Let’s tell a story

Server Key - XAea9lg8(open fire)

1. User name creation(ask once and make sure user knows it can never be changed once set)( <https://www.youtube.com/playlist?annotation_id=annotation_195647079&feature=iv&list=PLe60o7ed8E-TztoF2K3y4VdDgT6APZ0ka&src_vid=nHmPErsoe64>)

Google sign in (<https://developers.google.com/identity/sign-in/android/sign-in#configure_google_sign-in_and_the_googleapiclient_object>)

1. Saving said info in an online database
2. Checking newer user name’s with those inserted into the database(<https://www.google.ca/search?q=use+google+email+with+my+domain&oq=use+google+email&aqs=chrome.1.69i57j0l5.27120j0j4&sourceid=chrome&ie=UTF-8#q=cross+check+user+name+android>)
3. Use google sign in email or user name as FK (<https://developers.google.com/cloud-messaging/upstream>)
4. Send sqlite data(story, genre) to server (might not need to if I just send data to clients instead) (<https://developers.google.com/cloud-messaging/upstream>)
5. Then send said data to the next person to post along with all the people that already wrote in the story
6. Store details for future authentication ([https://developers.google.com/identity/smartlock-passwords/android/store-credentials)](https://developers.google.com/identity/smartlock-passwords/android/store-credentials)
7. Notification also send to said users to notify them of this update(<https://developers.google.com/cloud-messaging/notifications>)
8. Cloud messaging([https://developers.google.com/cloud-messaging/](https://www.youtube.com/watch?v=m3VHQE86juo))

How to develop client server application

<https://www.youtube.com/watch?v=m3VHQE86juo>

Login and signup sqlite

<https://www.youtube.com/watch?v=NT1qxmqH1eM>

Send data to a remote DB

<https://www.youtube.com/watch?v=FL37oah1k8k>

Save data into server

<https://www.youtube.com/watch?v=xGr5voR6B1Y>

Using XMPP(<https://en.wikipedia.org/wiki/XMPP>)

Make the page for telling the story a messaging activity, where only the person with permission to post has an active “POST” button.

XMPP will deal with contacts and updating the story across all users.

My layout will make the story activity look more like a story and less like a messaging app.

Might have to use XMPP along with GCM because XMPP does not work when the app is closed as it would constantly and wastefully pull the server

Validate mobile phone numbers for user sign ups

<https://www.quora.com/How-do-services-like-WhatsApp-validate-mobile-phone-numbers-for-user-sign-ups-subscribers>

use phone number for sign up

<https://stackoverflow.com/questions/28161341/sign-up-with-a-phone-number-using-parse>

Whatsapp signup

<https://www.youtube.com/watch?v=aDjTb4Or_Jw>

Parse

<https://www.youtube.com/user/parseplatform/videos>

<http://docs.parseplatform.org/android/guide/>

<http://docs.parseplatform.org/android/guide/#installation>

Parse example

<https://www.youtube.com/watch?v=-AB_uu_Lj8Q>

OTP

<https://stackoverflow.com/questions/39481860/parse-how-to-get-user-to-sign-up-with-phone-number>

[https://www.simplifiedcoding.net/android-sms-verification-app-phone-verification-with-otp/](https://stackoverflow.com/questions/39481860/parse-how-to-get-user-to-sign-up-with-phone-number)

using Fire base(FREE)

<https://www.youtube.com/watch?v=Xn0tQHpMDnM>

<https://firebase.google.com/docs/auth/android/phone-auth> (phone verification)

(July 7, 2017)

Work on activity to activity navigation

Set up “pass” and “pass a start”

Local stories

Set up grid view to save new stories

Redesign create a story activity

Create fragment for My stories, local stories, invited stories(done)

Remove auto rotate(done)

Remove multi window view support or figure out how to make it work

Pressing back in the main activity should close the app

Either make start a story page portrait only or add scrollview

onResume , onStart…(Activity lifecycle)

Not Needed

Story

SimpleFragmentPagerAdaptor

**package** com.letstellastory.android.letstellastory.adapter;  
  
**import** android.content.Context;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.BaseAdapter;  
**import** android.widget.TextView;  
  
**import** com.letstellastory.android.letstellastory.R;  
**import** com.quickblox.chat.model.QBChatDialog;  
  
**import** java.util.ArrayList;  
  
*/\*\*  
 \* Created by dozie on 2017-07-12.  
 \*/***public class** StoryDialogAdapters **extends** BaseAdapter{  
 **private** Context **context**;  
 **private** ArrayList<QBChatDialog> **qbChatDialogs**;  
  
 **public** StoryDialogAdapters(Context context, ArrayList<QBChatDialog> qbChatDialogs) {  
 **this**.**context** = context;  
 **this**.**qbChatDialogs** = qbChatDialogs;  
 }  
  
  
 @Override  
 **public int** getCount() {  
 **return qbChatDialogs**.size();  
 }  
  
 @Override  
 **public** Object getItem(**int** position) {  
 **return qbChatDialogs**.get(position);  
 }  
  
 @Override  
 **public long** getItemId(**int** position) {  
 **return** position;  
 }  
  
 @Override  
 **public** View getView(**int** position, View convertView, ViewGroup parent) {  
 View view = convertView;  
  
 **if**(view == **null**){  
 LayoutInflater inflater = (LayoutInflater) **context**.getSystemService(Context.***LAYOUT\_INFLATER\_SERVICE***);  
 view = inflater.inflate(R.layout.***story\_grid\_layout***, **null**);  
  
 TextView story,genre;  
 story = (TextView) view.findViewById(R.id.***storyView***);  
 genre = (TextView) view.findViewById(R.id.***genreView***);  
 }  
 **return** view;  
 }  
}

**package** com.letstellastory.android.letstellastory.adapter;  
  
**import** android.content.Context;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.BaseAdapter;  
**import** android.widget.TextView;  
  
**import** com.letstellastory.android.letstellastory.R;  
**import** com.quickblox.chat.model.QBChatDialog;  
  
**import** java.util.ArrayList;  
  
*/\*\*  
 \* Created by dozie on 2017-07-12.  
 \*/***public class** StoryDialogAdapters **extends** BaseAdapter{  
 **private** Context **context**;  
 **private** ArrayList<QBChatDialog> **qbChatDialogs**;  
  
 **public** StoryDialogAdapters(Context context, ArrayList<QBChatDialog> qbChatDialogs) {  
 **this**.**context** = context;  
 **this**.**qbChatDialogs** = qbChatDialogs;  
 }  
  
  
 @Override  
 **public int** getCount() {  
 **return qbChatDialogs**.size();  
 }  
  
 @Override  
 **public** Object getItem(**int** position) {  
 **return qbChatDialogs**.get(position);  
 }  
  
 @Override  
 **public long** getItemId(**int** position) {  
 **return** position;  
 }  
  
 @Override  
 **public** View getView(**int** position, View convertView, ViewGroup parent) {  
 View view = convertView;  
  
 **if**(view == **null**){  
 LayoutInflater inflater = (LayoutInflater) **context**.getSystemService(Context.***LAYOUT\_INFLATER\_SERVICE***);  
 view = inflater.inflate(R.layout.***story\_grid\_layout***, **null**);  
  
 TextView story,genre;  
 story = (TextView) view.findViewById(R.id.***storyView***);  
 genre = (TextView) view.findViewById(R.id.***genreView***);  
 }  
 **return** view;  
 }  
}