 'better' interface.

The large majority of users preferred that song choices remain anonymous within StreamBot's design. While it was not a main objective within this paper, results fall in line with research from J. Seeburger et al, (see section 2.5), most users preferred anonymity in their choices for reasons such as hiding embarrassing choices whilst a minority felt like they weren't sharing too much information and were happy to show themselves.

When asked to compare Twitter as an input method to a keyboard/mouse - only seven participants provided an actual comparison. Responses did include negatives and critiques to the system but failed to answer the question. Due to how this question was answered, a definite result cannot be concluded as to whether there is a clear and 'better' input method. It is unclear if all participants understood the questions meaning or not and thus, we cannot draw real results for this question from the answers provided. Although a large amount of answers were irrelevant to this specific question, they still provided useful, largely positive feedback showing that Twitter could be considered as a very viable input method.

Conclusions

Results from the System Usability Scale show that average scores fell into the 78th and 85th percentiles of the curved grading scale of SUS scores. Proving that both StreamBot interfaces were highly satisfactory and usable. Users had very little trouble navigating and using the system and qualitative results also show a high rate of positive comments.

While both interfaces scored highly, results from this study found that the difference between the two interfaces and design patterns was negligible in terms of how well users interacted with the system, it is highly important to note that the two interfaces had only a limited set of variables that were altered between each test. Another research question was to investigate factors that would influence engagement rates: again, no significant difference was detected within usage rates. From these results, it would not be possible to say conclusively that design is not important: indeed, the majority of users remarked upon the simplicity of StreamBot's design, so obviously this is still a concern. This study focused mainly on layout and presentation, it is asserted here that modifying other variables might have greater impact on the use of StreamBot. Moreover, the inherent nature of StreamBot as a music application means that its main function is auditory in nature, not visual. This means that the user is perhaps more focused on what they are hearing, over what they are looking at and design becomes a less important factor. Another concern in this study is that participant's answers did not sufficiently address some questions - e.g. limited or vague responses - so results are limited in this sense.

5. Reflection & Evaluation

5.1 Introduction

This dissertation has set out to find out how we can display and interact with a public display. Due to such mass and varied usage of public displays, it would be difficult to find an all-encompassing, singular behaviour, trend or style. This dissertation has tried to focus in on smaller areas to provide an answer. This chapter will now reflect upon how well aims and objectives were met, acknowledge some limitations to this project and consider future development that has been identified as a result of this completed dissertation.

5.2 Fulfilment of Aims and Objectives

This study's research goals were to:

- To provide future researchers with recommendations regarding how to display and interact with a shared music player interface on a public display
- To design a visually appealing and enticing user interface for displaying a shared music interface on a public display.
- To explore in detail how usable Twitter is as an input method in the context of a shared public display

This dissertation believes that these goals have been fulfilled. While an exhaustive list of do's and don'ts were not identified during this project, valuable and usable feedback data was found regarding user's preferences to a public display. Future researchers can build upon these results.

Both interfaces tested scored above the global mean of 68 on the System Usability Scale, showing that participants of the study found both interfaces more than adequate to use.

A system was successfully built; users could concurrently input their requests to StreamBot using Twitter with little to no guidance. Again, this goes in tandem with System Usability Scores - showing that Twitter is usable and a viable alternative input method in regards to a public display.

Three research questions were asked in an attempt to meet the above goals:

- What are users' preferences regarding layout for StreamBot on a public display?
- What factors encourage/discourage a potential user from using a public display for a shared music application?
- How can Twitter be used as a real-time input method and how does this compare to *a keyboard and/or mouse?*

In regards to users' preferences regarding layout, qualitative feedback shows that users prefer bigger elements on screen. All other data provides inconclusive definitive results. Many liked the simplicity and basic layouts while many disliked them for the same reasons. Results and analysis from quantitative data had shown that both interfaces were treated the same. Results were far too similar to discern a greater overall option; both interfaces had very close System Usability scores and tweets per minute.

This dissertation shows that at least from the variables and design patterns chosen, they do not influence a potential user either way to use StreamBot.

Finally, as shown throughout this dissertation, Twitter can be used as an alternative real-time input method - a successful project was built around this idea.

5.3 Limitations of Research

StreamBot is a music application, tested with multiple interfaces with variations in layout and design. All of these are inherently subjective; song choices and individual preferences to layout and design may have affected an individuals result. Choices seen as a positive to one user may be seen as a negative to another.

It was the intention of this dissertation to compare Twitter as an input to traditional input methods. However, results were inconclusive, as one core question seemed to confuse a sample of those tested. Many misinterpreted how they were supposed to answer and compare input methodologies and so this hampered results and final analysis. True statistical samples could not be taken with SPSS from the questionnaire as not everyone answered every question. Participants were advised to answer every question but due to the nature of qualitative research, they can choose to answer poorly (or not at all). Feedback could still be used but could not hold any significant result.

I consider that although certain limitations to the data gathering impacted the quality of said data, the results collected have delivered a strong enough argument to achieve the aims and objectives of this study.

5.4 Future development

As StreamBot has developed and evolved, research and feedback from users show that future development could include:

- Implementation of stretch goals

  StreamBot’s incremental development was split into core, release and stretch goals. Core and release goals were met. Future development could include integration of stretch goals (Appendix 7.1).

- Further, independent study comparing Twitter as an input method to a keyboard/mouse.
- The results of this study show that changing layouts does not affect how an application is used. This area would be worth further investigation to assess whether this is an anomaly within just music applications or the norm.

5.5 Personal Reflections

This dissertation has provided me with some invaluable skills and experience - particularly in research methods, writing skills and statistical analysis. I have always been particularly interested in interaction design and this project allowed me to indulge in that. Completing this research has shown me just how important it is to gather data rather than make assumptions as to what the user may prefer.

6. References

- Schmidt, G. S., Staadt, O. G., Livingston, M. a., Ball, R., & May, R. (2006). A Survey of Large High-Resolution Display Technologies, Techniques, and Applications. IEEE Virtual Reality Conference (VR 2006), 223–236. doi:10.1109/VR.2006.20
- Brignull, H., & Rogers, Y. (2003). Enticing people to interact with large public displays in public spaces. Proceedings of INTERACT.
- Finin, T., & Tseng, B. (2007). Why We Twitter : Understanding Microblogging.
- MacMillan, G., 80% of UK users access Twitter via their mobile, TwitterUK, https://blog.twitter.com/en-gb/2014/80-of-uk-users-access-twitter-via-their-mobile (accessible on 15/04/2014)
- Hole, G., & Field, A., How to Design and Report Experiments, Sage, 2003

- Kuhn, M., Wattenhofer, R., & Welten, S. (2010). Social audio features for advanced music retrieval interfaces. Proceedings of the international conference on Multimedia (MM ’10)
- Kukka. H, Oja. H, & Kostakos. V. (2013). What makes you click: exploring visual signals to entice interaction on public displays. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13). ACM, New York, NY, USA, 1699-1708
- Liu, K., & Reimer, R. (2008). Social playlist: enabling touch points and enriching ongoing relationships through collaborative mobile music listening. In Proceedings of the 10th international conference on Human computer interaction with mobile devices and services (MobileHCI '08). ACM, New York, NY, USA, 403-406.
- MacDonald, L. W. (1999). Using color effectively in computer graphics. IEEE Computer Graphics and Applications, 19. doi:10.1109/38.773961
- Müller, J., Alt, F., Michelis, D., & Schmidt, A. (2010). Requirements and design space for interactive public displays. Proceedings of the international conference on Multimedia(MM '10). ACM, New York, NY, USA, 1285-1294
- Sauro, J., & Lewis, J.R., Quantifying the User Experience, Elsevier, 2012
- Seeburger, J., Foth, M., & Tjondronegoro, D. (2012). The sound of music: sharing song selections between collocated strangers in public urban places. Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia - MUM ’12, 1.
- Yahoo Design Pattern Library, https://developer.yahoo.com/ypatterns/
- Komulainen, S., & Oyj, N. (2010). Social Music Services in Teenage Life – A Case Study, (April).

7. Appendix

7.1 Core, release and stretch goals

Users will be treated like customers for an agile project and as such, may request further feature requests. If deemed viable, further features may be implemented.

Application development is split into core, release and stretch goals to ensure that a minimum of work is done in the event of any interruption to workflow. Due to the nature of using iterative development, it's difficult to predict what will be finished when so features are split into minimum, final and extra sets.

Core - Features for StreamBot that are guaranteed for final submission

- Parse Twitter Requests - incoming tweets need to be parsed and recognized as an Artist and song requested.
- Save incoming Tweets to a database - Once tweets have been parsed, save requests to a local database for further analysis and typical usage patterns.
- Live updating/no page reloads - Using AJAX requests so that when new requests come in there is no need for page reloads and new content automatically appears within the application.
- Playlist interfaces - Create multiple interfaces and layouts to show what songs have been requested and what is queued to play next.
- Input from website - the ability to make a song request using traditional input methods like keyboard and mouse.

Release - Features for StreamBot that are planned for final submission

- Handle parsing errors/exceptions - some users may use the system unexpectedly, there needs to be a system in place to gracefully handle such errors should they arise.
- Skip/master user functionality - A master user function could be available so if for example, the host of party wants to skip or remove a song from the playlist then they have the ability to do so.
- Save extra tweet meta data - Save more tweet information. For example, the time and date the request was made, how popular the request is etc.

Stretch - Features for StreamBot that may be included depending on time constraints

- Idle screen - display an idle screen while StreamBot is not in use
- Tweet back to users - Tweet back a confirmation of their request once it has been processed
- Find alternative sources - StreamBot will use Youtube for audio sources. Should a specific request be unavailable then look at alternative sources (e.g. Soundcloud) for the audio.
- Recommended/similar songs - When a song is playing, display other songs that the party/user may also be interested in.

7.2 Questionnaire

Two copies of the questionnaire were given to each participant. Marked interface 1 and 2 appropriately.

STREAMBOT SYSTEM USABILITY SCALE

Feedback can also be left via twitter @stream_bot or email dh91@hw.ac.uk

Name:
Age:
Gender:
I own a twitter account (Circle appropriate) - YES/NO

For each item identified below, circle the number to the right that best fits your judgment of its quality.

@stream_bot	Scale				
	Disagree				Agree
I think that I would like to use this system frequently.	1	2	3	4	5
I found the system unnecessarily complex.	1	2	3	4	5
I thought the system was easy to use.	1	2	3	4	5
I think that I would need the support of a technical person to be able to use this system.	1	2	3	4	5
I found the various functions in this system were well integrated.	1	2	3	4	5
I thought there was too much inconsistency in this system.	1	2	3	4	5
I would imagine that most people would learn to use this system very quickly.	1	2	3	4	5

I found the system very cumbersome to use.	1	2	3	4	5
I felt very confident using the system.	1	2	3	4	5
I needed to learn a lot of things before I could get going with this system.	1	2	3	4	5

List the most positive aspect(s) of this system:

1. ____

2. ____

3. ____

List the most negative aspect(s) of this system:

1. ____

2. ____

3. ____

These questions apply to all iterations of Streambot:

How often would you use streambot?

What other alternatives would you/do you use to streambot? (Youtube,spotify etc)

How important do you feel it is to show who requested a song?

Compared to traditional input methods (keyboard, mouse), how do you find twitter as an input for a collaborative public display?