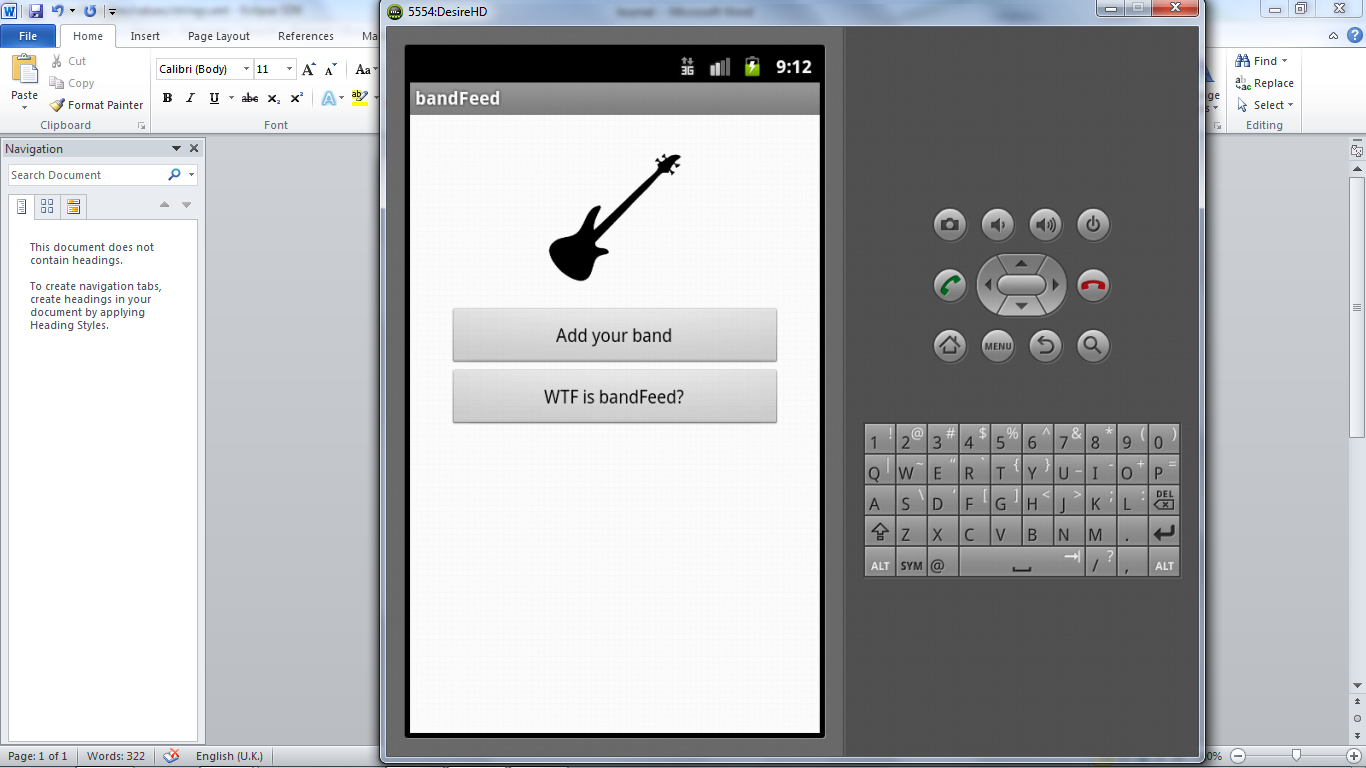
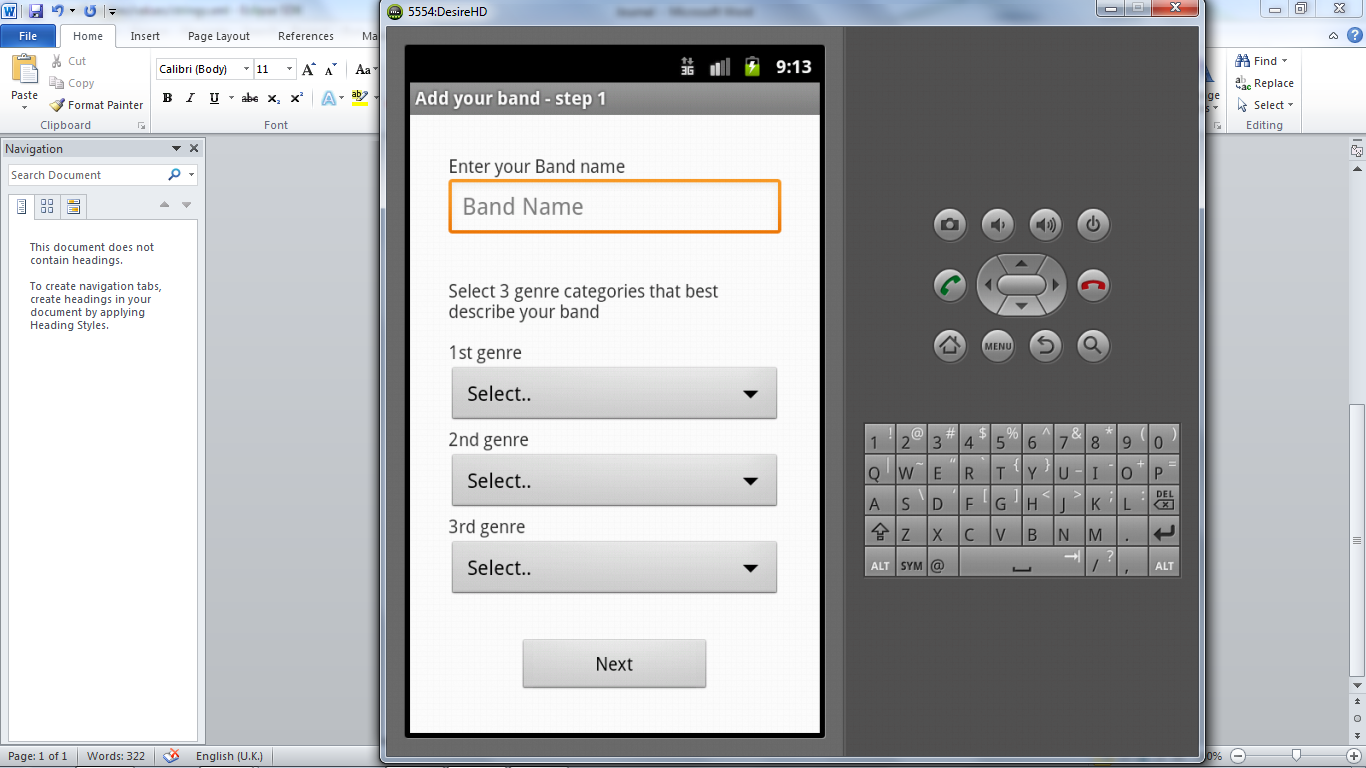
30/07/2012 – Saw Martin, discussed ideas. Started work on an Android app. <http://www.zazzle.co.uk/bass_guitar_bassist_card-137539554697266595> for app icon.

**Prototype 1**

01/08/2012 – Started working on Band app. Implemented the Main class which asks the user whether they wish to add their new band, view WTF (about page), or if the user has already added a band their band will be listed to be selected. These are all 3 buttons which take the user to another activity. ‘Add New Band’ button takes you to the activity\_step\_one.xml (the first step of setting up a band profile), ‘WTF’ button takes you to the activity\_wtf.xml (which is an about page).

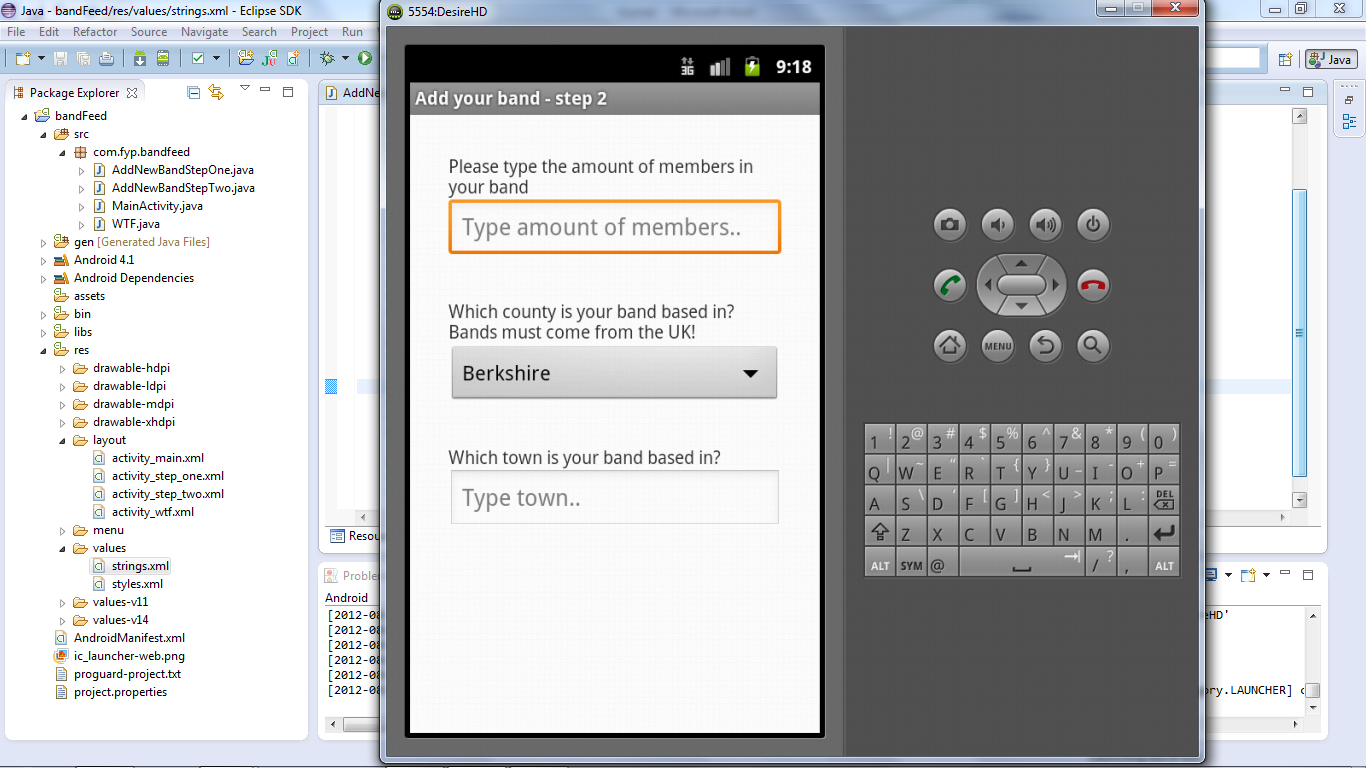


02/08/2012 – Added Spinners (drop down boxes) to the activity\_step\_one.xml to allow the user to select 3 genres that best describe their band. I had to work out how to incorporate a standard Java ArrayList to a spinner (formed in XML). The result was that I had to use an adapter to add the String objects held in the arrayList to the spinner. Latter into the project I may draw the genre data from an external course such as SoundCloud. A button was used which displays the text ‘Next’, this takes the user to the next step of setting up a band profile (see below).



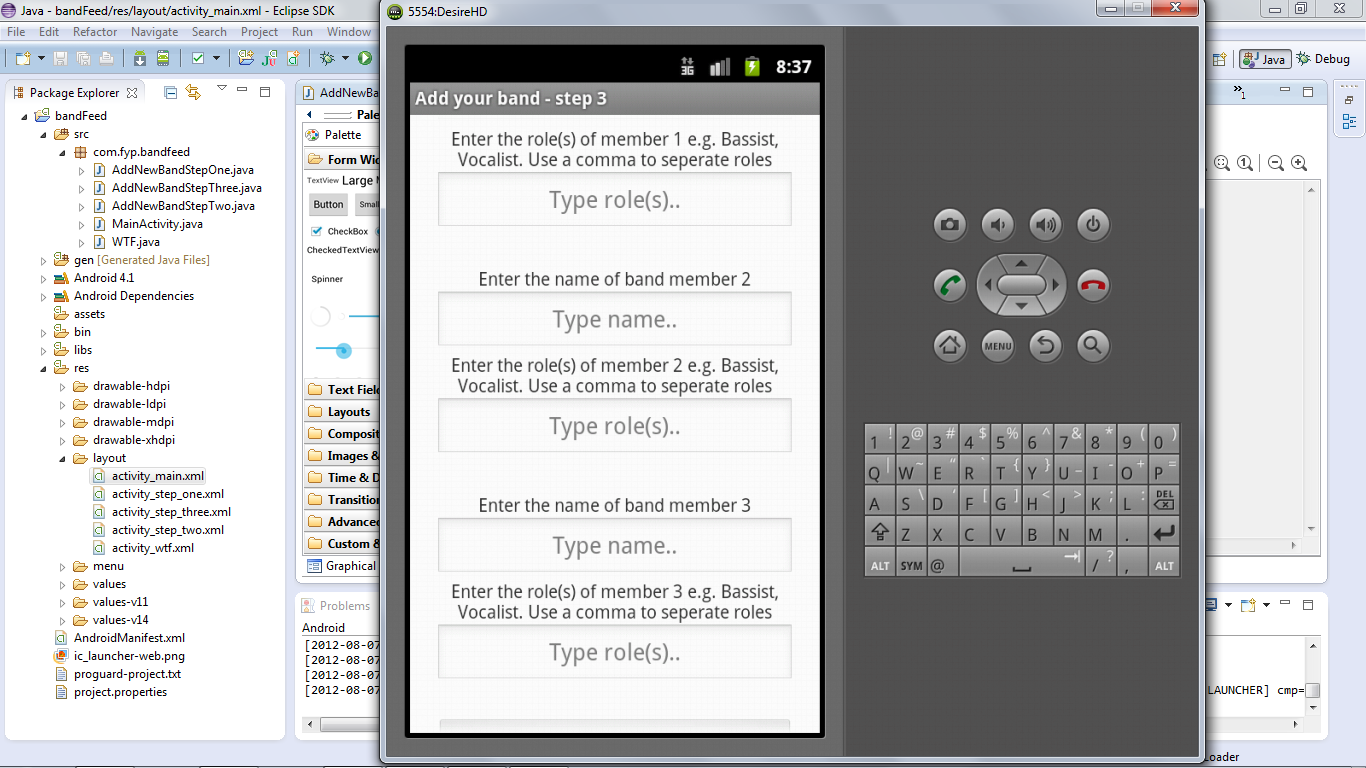
This new activity (activity\_step\_two.xml) asks for the amount of members in the band (using an editText for input). This activity leads on from activity\_step\_one.xml (as mentioned above). Like step one this activity includes a spinner linked to a java array using an adapter to display all the counties of the UK. This will be used for users to find bands in their area. Again, later into the project I may draw up the county data from an external source.

03/08/2012 – set up a Git repository. Continuation with activity step two of the setting up a new band profile process, adding an edit box for the user to provide their town that the band is based in.



05/08/2012 – ‘next’ button added to the bottom of step 2 (imagine the picture above with a button at the bottom just like step 1). Spent the rest of the day researching something I didn’t need to.. I tried looking for a way to add variables to XML but found out you can’t the hard way as these files are static. Consequently I discovered a tutorial on creating a dynamic layout in java which allows me to create / change activities on-the-fly. This is useful if the next activity depends on the user to input something.

07/08/2012 – Put the tutorial to use (mentioned in the previous journal entry). Step 2 of the band profile set-up process requires the user to input the amount of band member which is then used to create the Step 3 activity which requests the user to input the names of members in the band and their role(s) e.g. are they the guitarist or vocalist. The ‘next’ button in Step 2 uses the Intent method putExtra() which passes the amountOfMembers to Step 3 where the needed amount of editText’s can be created to enter the band member’s details.  
This is the first layout that uses scrollView. This allows the user to scroll down the page with their finger should they have many members in the band.

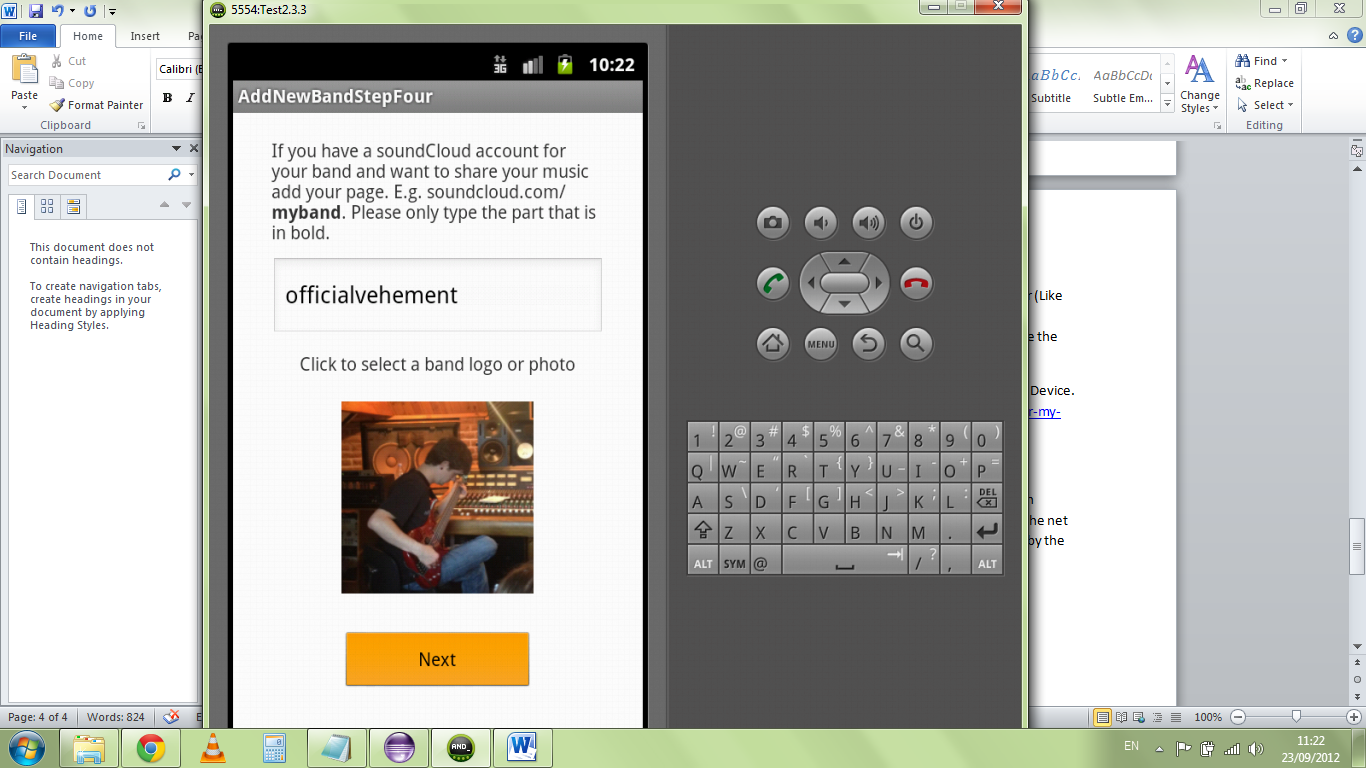


17/08/2012 – Start to think about changing the way I go from one activity to another during the setting up of band profile process. I’ve currently been carrying across important data held in variables to each activity (un-bundling the extras) regardless of whether the next activity needs it. The reason behind this is so that the very last step in this process will then write all these variables to file. I’ve decided to stop carrying over these variables and instead write them to file straight away as the steps are carried out. I have spent time researching and have found this page: <http://www.anddev.org/working_with_files-t115.html>

While researching I have discovered that you can not use the standard Java filewriter because.. “Each \*.apk File that is installed on the Emulator/Device gets its own User-ID from the Linux System. This ID is the key to the sandbox of the application. This 'sandbox' protects the application (and its files) from other bad apps, that i.e. want to manipulate the files we created in a bad manner (Like writing into them: "Whos reads this is... dumb ! Hhahaha").” Taken from <http://www.anddev.org/working_with_files-t115.html> But writing to an SD Card can be done the usual java way!

I’ve also done to some research into how to select an image from the gallery on the Android Device. <http://stackoverflow.com/questions/2507898/how-to-pick-a-image-from-gallery-sd-card-for-my-app-in-android>

23/09/2012 – Progress has been slow as I’ve found myself having to do a lot of reading which involved buying a another book due to the last not being detailed enough and examples on the net not being very intuitive. I have finally got step4 of the band profile process complete where by the user now inputs a biography, their url to SoundCloud and select a picture.



Once the user has completed the final step of the band process, the ‘Next’ button is clicked and the program creates a new folder under the name of the band and saves the image selected and creates a file to hold the data the user has inputted. The files in this folder will now be used by the band profile to display the appropriate data.

The reason for creating a new folder per profile was that the user may be several bands and each band profile would at least consist of a data file and an image file with the possibility of the amount of files increasing as the project evolves. I felt this kept things cleaner and with the possibility of many users creating band profiles the server would becoming quite cluttered without each band having their designated folder.

Trying to get the program to write this files correct was a real pain and with little and unintuitive examples the task took a long time.

25/09/2012 – Changed Main Activity to a dynamic layout to cater for the possible band profile that could have been created previously (the program will dynamically search for existing folder and then create buttons accordingly). This meant creating id’s for elements such as the profile buttons (amount depending on how many band profile has been created so far), the ‘Create a band profile’ and ‘about’ buttons. The id’s along with a string value (name of the button) are kept in a SparseArray (was suggested by the compiler that this would be a more efficient approach than using as HashMap. So that the appropriate profile can be open when clicking a button etc. because I don’t know the name of the users band(s) I can’t just simply use if (id == 6) then open the Bandit’s profile. The global ‘onClick()’ method requires an id number as an argument and I won’t know these id numbers that are applied to specific band profile. (Hope this make sense)!

07/10/2012 – Spent the last week research and experimenting with RabbitMQ. Had really problems trying to get the RabbitMQ client libraries to run through Command Prompt but got there in the end. Need to remember to include the library’s file locations in the Class Path, libraries also inside the current java package and use the following commands:

javac -cp rabbitmq-client.jar Send.java

java -cp .;rabbitmq-client.jar Send

Also run into problems with trying to include the management plugin when running the RabbitMQ server. Got this working in the end also!!

I learnt that RabbitMQ is really great so masses of messages to be sent masses of consumers. I learnt that when a message is sent out in a Fan exchange (or any other type of exchange to be exact (if the same message is being sent to many)) that there is in fact only one message saved to disk and it’s simply references to this message in all the Queues to all the consumers, this saves a hell of a lot of space. I also learnt that queues that haven’t been used within 10 seconds go into hibernation mode where the amount of RAM currently being used is dramatically reduced. Queues can remain open for as long as you like i.e. years if need be but also have choice of killing queues should the consumer never connect to the save within a certain amount of days etc.

I learnt about the different types of routing that RabbitMQ can do e.g. topic routing. These url’s became very useful:

Help setting up RabbitMQ with the management plugin  
<http://lostechies.com/derekgreer/2012/03/05/rabbitmq-for-windows-introduction/>  
Tutorials  
<http://www.rabbitmq.com/getstarted.html>  
Routing Topologies for performance  
<http://blog.springsource.org/2011/04/01/routing-topologies-for-performance-and-scalability-with-rabbitmq/>

08/10/2012 – Had meeting with Martin. I had doubts whether using a MOM would be suitable for my project and he said he had no doubts and said it would be fine. We discussed User Authentication / Password encryption too, where he advised to looking to BCrypt. Martin emailed me the following urls:  
<http://codahale.com/how-to-safely-store-a-password/>  
<http://en.wikipedia.org/wiki/SHA-2>  
<http://en.wikipedia.org/wiki/Cryptographic_hash_function>

Also sent off my project proposal review (whatever it’s called) today.

10/10/2012 – Creating two new logic drives on my home desktop, one to run RabbitMQ Server and one to run a MySQL Database. Using MySQL because it is what I learn’t in year two. I found the following tutorial to help connect to my Android application:  
<http://www.androidhive.info/2012/05/how-to-connect-android-with-php-mysql/>  
//Haven’t used this yet

sql!2M

Great website explaining PHP, MySQL, Apache etc.  
<http://www.webdevclips.com>

Spent the whole day trying to get the server’s ports forwarded to find that for some reason they’re not displaying on my own network yet everyone outside the network can access the servers. STRANGE!!

you might just want to try, as a test, reserving (explicitly mapping mac address to ip address) an ip address for the server in your router. There should be a dhcp section in your router where you can do this.

15/10/2012 – Tried the network again including explicitly mapping the mac address to the ip address of the machine running the servers and using the dhcp section in the router but still no luck. Now I’m just having to make sure I’m viewing the servers as localhost when working on my network and using my smartphone connected to 3G to check the changes are actually being made.

Today I’m working on connecting to a MySQL database where I will be adding band profiles to.

Created a band profile table with the following code:

CREATE TABLE bprofile (

band\_name VARCHAR(50) PRIMARY KEY,

genre1 VARCHAR(50) NOT NULL,

genre2 VARCHAR(50) NOT NULL,

genre3 VARCHAR(50) NOT NULL,

county VARCHAR(50) NOT NULL,

town VARCHAR(50) NOT NULL,

members INT(11) NOT NULL,

soundc\_link VARCHAR(50) NOT NULL,

pic\_link VARCHAR(50)NOT NULL,

created\_at TIMESTAMP default NOW(),

updated\_at TIMESTAMP

);

Created PHP scripts to be used to manipulate the MySQL device.

Run into problems with HTTPResponse hanging on emulator and phone even though <http://192.168.0.3:3401/bandFeed/api/create_profile_local.php> is accessing the php file. Need to try a simpler example.

17/10/2012 – It would seem the reason behind the networking problems I have i.e. I can’t access the servers from within my LAN but in WAN can access would be to do with the fact that my Router doesn’t support NAT LoopBack!

Got database to work with WAMP server on the same laptop as the emulator being run using the ip address 10.0.2.2 . I have no idea why I can’t connect to the database when WAMP is running on my other machine (the machine to be the server machine) using the ip address 192.168.0.3:3401 . Maybe it’s a port issue but I can access the database on the server machine via my laptop which is weird. I will have to carry out some more testing to see if I can nail it down.

19/10/2012 – Finally got the Threading working in Android 4.0. Originally the app worked in 2.3 but when trying to run it in 4.0 it just crashed. I sorted this by using ASyncTask. Considering renting a Virtual server and space due to the problems I’ve had with my router. I’m about to move house and it usually takes a while for the Internet Provider to set up the new line. Paying for a virtual server will allow me to continue you my work at the University.

So yeah got band profiles being sent to my Database which currently running on the same laptop. I’ve just started working on making search queries to the database to find band profiles and then the messaging will commence.

29/10/2012 – Virtual Server confirmation through. Currently setting server up. It’s a linux Ubuntu 12.04 LTS server. Using Putty and WinSCP to perform SSH commands. Had to install additional packages into the server such as Apache, MySQL and php.

<http://www.sussex.ac.uk/Users/mfb21/osnet/ex/1/lab1.html>

<http://www.bsd.org/unixcmds.html>

<https://help.ubuntu.com/community/ApacheMySQLPHP>

30/10/2012 – Had trouble connecting to mysql database as I was using the IP address of the server with the port of the database. While investigating to see whether it could have been due to the server’s firewall it clicked in my head that the database port doesn’t actually need to be open to external clients/interfaces. Because I have already made a request to a php document successfully the php document can communicate with the database via ‘localhost’ and there is no need to try and access through the IP address with the database port.

I now have the database accessible from my phone. At the moment I’m using the ‘root’ user with it’s password. User accounts will set up in later prototypes where users will have restricted privileges etc. But for now using the ‘root’ is fine.

## 07/11/2012 – Installed RabbitMQ on Virtual Server. Managing the Broker

## As an administrator, start and stop the server as usual for Debian using

## invoke-rc.d rabbitmq-server *stop/start/etc*.

To stop the server or check its status, etc., you can use rabbitmqctl (as an administrator). It should be available on the path. All rabbitmqctl commands will report the node absence if no broker is running.

* Use rabbitmqctl stop to stop the server.
* Use rabbitmqctl status to check whether it is running.

rabbitmqctl — command line tool for managing a RabbitMQ broker

COMMANDS  
<http://www.rabbitmq.com/man/rabbitmqctl.1.man.html>

* The web UI is located at: http://*server-name*:55672/mgmt/
* The HTTP API and its documentation are both located at: [http://*server-name*:55672/api/](http://server-name:55672/api/)

Struggled to get RabbitMQ up and running but found this tutorial which seemed to solve it

http://www.theprogrammer.co.za/wordpress/2012/08/installing-rabbitmq-on-ubuntu-12-04-lts/

RabbitMQ management available via http://81.169.135.67:55672 and <http://bandfeed.co.uk:55672>

Found an interesting Q & A plus a youtube video  
<http://rabbitmq.1065348.n5.nabble.com/How-many-queues-can-one-broker-support-td21539.html>

<http://www.youtube.com/watch?v=wDk6l3tPBuw>

Someone messages left on a forum  
“I use rabbitmq for my android app. It's just a matter of sending a byte array between the client and the server. Then it's just a matter of reading the bytes into the appropriate fields. (both server and client) This may be a bit of work but it's really fast and efficient.”

Using Android / RabbitMQ tutorial

http://simonwdixon.wordpress.com/2011/06/03/getting-started-with-rabbitmq-on-android-part-1/

Need to research Handlers and Runnables.  
<http://developer.android.com/reference/android/os/Handler.html>  
<http://developer.android.com/reference/java/lang/Runnable.html>

Understanding Threads  
<http://buildmobile.com/understanding-and-utilizing-threads-in-android/>

<http://rabbitmq.1065348.n5.nabble.com/Use-a-mysql-databse-table-as-the-provider-for-rabbitmq-queue-td22617.html>

15/11/2012

HeartBeat

getting reliable TCP connections over 3G can drive you batty...

You will need to implement heartbeat - very simple - to verify if the connection is alive as in our experience it is very difficult to maintain a connection over 3G.  Pay close attention to error handling  - specifically ShutdownSignalException.

<http://rabbitmq.1065348.n5.nabble.com/RabbitMQ-as-communication-server-for-mobile-devices-td23385.html>

<http://grokbase.com/t/rabbitmq/rabbitmq-discuss/124kfphyrw/rabbitmq-as-android-service-keep-alive-heartbeat>

<http://lists.rabbitmq.com/pipermail/rabbitmq-discuss/2012-January/017138.html>

"If heartbeats are enabled on a connection, the server periodically sends   
heartbeat frames to the client and waits for its response.  The heartbeat   
frames are effectively empty messages.  If either side misses a certain   
number of heartbeats, the connection is closed.  This is usually used to   
a) check that the other side is still online and hasn't crashed without   
closing the connection or the socket and b) to bypass certain routing   
devices that timeout a connection due to inactivity. "

My messages on RabbitMQ Forum about my Architecture  
<http://rabbitmq.1065348.n5.nabble.com/Exchange-using-MySQL-td23396.html>  
Queue created when new user signs up, deleted when leaves, inactive queues deleted after so long. Queue name is keep with user account, bindings are made and unmade when user subscribes.

Who does what in messaging  
<http://stackoverflow.com/questions/12597006/rabbitmq-exchanges-queues-and-bindings-who-does-setup-what>

16/11/2012 – Need to look at Queue Durability, Queues can only be durable if their exchange is durable so that bindings can be restored in case of RabbitMQ going down

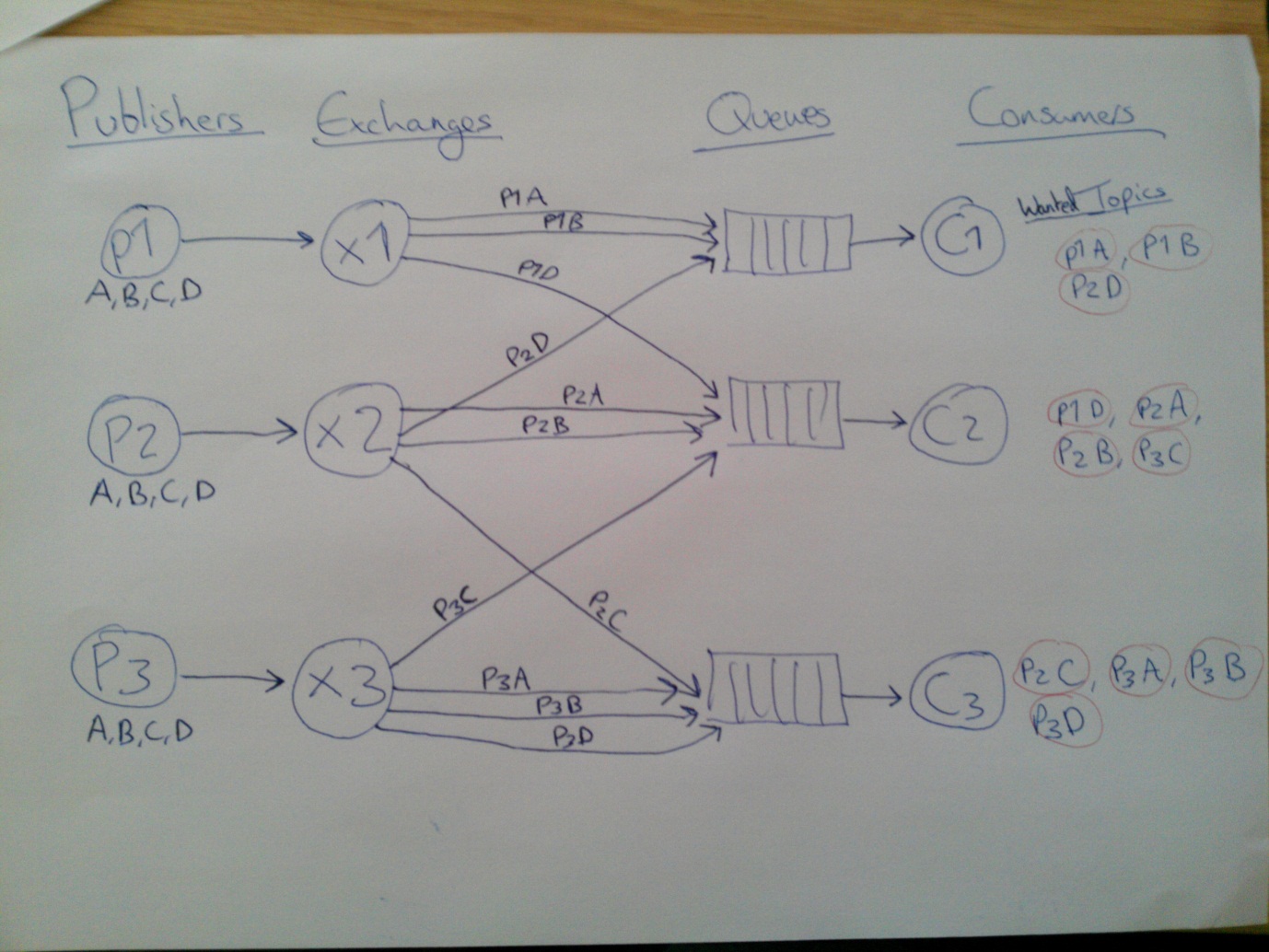
<http://rubyamqp.info/articles/durability/#durability_of_queues>

19/11/2012 – The application now allows for queues, exchanges and bindings to happen. If the app is started for the first time the user is asked to create username which is then stored on SD card. If the app is starting from now on the app will not request a username because it will find it on the SD Card. This will need to be changed so that it just stores it in the cache of the program. Currently there is no user account created in the database. But a queue is created in RabbitMQ, a personal queue to the user.

When the above is implemented the user will also need to provide a password along with the username when creating an account. The user will then be able to access their account only held on the database. So… eventually when the app is started for the first time the user will be asked to provide a username and password or to create an account, if successful the users name will be stored in cache so that the username is easily accessible when requested for binding to band exchanges.

When a band account is created there is now an exchange created in RabbitMQ. When a user browses the database of profile the user can click subscribe to the profile creating a bind between their personal queue and the band exchange.

The messages architecture I have designed is shown in the following picture I drew.



Had real issues with the Asynctask threading option it won’t let me call execute() on it from the onCreate() method within the FeedAll activity. But it will work if I set the activity up first then call execute() from a button. Nevermind is still has the functionality I need.

Also now got messages being sent using some test classes for the time being called TestFanOut (even though I’m not using fanout anymore) and FeedAll. You can currently send as many messages from TestFanOut and the messages will be queued up waiting until FeedAll is opened and the ‘Get’ button is clicked. All the messages are then consumed and viewed in FeedAll. If there are no messages to be consumed then a Toast pops up saying “No new messages”.

**21/11/2012 –** Now that messaging is working I’ve decided to work creating proper profiles for the user to view. Currently the profile just displays the text inputted during a band profile creation, so on which is actually missing. I’m now having to learn a bit more PHP, so that I can concatenate variables as one String. <http://stackoverflow.com/questions/4990901/use-counter-as-part-of-variable>

Learnt about arrays etc.

**26/11/2012**

Sending photos

<http://stackoverflow.com/questions/5415390/sending-pictures-to-a-web-server>

I’ve used this!

<http://www.coderzheaven.com/2011/04/25/android-upload-an-image-to-a-server/>

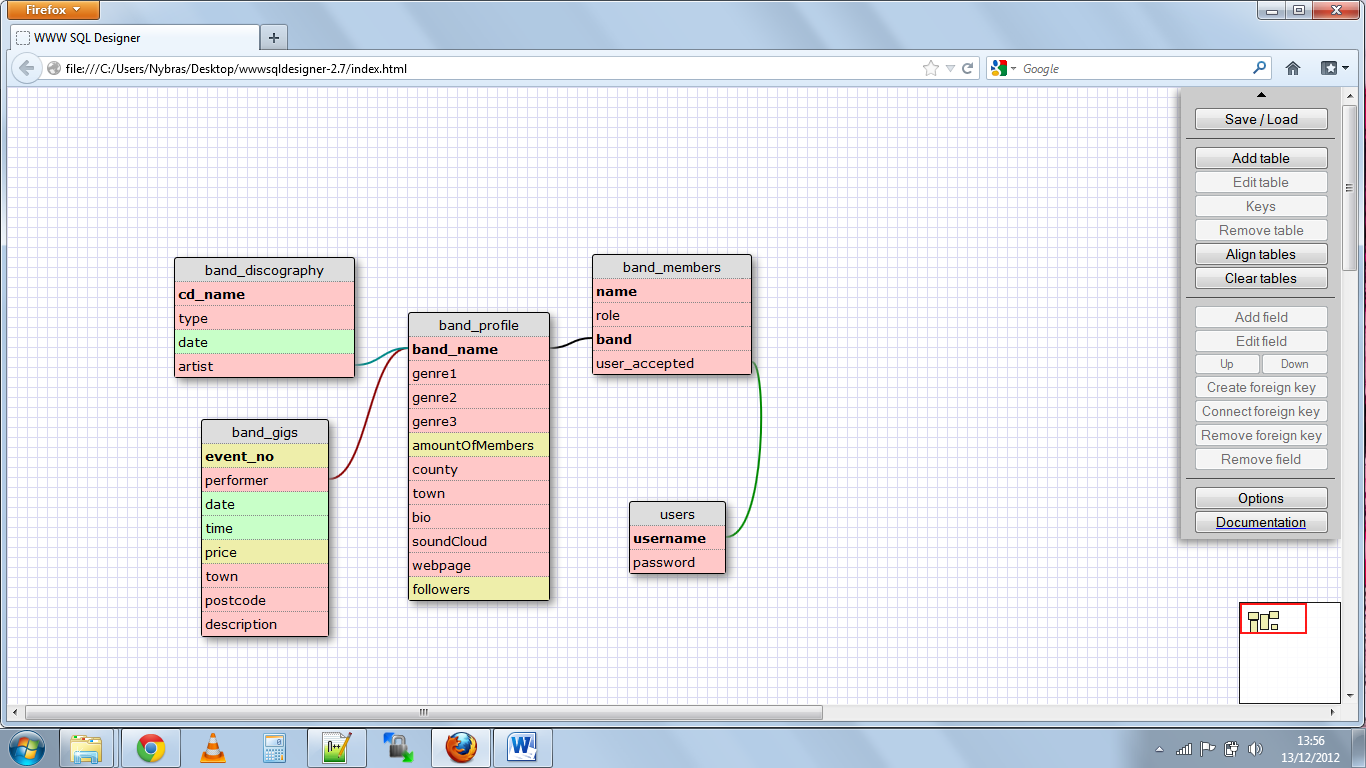
<http://www.coderzheaven.com/2012/03/29/uploading-audio-video-or-image-files-from-android-to-server/>

**29/11/2012 -** [**http://misha.beshkin.lv/android-swipe-gesture-implementation/**](http://misha.beshkin.lv/android-swipe-gesture-implementation/)

**So I can create swipe to left and right for earlier versions of android**

**06/12/2012 –** Images can be uploaded and pulled from server for band profiles. Work started on creating user accounts (Database updated with new busers table).

**07/12/2012** – Big changes made.. Classes renamed for easier readability. Slight clean up in code in some classes. Users can now create an account with password which is held in database. This is not secure though! User can then login into their account, again not secure where the users account name is stored in phone memory so that the user doesn’t have to log in everytime. To login the user provides their name and password where the database then checks for a match. If theres a match then the program turns a Boolean to true and the users name is saved to phone memory.



Now that the program is taking shape and the fundamental features are working I decided to create the whole database. The design for this is above. I used wwwsqldesigner 2.7

I use the Database the way I do through PHPbecause I didn’t at first realise you can save queries actually in the database. (Stored routines)

**19/12/2012 – Lots of changes and bug fixes.**I’m now using SharedPreferences instead of saving the user’s preferences to the phone’s memory. This cuts down on the amount of code written and the process should use a lot less resources. Also meant I could reduce the amount of items I was having to load and unload in a bundle as I went through activities. SharedPreferences allows me to save some primitive objects which can be called upon and edited from any Activity.. I wish I had discovered this much earlier on, would have saved a lot of time. Never mind at least Inow know how to save to phone memory should I ever need to.

The primitives saved in SharedPreferences also remain saved even if the app is closed or fully exited. This is a great way for the user to be “always logged in”. I will able to create a function to allow the user to log out which will just be a case of deleting these primitives. I’ve now made the program to check these SharedPreferences when first running. If there is no primitives saved then the user is not currently logged in. and will be redirected to the Login Activity. Once the user has logged in by checking the user’s username and password against the database the users details are then placed into the SharedPreferences.

When the user logs in there is a check made to see if the user is a member of any bands. If so, the user’s bands are also placed into SharedPreferences. The Main Activity then checks to see if the user is a member of a band from these SharedPreferences and if so the user has the option of sending messages. If the user is not in a band then he/she can only receive messages (from the bands they have subscribed to).

When the user uses the option to send out a message to the band’s followers the user has the option to select which band they wish to send the message out from should they be in a few bands. The possible bands are listed in a drop-down menu.The editText Box the user users to type their message will increase in size as the user types their long message. To make sure all elements are still visible should the message get very long a ScrollView is used on the Activity. I may need to look into the size limit of messages that can be sent using RabbitMQ. To enable the user to type a message quickly the EditText Box has inputTypes AutoCorrect, AutoComplete set.

I used this webpage to understand SharedPreferences <http://stackoverflow.com/questions/3150513/keep-user-state-in-android>

I had a bug where subscribing to a band was failing. This was due to the new method of using SharedPreferences had not be implemented for this Activity yet.

**28/12/2012** – Using Apache TomCat instead of Apache HTTP might have been better for high demand traffic with my database. “Servlets provides middle-tier support for web applications. Sophisticated business logic or database connectivity management can be handled smoothly due to the persistence of the Servlet runtime environment. For example, database connections can be shared and reused by different clients.” Not really in the scope of this project at the moment where there isn’t a large scale of users using the project.   
On the topic of possible future improvements, remember stored routines in database, can talk about how RabbitMQ is highly extensible and can cope with a high demand of users. Having each service e.g. RabbitMQ, MySQL Database, WebServer as their own machine/server would be necessary should the app had a large amount of users.  
PHP not necessary the most safest way of doing things?

**I made a little PHP script that keeps logs of any connections made to the server e.g. that an exchange named ‘bob’ has been created or ‘bob’ has sent the message: “……blah…. Blah”. This will help to keep track of the messages and connections made through the RabbitMQ server and MySQL Database making it easier to debug.**

**02/01/2013 –** Created the option to log out of app, this means the user can now have multiple logins or share their device with another user who wishes to use their own account. This log out feature is located in the app’s main menu which also includes ‘About’ (taken off the main Activity), ‘Send Feedback’ (yet to be implemented), ‘Settings’ (yet to be implemented). This menu can be access throughout the majority of the app by pressing the ‘menu’ hardkey/softkey on the user’s device.

I was experiencing visual glitches when a new Activity is started and the soft-keyboard was open on the previous activity. It was like as if the new activity had completed the starting sequence before shutting down the keyboard. To stop these glitches I made a method which will close the Keyboard well in advance of opening a new activity. This fixed the issue!

I noticed a bug in the SignUp activity which allowed users to input a username and passwords as blank. I rectified this by checking the strings were not equal to ””.

**03/01/2013** – Might want to use this tutorial to guide me in created a table\_layout for Discography & Gigs. <http://mobile.tutsplus.com/tutorials/android/android-sdk_table-layout/>

Also came across this too to help with password & security. <http://android-developers.blogspot.co.uk/2013/01/verifying-back-end-calls-from-android.html>

**24/01/2013** – Could have used Java instead of PHP to run web server. Working started on allowing the user to update their band profile. They just need to long press the part of the profile they wish to update and the appropriate activity opens up allowing them to update each part. Only made a start, still lots to do. PHP script has been made to allow the update of data to the database.

**26/01/2013** – Discovered the RabbitMQ HTTP API to query the RabbitMQ database. This has meant I can return a list of name/value pairs of bindings so the user can see what they have or haven’t subscribed to. I will also need to use this api to query the amount of followers (subscriptions) each band has.

When making a GET request I had to refer to this for some guidance. <http://stackoverflow.com/questions/10647631/rabbitmq-http-api-request-unauthorized>

Learnt how to use check boxes and users are now able to select from our categories of message they wish to receive. Once they’ve made their selection the necessary bindings are created. Users can also unselect any categories at any time and the binding will be deleted.

**28/01/2013** - Added a function in menu to allow user to send feedback to developer relating to the current activity being viewed.

Need to remember to set permission for a folder on the server to access it!

Managed to get Menu system working for all versions of Android :-)

I changed MainActivity so that it now uses a XML layout to overcome swipe listener issues.

**02/02/2013 –** Needed to root device. <http://stackoverflow.com/questions/4867379/android-eclipse-ddms-cant-access-data-data-on-phone-to-pull-files>

**05/02/2013** - Added the first "All" feed.

Make messages appear with band name and picture. Add picture to cache for quicker loading next time. Used someone else’s code for displaying feeds in a ListView -<https://github.com/thest1/LazyList/tree/master/res>

Made the size of buttons consistent throughout app.

**06/02/2013** - Changed feeds.xml to 'username'.xml so that the user name login in and out and have the appropriate feedlist used according to the username currently signed in.

Added the rest of the Feeds.

<http://www.ibm.com/developerworks/xml/library/x-tipmvdom/index.html>

<http://stackoverflow.com/questions/4903758/android-how-to-refresh-listview-contents>

SendMessages now only includes a spinner to select the band for which to send a message out to followers from, when the user is linked to more than two bands. This is done using: view.setVisiblility(View.GONE) which sets the view in the XML layout to be hidden.

Added menus to all theFeeds.

Swipe now works well! I used the following url to help me with this: <http://stackoverflow.com/questions/8326599/horizontal-swipe-not-detected-in-scrollviews-parent/8327453#8327453>

A check to see if band name is already in use during profile creation (at StepOne), is now implemented!

Search results are now presented in a Listview with a pic, just like the feeds are presented. To do this I just edited the LazyAdpater and it’s linked classes.

Users can now send requests to be linked to a band profile. The user creates a temporary exchange and sends a message to all existing queues associated with the band i.e. existing linked members. The members then receive this link request in their ‘All Feed’ and can approve the link or just cancel. Should the link be approved the band\_members table in the database is updated with the user\_accepted column now containing the users name against the appropriate member.

**07/02/2013 – Server Log architecture is actually quite good as it is a similar arch to what the big apps would use. Companies like Facebook would have separate servers for their logs, where they could then use data mining/machine learning methods to understand their users etc.**