```java
/* Help.java
   =============================================================================
                          Josh Talley and Daniel O'Donnell
                                 Dulaney High School
                       Mobile Application Development 2016-17
   =============================================================================
   Purpose: This activity displays all of the help information using a
   recycler view.
*/
package com.fbla.dulaney.fblayardsale;

import android.databinding.DataBindingUtil;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.LinearLayoutManager;
import android.view.View;

import com.fbla.dulaney.fblayardsale.databinding.ActivityHelpBinding;

public class Help extends AppCompatActivity implements View.OnClickListener
{
    ActivityHelpBinding mBinding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_help);

        mBinding = DataBindingUtil.setContentView(this, R.layout.activity_help);
        mBinding.done.setOnClickListener(this);
        mBinding.listHelp.setLayoutManager(new LinearLayoutManager(this));
        mBinding.listHelp.setAdapter(new HelpAdapter());
        setSupportActionBar(mBinding.myToolbar);
    }

    @Override
    public void onClick(View v)
    {
        switch(v.getId())
        {
            case R.id.done:
                this.finish();
                break;
        }
    }

    @Override
    public void onBackPressed()
    {
        this.finish();
    }
}

```

```java
/* MySales.java
   =============================================================================
                         Josh Talley and Daniel O'Donnell
                                Dulaney High School
                      Mobile Application Development 2016-17
   =============================================================================
   Purpose: This activity lists all of the items you have for sale. It allows
   you to delete any item, or look at their comments.
*/
package com.fbla.dulaney.fblayardsale;

import android.content.Intent;
import android.databinding.DataBindingUtil;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.LinearLayoutManager;
import android.util.Log;
import android.view.View;

import com.fbla.dulaney.fblayardsale.controller.MySalesController;
import com.fbla.dulaney.fblayardsale.databinding.ActivityMysalesBinding;

public class MySales extends AppCompatActivity implements View.OnClickListener {

    ActivityMysalesBinding mBinding;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_mysales);

        mBinding = DataBindingUtil.setContentView(this, R.layout.activity_mysales);
        mBinding.list.setLayoutManager(new LinearLayoutManager(this));
        MySalesAdapter adapter = new MySalesAdapter(this, this);
        MySalesController.AttachAdapter(adapter);
        mBinding.list.setAdapter(adapter);
        setSupportActionBar(mBinding.myToolbar);

        Log.d("MySales", "onCreate");
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()) {

            case R.id.comments:
                this.startActivity(new Intent(this, Comments.class));
                break;
            default:
                break;
        }
    }

    @Override
    public void onBackPressed()
    {
        this.finish();
    }

}

```

```java
/* AddSales.java
   =============================================================================
                          Josh Talley and Daniel O'Donnell
                                 Dulaney High School
                       Mobile Application Development 2016-17
   =============================================================================
   Purpose: This activity is used to add a new sale item.
*/
package com.fbla.dulaney.fblayardsale;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.databinding.DataBindingUtil;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Build;
import android.os.Bundle;
import android.provider.MediaStore;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.Toast;

import com.fbla.dulaney.fblayardsale.controller.MySalesController;
import com.fbla.dulaney.fblayardsale.databinding.ActivityAddsalesBinding;

import java.io.InputStream;
import java.util.UUID;

import com.fbla.dulaney.fblayardsale.model.*;
import com.microsoft.windowsazure.mobileservices.table.MobileServiceTable;

public class AddSales extends AppCompatActivity implements View.OnClickListener {
    ActivityAddsalesBinding mBinding;

    private MobileServiceTable<SaleItem> mSaleItemTable;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_addsales);

        if (!FblaLogon.getLoggedOn()) {
            Toast.makeText(this, "Unable to connect to Azure. Please try again.", Toast.
LENGTH_LONG).show();
            finish();
            return;
        }

        mSaleItemTable = FblaLogon.getClient().getTable(SaleItem.class);

        mBinding = DataBindingUtil.setContentView(this, R.layout.activity_addsales);
        FblaPicture.setLayoutImage(mBinding.activityAddsales);
        setSupportActionBar(mBinding.myToolbar);
        mBinding.gallery.setOnClickListener(this);
        mBinding.camera.setOnClickListener(this);
        mBinding.back.setOnClickListener(this);
        mBinding.finish.setOnClickListener(this);
        mBinding.another.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.gallery:
                // Ask for permission first
                if (android.os.Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                    int permissionCheck = ContextCompat.checkSelfPermission(this, Manifest.permission.READ_EXTERNAL_STORAGE);
                    if (permissionCheck != PackageManager.PERMISSION_GRANTED) {
                        // Should we show an explanation?
                        if (ActivityCompat.shouldShowRequestPermissionRationale(this, Manifest.permission.READ_EXTERNAL_STORAGE)) {
                            // Explain to the user why we need to read the contacts
                        } else {
                            ActivityCompat.requestPermissions(this,
                                    new String[]{Manifest.permission.READ_EXTERNAL_STORAGE}, 0);
                        }
                        return;
                    }
                }

                Intent i = new Intent(Intent.ACTION_PICK, android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
                Log.d("CameraFragment", "Starting GALLERY Intent");
                this.startActivityForResult(i, 1);
                break;
            case R.id.camera:
                // Ask for permission first
                if (android.os.Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                    int permissionCheck = ContextCompat.checkSelfPermission(this, Manifest.permission.CAMERA);
                    if (permissionCheck != PackageManager.PERMISSION_GRANTED) {
                        // Should we show an explanation?
                        if (ActivityCompat.shouldShowRequestPermissionRationale(this, Manifest.permission.CAMERA)) {
                            // Explain to the user why we need to read the contacts
                        } else {
                            ActivityCompat.requestPermissions(this,
                                    new String[]{Manifest.permission.CAMERA}, 0);
                        }
                        return;
                    }
                }

                Intent j = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
                this.startActivityForResult(j, 2);
                break;
            case R.id.another:
                addItem(v);
                this.finish();
                this.startActivity(new Intent(this, AddSales.class));
                break;
            case R.id.finish:
                addItem(v);
                this.finish();
                break;
            default:
                this.finish();
                break;
        }
    }

    @Override
    public void onBackPressed()
    {
        this.finish();
    }

    // Add a new item to the database.
    private void addItem(View view) {
        if (!FblaLogon.getLoggedOn()) return;

        // Create a new item from the SaleItem model.
        final SaleItem item = new SaleItem();
        item.setId(UUID.randomUUID().toString());
        item.setName(mBinding.editname.getText().toString());
        item.setUserId(FblaLogon.getUserId());
        item.setDescription(mBinding.editdesc.getText().toString());
        String sPrice = mBinding.editprice.getText().toString();
        if (sPrice == null || sPrice.equals("")) item.setPrice(0);
        else item.setPrice(Float.parseFloat(mBinding.editprice.getText().toString()));
        Bitmap b = FblaPicture.GetPictureFromView(mBinding.picture);
        if (b != null) {
            item.setPicture(b);
        }

        // Save the item to the database over the internet.
        AsyncTask<Void, Void, Void> task = new AsyncTask<Void, Void, Void>() {
            @Override
            protected Void doInBackground(Void... params) {
                try {
                    mSaleItemTable.insert(item);
                    Log.d("AddSales:insert", "Created item " + item.getName());
                    runOnUiThread(new Runnable() {
                        @Override
                        public void run() {
                            item.setAccount(FblaLogon.getAccount());
                            MySalesController.addItem(item);
                        }
                    });
                } catch (Exception e) {
                    Log.d("AddSales:insert", e.toString());
                }
                return null;
            }
        };
        task.executeOnExecutor(AsyncTask.SERIAL_EXECUTOR);
    }

    @Override
    public void onActivityResult(int requestCode, int resultCode, Intent data) {
        // Results can come from the Camera, Gallery, or the Comments activity.
        if (resultCode == android.app.Activity.RESULT_OK) {
            if (requestCode == 2 && data != null) { // From Camera
                Log.d("AddSales", "Result from Camera");

                try {
                    Bundle extras = data.getExtras();
                    Bitmap image = (Bitmap) extras.get("data");
                    image = FblaPicture.ResizePicture(this.getApplicationContext(), image);
                    FblaPicture.LoadPictureOnView(mBinding.picture, image);
                } catch (Exception ex) {
                    Log.e("AddSales:camera", ex.getMessage());
                }
            } else if (requestCode == 1 && data != null) // Gallery
            {
                // Gallery
                Log.d("AddSales", "Result from Gallery");

                try {
                    Uri pickedImage = data.getData();
                    InputStream stream = getContentResolver().openInputStream(pickedImage);
                    Bitmap image = BitmapFactory.decodeStream(stream);
                    image = FblaPicture.ResizePicture(this.getApplicationContext(), image);
                    FblaPicture.LoadPictureOnView(mBinding.picture, image);
                } catch (Exception ex) {
                    Log.e("AddSales:gallery", ex.getMessage());
                }
            }
        }
    } // onActivityResult
}

```

```java
/* Comments.java
   =============================================================================
                         Josh Talley and Daniel O'Donnell
                                Dulaney High School
                      Mobile Application Development 2016-17
   =============================================================================
   Purpose: This activity allows you to both add new comments, review existing
   comments posted on a sale item, and delete comments.
*/
package com.fbla.dulaney.fblayardsale;

import android.content.Context;
import android.databinding.DataBindingUtil;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.LinearLayoutManager;
import android.util.Log;
import android.view.View;
import android.view.inputmethod.InputMethodManager;

import com.fbla.dulaney.fblayardsale.controller.CommentListController;
import com.fbla.dulaney.fblayardsale.databinding.ActivityCommentsBinding;
import com.fbla.dulaney.fblayardsale.model.ItemComment;
import com.microsoft.windowsazure.mobileservices.table.MobileServiceTable;

import java.util.UUID;

public class Comments extends AppCompatActivity implements View.OnClickListener {

    ActivityCommentsBinding mBinding;
    MobileServiceTable<ItemComment> mCommentTable;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_comments);

        mBinding = DataBindingUtil.setContentView(this, R.layout.activity_comments);
        mBinding.post.setOnClickListener(this);
        mBinding.list.setLayoutManager(new LinearLayoutManager(this));
        CommentsAdapter adapter = new CommentsAdapter(this, this);
        CommentListController.AttachAdapter(adapter);
        mBinding.list.setAdapter(adapter);
        setSupportActionBar(mBinding.myToolbar);

        mCommentTable = FblaLogon.getClient().getTable(ItemComment.class);

        Log.d("Comments", "onCreate");
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()) {

            case R.id.comments:
                this.finish();
                break;
            case R.id.post:
                if (!mBinding.newcomment.getText().toString().equals("")) {
                    addItem(v);
                }
                //this.finish();
                break;
            default:
                break;
        }
    }

    @Override
    public void onBackPressed()
    {
        this.finish();
    }

    // Add a new item to the database.
    private void addItem(View view) {
        if (!FblaLogon.getLoggedOn()) return;

        // Create a new comment from the ItemComment model.
        final ItemComment comment = new ItemComment();
        comment.setId(UUID.randomUUID().toString());
        comment.setComment(mBinding.newcomment.getText().toString());
        comment.setUserId(FblaLogon.getUserId());
        comment.setItemId(CommentListController.getItem().getId());

        // Save the item to the database over the internet.
        AsyncTask<Void, Void, Void> task = new AsyncTask<Void, Void, Void>() {
            @Override
            protected Void doInBackground(Void... params) {
                try {
                    mCommentTable.insert(comment);
                    Log.d("Comments:insert", "Created comment " + comment.getComment());
                    runOnUiThread(new Runnable() {
                        @Override
                        public void run() {
                            comment.setAccount(FblaLogon.getAccount());
                            CommentListController.addComment(comment);
                        }
                    });
                } catch (Exception e) {
                    Log.d("Comments:insert", e.toString());
                }
                return null;
            }
        };
        task.executeOnExecutor(AsyncTask.SERIAL_EXECUTOR);
        if (this.getCurrentFocus() != null) {
            InputMethodManager imm = (InputMethodManager) getSystemService(Context.
INPUT_METHOD_SERVICE);
            imm.hideSoftInputFromWindow(this.getCurrentFocus().getWindowToken(), 0);
            mBinding.newcomment.setText("");
        }
    }

}

```

```java
/* FblaLogon.java
   ============================================================================
                         Josh Talley and Daniel O'Donnell
                                Dulaney High School
                      Mobile Application Development 2016-17
   ============================================================================
   Purpose: This class establishes the connection with the Azure Mobile App server
   and the entire logon process. Part of the logon includes creating and/or fetching
   the user's Account information. It's done with the logon to make sure communication
   with the Azure server is working. This class extends AsyncTask because almost all
   of the login processes must be done in the background.
*/

package com.fbla.dulaney.fblayardsale;

import android.content.Context;
import android.content.SharedPreferences;
import android.os.AsyncTask;
import android.util.Base64;
import android.util.Log;
import android.webkit.CookieManager;
import android.webkit.ValueCallback;

import com.fbla.dulaney.fblayardsale.model.Account;
import com.google.common.util.concurrent.FutureCallback;
import com.google.common.util.concurrent.Futures;
import com.google.common.util.concurrent.ListenableFuture;
import com.microsoft.windowsazure.mobileservices.MobileServiceClient;
import com.microsoft.windowsazure.mobileservices.authentication.
MobileServiceAuthenticationProvider;
import com.microsoft.windowsazure.mobileservices.authentication.MobileServiceUser;
import com.microsoft.windowsazure.mobileservices.table.MobileServiceTable;

import org.json.JSONObject;

import java.net.URLDecoder;
import java.nio.charset.StandardCharsets;
import java.util.ArrayList;
import java.util.Date;

public class FblaLogon extends AsyncTask {
    final public static String AZUREURL = "https://fbla-yardsale.azurewebsites.net";

    private static boolean mLoggedOn = false;
    private static String mUserId = null;
    private static String mToken = null;
    private static MobileServiceClient mClient = null;
    private static Account mAccount = null;
    private static MobileServiceTable<Account> mAccountTable = null;

    private Context mContext;
    private ArrayList<LogonResultListener> mListeners = new ArrayList<LogonResultListener>()
;

    // Initialize the MobileServiceClient
    public FblaLogon(Context context) {
        mContext = context;
        if (getLoggedOn()) return;
        // Clear cookies now to support being able to logout easily later.
        clearCookies();
        try {
            mClient = new MobileServiceClient(AZUREURL, mContext);
        } catch (Exception e) {
            Log.d("FblaLogon:init", e.toString());
            mClient = null;
        }
    }

    @Override
    protected Object doInBackground(Object[] params) {
        doLogon();
        return null;
    }

    @Override
    protected void onPostExecute(Object result) {

    }

    public static boolean getLoggedOn() {
        return mLoggedOn;
    }
    public void Logoff() {
        mLoggedOn = false;
        clearCookies();
        setCache(mContext, null, null);
        mAccountTable = null;
        mAccount = null;
    }

    public static MobileServiceClient getClient() {
        return mClient;
    }

    public static String getUserId() {
        return mUserId;
    }

    public static Account getAccount() {
        return mAccount;
    }

    public static void setAccount(Account account) {
        mAccount = account;
    }

    // Once all of the logon processes are complete, notify any listeners
    private void onLogonSuccess() {
        mLoggedOn = true;
        for (LogonResultListener listener : mListeners) {
            listener.onLogonComplete(null);
        }
        Log.d("FblaLogon", "onLogonSuccess");
    }

    // Notify any listeners that the logon has failed.
    private void onLogonFailure(Exception e) {
        for (LogonResultListener listener : mListeners) {
            listener.onLogonComplete(e);
        }
        Log.d("FblaLogon", "onLogonFailure");
    }

    // This starts the whole logon chain of asynchronous calls.
    // So many things can happen in callbacks that I have to chain them all together.
    private void doLogon() {
        if (mClient == null) return;
        getCache(mContext);
        if (mUserId == null || mToken == null || isTokenExpired(mToken)) {
            // Missing or expired token, so need to kick off the Google+ logon process.
            googleLogon();
        } else {
            // The cached token seems to be good, so load the account with it.
            loadAccount();
        }
    }

    // This is the last part of the chain of asynchronous calls.
    // A successfully loaded account means the token actually works and we can talk to Azure.
    // So the callback from this one will call either onLogonSuccess or onLogonFailure.
    private void loadAccount() {
        MobileServiceUser user = new MobileServiceUser(mUserId);
        user.setAuthenticationToken(mToken);
        mClient.setCurrentUser(user);

        mAccountTable = mClient.getTable(Account.class);
        ListenableFuture<Account> account = mAccountTable.lookUp(mUserId);
        Futures.addCallback(account, new FutureCallback<Account>() {
            @Override
            public void onFailure(Throwable exc) {
                // See what kind of exception it is
                if (exc.getMessage().equals("{\"error\":\"The item does not exist\"}")) {
                    // The user is not in the table, so insert a new record for them.
                    Account act = new Account();
                    act.setId(mUserId);
                    mAccountTable.insert(act);
                    setAccount(act);
                    Log.d("FblaLogon:account", "AccountEdit Created");
                    onLogonSuccess();
                } else {
                    // Something else bad happened.
                    Log.d("FblaLogon:account", exc.toString());
                    onLogonFailure((Exception)exc);
                }
            }

            @Override
            public void onSuccess(Account result) {
                // Found the account record, so set it on the Data object.
                Log.d("FblaLogon:account", "onSuccess - "+result.getId());
                setAccount(result);
                clearCookies();
                onLogonSuccess();
            }
        });
    }

    // This is the longest chain of asynchronous processes.
    // It seems to use WebKit to perform the Google+ OAuth authentication via Azure.
    // If successful, chain it to loading Accounts.  Otherwise call onLogonFailure to notify
    // the listeners that it didn't work.
    private void googleLogon() {
        ListenableFuture<MobileServiceUser> mLogin = mClient.login(MobileServiceAuthenticationProvider.Google);
        Futures.addCallback(mLogin, new FutureCallback<MobileServiceUser>() {
            @Override
            public void onSuccess(MobileServiceUser mobileServiceUser) {
                Log.d("FblaLogon:login", "Logged On");
                setCache(mContext, mobileServiceUser.getUserId(), mobileServiceUser.
getAuthenticationToken());
                loadAccount();
            }

            @Override
            public void onFailure(Throwable throwable) {
                Log.d("FblaLogon:login", throwable.toString());
                onLogonFailure((Exception)throwable);
            }
        });
    }

    private void clearCookies() {
        // Clear cookies and cache before logging on
        CookieManager.getInstance().removeAllCookies(new ValueCallback<Boolean>() {
            @Override
            public void onReceiveValue(Boolean value) {
                Log.d("FblaLogon:cookies", "Cookies cleared");
            }
        });
    }

    // Helper functions to set and clear Shared Preferences cache.
    private void getCache(Context context) {
        SharedPreferences prefs = context.getSharedPreferences("auth", Context.MODE_PRIVATE
);
        mUserId = prefs.getString("usr", null);
        mToken = prefs.getString("tkn", null);
    }
    private void setCache(Context context, String userId, String token) {
        SharedPreferences prefs = context.getSharedPreferences("auth", Context.MODE_PRIVATE
);
        SharedPreferences.Editor editor = prefs.edit();
        if (userId == null) editor.remove("usr");
        else editor.putString("usr", userId);
        if (token == null) editor.remove("tkn");
        else editor.putString("tkn", token);
        editor.commit();
        mUserId = userId;
        mToken = token;
    }

    // Need to make sure the token has not expired.
    // https://blogs.msdn.microsoft.com/writingdata_services/2015/04/27/check-for-expired-
azure-mobile-services-authentication-tokens/
    private boolean isTokenExpired(String token) {
        // The expiration is embedded in the token.  This will rip apart the token to get
the expiration.
        try {
            // Get the string after the period
            String jwt = token.split("\\.")[1];
            // Decode the URL encoding
            jwt = URLDecoder.decode(jwt, "UTF-8");
            // Decode from Base64
            byte[] bytes = Base64.decode(jwt, Base64.DEFAULT);
            // Convert the byte array into a string, which is JSON
            String jsonString = new String(bytes, StandardCharsets.UTF_8);
            // Create the JSON object
            JSONObject json = new JSONObject(jsonString);
            // Extract the expiration from the JSON object
            jwt = json.getString("exp");
            // The value is the number of seconds since 1/1/1970 UTC (epoch)
            long exp = Long.parseLong(jwt) * 1000;
            boolean isExpired = System.currentTimeMillis() >= exp;
            Log.d("FblaLogon:Token", isExpired ? "Expired " : "Valid " + (new Date(exp)).
toString());
            return isExpired;
        } catch (Exception e) {
            Log.d("FblaLogon:Token", e.toString());
        }
        return false;
    }

    // Add a listener to call after logon is complete
    public void setLogonListener(LogonResultListener listener) {
        mListeners.add(listener);
    }

    // This is the interface to use on the logon callbacks.
    public interface LogonResultListener {
        void onLogonComplete(Exception e);
    }
}

```

```java
/* AccountEdit.java
   ============================================================================
                            Josh Talley and Daniel O'Donnell
                                   Dulaney High School
                         Mobile Application Development 2016-17
   ============================================================================
   Purpose: This activity is used to display and edit account information. When
   a user first logs it, you are forwarded directly to this activity.
*/
package com.fbla.dulaney.fblayardsale;

import android.databinding.DataBindingUtil;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Toast;

import com.fbla.dulaney.fblayardsale.controller.LocalController;
import com.fbla.dulaney.fblayardsale.databinding.ActivityAccountBinding;
import com.fbla.dulaney.fblayardsale.model.Account;
import com.microsoft.windowsazure.mobileservices.table.MobileServiceTable;

public class AccountEdit extends AppCompatActivity implements View.OnClickListener {

    ActivityAccountBinding mBinding;
    ArrayAdapter<CharSequence> mStateAdapter;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_account);

        if (!FblaLogon.getLoggedOn()) {
            Toast.makeText(this, "Unable to connect to Azure. Please try again.", Toast.
LENGTH_LONG).show();
            finish();
            return;
        }

        mBinding = DataBindingUtil.setContentView(this, R.layout.activity_account);
        mBinding.save.setOnClickListener(this);
        mBinding.cancel.setOnClickListener(this);

        // Load the states onto the spinner from the resource file
        mStateAdapter = ArrayAdapter.createFromResource(this, R.array.states_list, android.R
.layout.simple_spinner_item);
        mStateAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
;
        mBinding.state.setAdapter(mStateAdapter);
        setSupportActionBar(mBinding.myToolbar);

        Account account = FblaLogon.getAccount();
        int spinnerPosition = mStateAdapter.getPosition(account.getState());
        mBinding.name.setText(account.getName());
        mBinding.state.setSelection(spinnerPosition);
        mBinding.address.setText(account.getAddress());
        mBinding.chapter.setText(account.getChapter());
        mBinding.region.setText(account.getRegion());
        mBinding.zip.setText(account.getZipCode());
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()) {

            case R.id.save:
                if (FblaLogon.getLoggedOn()) {
                    Account account = FblaLogon.getAccount();
                    account.setName(mBinding.name.getText().toString());
                    account.setAddress(mBinding.address.getText().toString());
                    account.setChapter(mBinding.chapter.getText().toString());
                    account.setRegion(mBinding.region.getText().toString());
                    // If zip code changed, refresh the LocalController
                    if (!account.getZipCode().equals(mBinding.zip.getText().toString())) {
                        account.setZipCode(mBinding.zip.getText().toString());
                        LocalController.Refresh();
                    }
                    int statePosition = mBinding.state.getSelectedItemPosition();
                    account.setState(mStateAdapter.getItem(statePosition).toString());

                    // Save the item to the database over the internet.
                    AsyncTask<Void, Void, Void> task = new AsyncTask<Void, Void, Void>() {
                        @Override
                        protected Void doInBackground(Void... params) {
                            try {
                                MobileServiceTable<Account> mAccountTable = FblaLogon.
getClient().getTable(Account.class);
                                mAccountTable.update(FblaLogon.getAccount());
                                Log.d("AccountEdit:onClick", "AccountEdit Saved");
                                runOnUiThread(new Runnable() {
                                    @Override
                                    public void run() {
                                        finish();
                                    }
                                });
                            } catch (Exception e) {
                                Log.d("AddItem", e.toString());
                            }
                            return null;
                        }
                    };
                    task.executeOnExecutor(AsyncTask.SERIAL_EXECUTOR);
                }
                break;
            case R.id.cancel:
                this.finish();
                break;
            default:
                break;
        }
    }

    @Override
    public void onBackPressed()
    {
        this.finish();
    }
}

```

```java
/* FblaPicture.java
   ============================================================================
                             Josh Talley and Daniel O'Donnell
                                    Dulaney High School
                          Mobile Application Development 2016-17
   ============================================================================
   Purpose: This class contains a bunch of helper methods that make it easy to
   manage pictures.
*/
package com.fbla.dulaney.fblayardsale;

import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Point;
import android.graphics.drawable.BitmapDrawable;
import android.graphics.drawable.Drawable;
import android.util.Base64;
import android.util.Log;
import android.view.Display;
import android.view.WindowManager;
import android.widget.ImageView;
import android.widget.LinearLayout;

import java.io.ByteArrayOutputStream;

public class FblaPicture {
    private static LinearLayout mLayoutImage;

    public static void setLayoutImage(LinearLayout layout)
    {
        mLayoutImage = layout;
    }
    public static int getImageHeight()
    {
        if (mLayoutImage.getHeight() > 0)
            return mLayoutImage.getHeight() / 2;
        else return 0;
    }

    // Returns dimensions of phone in pixels
    public static Point GetSize(Context c)
    {
        WindowManager wm = (WindowManager) c.getSystemService(Context.WINDOW_SERVICE);
        Display display = wm.getDefaultDisplay();
        Point size = new Point();
        if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.HONEYCOMB_MR2
)
        {
            display.getSize(size);
        }
        else // Old Version
        {
            size.set(display.getWidth(), display.getHeight());
        }
        return size;
    }

    // Loads a bitmap picture onto the ImageView item on the layout.
    public static void LoadPictureOnView(ImageView view, Bitmap original) {
        int vh = getImageHeight();
        view.setMinimumHeight(vh);
        view.setMaxHeight(vh);
        view.setImageBitmap(original);
    }

    public static Bitmap GetPictureFromView(ImageView view) {
        Drawable d = view.getDrawable();
        if (d == null) return null;
        return ((BitmapDrawable)d).getBitmap();
    }

    // Resizes a picture selected from the gallery or taken by the camera so they are a
common size.
    public static Bitmap ResizePicture(Context c, Bitmap original) {
        int w = original.getWidth();
        int h = original.getHeight();
        Point screen = GetSize(c);
        // Force everything to be 500 pixels long
        int screenL = 500;
        int originL = (w > h) ? w : h;
        int originS = (w > h) ? h : w;

        int newS = (int)((float)screenL * ((float)originS / (float)originL));
        if (w > h)
        {
            Log.d("Picture:ResizePicture", "Screen " + screen.x + "x" + screen.y + " From "
 + w + "x" + h + " to " + screenL + "x" + newS);
            return Bitmap.createScaledBitmap(original, screenL, newS, true);
        }
        else
        {
            Log.d("Picture:ResizePicture", "Screen " + screen.x + "x" + screen.y + " From "
 + w + "x" + h + " to " + newS + "x" + screenL);
            return Bitmap.createScaledBitmap(original, newS, screenL, true);
        }
    }

    public static String EncodeToBase64(Bitmap image) {
        if (image == null) return "";
        // http://stackoverflow.com/questions/9768611/encode-and-decode-bitmap-object-in-
base64-string-in-android
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        image.compress(Bitmap.CompressFormat.PNG, 100, baos);
        byte[] b = baos.toByteArray();
        return Base64.encodeToString(b, Base64.DEFAULT);
    }

    public static Bitmap DecodeFromBase64(String image) {
        if (image == null || image.equals("")) return null;
        byte[] decodedImage = Base64.decode(image, Base64.DEFAULT);
        return BitmapFactory.decodeByteArray(decodedImage, 0, decodedImage.length);
    }
}

```

```java
/* HelpAdapter.java
   =============================================================================
                     Josh Talley and Daniel O'Donnell
                            Dulaney High School
                  Mobile Application Development 2016-17
   =============================================================================
   Purpose: This is the recycler view adapter for the Help activity.
*/
package com.fbla.dulaney.fblayardsale;

import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class HelpAdapter  extends RecyclerView.Adapter<HelpAdapter.ViewHolder> {

    @Override
    public int getItemViewType(int position) {
        return position;
    }

    @Override
    public HelpAdapter.ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View v;
        switch (viewType)
        {
            case 0:
                v = LayoutInflater.from(parent.getContext())
                        .inflate(R.layout.help01_register, parent, false);
                break;
            case 1:
                v = LayoutInflater.from(parent.getContext())
                        .inflate(R.layout.help02_login, parent, false);
                break;
            case 2:
                v = LayoutInflater.from(parent.getContext())
                        .inflate(R.layout.help03_switch, parent, false);
                break;
            case 3:
                v = LayoutInflater.from(parent.getContext())
                        .inflate(R.layout.help04_editaccount, parent, false);
                break;
            case 4:
                v = LayoutInflater.from(parent.getContext())
                        .inflate(R.layout.help05_addsales, parent, false);
                break;
            case 5:
                v = LayoutInflater.from(parent.getContext())
                        .inflate(R.layout.help06_viewsales, parent, false);
                break;
            case 6:
                v = LayoutInflater.from(parent.getContext())
                        .inflate(R.layout.help07_deletesales, parent, false);
                break;
            case 7:
                v = LayoutInflater.from(parent.getContext())
                        .inflate(R.layout.help08_comment, parent, false);
                break;
            case 8:
                v = LayoutInflater.from(parent.getContext())
                        .inflate(R.layout.help09_deletecomment, parent, false);
                break;
            case 9:
                v = LayoutInflater.from(parent.getContext())
                        .inflate(R.layout.help10_logout, parent, false);
                break;
            default:
                v = null;
                break;
        }
        if (v == null) return null;
        return new HelpAdapter.ViewHolder(v);
    }

    @Override
    public void onBindViewHolder(HelpAdapter.ViewHolder holder, int position) {

    }

    @Override
    public int getItemCount() {
        return 10;
    }

    class ViewHolder extends RecyclerView.ViewHolder {

        public ViewHolder(View itemView) {
            super(itemView);
        }
    }
}

```

```java
/* HomeFragment.java
   =============================================================================
                         Josh Talley and Daniel O'Donnell
                                Dulaney High School
                      Mobile Application Development 2016-17
   =============================================================================
   Purpose: This is the first fragment loaded on YardSaleMain. It shows the
   application icon and is used like a menu. Buttons take you other activities.
   You can also swipe left to get to the Local Sales fragment.
*/
package com.fbla.dulaney.fblayardsale;

import android.content.Context;
import android.content.Intent;
import android.databinding.DataBindingUtil;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentActivity;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import com.fbla.dulaney.fblayardsale.databinding.FragmentHomeBinding;

public class HomeFragment extends Fragment implements View.OnClickListener {

    private OnFragmentInteractionListener mListener;
    private FragmentActivity mParent;
    FragmentHomeBinding mBinding;

    @Override
    public void onClick(View v) {
        switch (v.getId()) {

            case R.id.account:
                if (FblaLogon.getLoggedOn()) {
                    getActivity().startActivity(new Intent(getActivity(), AccountEdit.class)
);
                }
                break;
            case R.id.add:
                if (FblaLogon.getLoggedOn()) {
                    getActivity().startActivity(new Intent(getActivity(), AddSales.class));
                }
                break;
            case R.id.my:
                if (FblaLogon.getLoggedOn()) {
                    getActivity().startActivity(new Intent(getActivity(), MySales.class));
                }
                break;
            case R.id.help:
                getActivity().startActivity(new Intent(getActivity(), Help.class));
                break;
            case R.id.logout:
                YardSaleMain parent = (YardSaleMain)getActivity();
                parent.Logoff();
                break;
            default:
                break;
        }
    }

    public interface OnFragmentInteractionListener {
        public void onHomeInteraction(View v);
    }

    // Implementation of Fragment
    public static HomeFragment newInstance(String param1, String param2) {
        HomeFragment fragment = new HomeFragment();
        Bundle args = new Bundle();
        fragment.setArguments(args);
        return fragment;
    }

    public HomeFragment() {
        // Required empty public constructor
    }

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        try {
            mListener = (OnFragmentInteractionListener) context;
        } catch (ClassCastException e) {
            throw new ClassCastException(context.toString()
                    + " must implement OnFragmentInteractionListener");
        }
    }

    @Override
    public void onDetach() {
        super.onDetach();
        mListener = null;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {

        mBinding = DataBindingUtil.inflate(
                inflater, R.layout.fragment_home, container, false);
        mBinding.account.setOnClickListener(this);
        mBinding.add.setOnClickListener(this);
        mBinding.my.setOnClickListener(this);
        mBinding.help.setOnClickListener(this);
        mBinding.logout.setOnClickListener(this);
        View view = mBinding.getRoot();

        // Inflate the layout for this fragment
        //View v = inflater.inflate(R.layout.fragment_home, container, false);
        //mParent = getActivity();
        return view;
    }

    // Initializes layout items
    @Override
    public void onActivityCreated(Bundle bundle) {
        super.onActivityCreated(bundle);
        mParent = getActivity();
    }

```

```
}

```

```java
/* LocalAdapter.java
   =============================================================================
                         Josh Talley and Daniel O'Donnell
                                Dulaney High School
                      Mobile Application Development 2016-17
   =============================================================================
   Purpose: This is the recycler view adapter for the Local fragment.
*/
package com.fbla.dulaney.fblayardsale;

import android.databinding.DataBindingUtil;
import android.graphics.Bitmap;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import com.fbla.dulaney.fblayardsale.controller.CommentListController;
import com.fbla.dulaney.fblayardsale.controller.LocalController;
import com.fbla.dulaney.fblayardsale.databinding.ListItemsBinding;
import com.fbla.dulaney.fblayardsale.model.Account;
import com.fbla.dulaney.fblayardsale.model.SaleItem;

public class LocalAdapter extends RecyclerView.Adapter<LocalAdapter.ViewHolder> implements
View.OnClickListener {
    private View.OnClickListener mParentListener;
    private ListItemsBinding mBinding;

    public LocalAdapter (View.OnClickListener onClickListener) {
        mParentListener = onClickListener;
    }

    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        ListItemsBinding mBinding = DataBindingUtil.inflate(
                LayoutInflater.from(parent.getContext()), R.layout.list_items, parent, false
);
        mBinding.sold.setOnClickListener(this);
        mBinding.sold.setVisibility(View.GONE); //this will be variable based on account info
        mBinding.comments.setOnClickListener(this);
        View view = mBinding.getRoot();

        return new ViewHolder(view, mBinding);
    }

    @Override
    public void onBindViewHolder(ViewHolder holder, int position) {
        if (!FblaLogon.getLoggedOn()) return;
        SaleItem item = LocalController.getItem(position);
        if (item != null) {
            mBinding = holder.getBinding();
            Log.d("LocalAdapter", "onBindViewHolder");
            mBinding.comments.setTag(position);
            mBinding.name.setText(item.getName());
            mBinding.price.setText(String.format("$%.2f", item.getPrice()));
            mBinding.description.setText(item.getDescription());
            Account account = item.getAccount();
            if (account != null) {
                mBinding.address.setText(account.getAddress());
                mBinding.chapter.setText(account.getChapter());
                mBinding.zipcode.setText(account.getZipCode());
            }
            Bitmap image = item.getPicture();
            if (image != null) {
                FblaPicture.setLayoutImage(mBinding.layoutPicture);
                FblaPicture.LoadPictureOnView(mBinding.picture, image);
            }
        }
    }

    @Override
    public int getItemCount() {
        return LocalController.getItemCount();
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.comments:
                if (FblaLogon.getLoggedOn()) {
                    int position = (int)v.getTag();
                    CommentListController.setItem(LocalController.getItem(position));
                    CommentListController.Refresh();
                    mParentListener.onClick(v);
                }
                break;
        }
    }

    public class ViewHolder extends RecyclerView.ViewHolder {
        private ListItemsBinding mBinding;

        public ViewHolder(View itemView, ListItemsBinding binding) {
            super(itemView);
            mBinding = binding;
        }

        public ListItemsBinding getBinding() {
            return mBinding;
        }
    }
}

```

```java
/* YardSaleMain.java
   =============================================================================
                         Josh Talley and Daniel O'Donnell
                                Dulaney High School
                      Mobile Application Development 2016-17
   =============================================================================
   Purpose: This is the main startup activity. It uses the FblaPagerAdapter to manage 2 different
   activity fragments. This activity has the title bar and navigation buttons.
   The ViewPager fills the center, which holds each page fragment. It automatically handles
   swipes and smooth transitions between each page. Most navigation is handled by this activity.
   This activity will also execute the initial login using a Google+ account.  The login
   uses Azure Mobile Apps to get the user's Google ID and Token, which is cached and used
   by all other database calls to Azure Mobile Apps.
*/

package com.fbla.dulaney.fblayardsale;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.databinding.DataBindingUtil;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Build;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

import android.view.View;
import android.widget.Toast;

import com.fbla.dulaney.fblayardsale.controller.LocalController;
import com.fbla.dulaney.fblayardsale.controller.MySalesController;
import com.fbla.dulaney.fblayardsale.databinding.ActivityYardsaleBinding;
import com.google.android.gms.appindexing.Action;
import com.google.android.gms.appindexing.AppIndex;
import com.google.android.gms.appindexing.Thing;
import com.google.android.gms.common.api.GoogleApiClient;

public class YardSaleMain extends AppCompatActivity implements View.OnClickListener,
        HomeFragment.OnFragmentInteractionListener,
        LocalFragment.OnFragmentInteractionListener,
        FblaLogon.LogonResultListener{

    // Class Variables
    FblaPagerAdapter mPagerAdapter;
    int mCurrentPage;
    ActivityYardsaleBinding mBinding;
    FblaLogon mLogon;
    public void Logoff() {
        mLogon.Logoff();
        //finish();
        mLogon = new FblaLogon(this);
        mLogon.setLogonListener(this);
        mLogon.executeOnExecutor(AsyncTask.SERIAL_EXECUTOR);
    }
    //private MobileServiceTable<AccountEdit> mAccountTable;

    /**
     * ATTENTION: This was auto-generated to implement the App Indexing API.
     * See https://g.co/AppIndexing/AndroidStudio for more information.
      */
     private GoogleApiClient client;
     //private static final String TEMP_FILENAME = "Camera";

     @Override
     protected void onCreate(Bundle savedInstanceState) {
         super.onCreate(savedInstanceState);
         setContentView(R.layout.activity_yardsale);

         mLogon = new FblaLogon(this);

         mBinding = DataBindingUtil.setContentView(this, R.layout.activity_yardsale);
         mPagerAdapter = new FblaPagerAdapter(getSupportFragmentManager(), this);
         mBinding.pager.setAdapter(mPagerAdapter);
         mBinding.home.setOnClickListener(this);
         mBinding.local.setOnClickListener(this);
         setSupportActionBar(mBinding.myToolbar);

         mLogon.setLogonListener(this);
         if (!FblaLogon.getLoggedOn())
             mLogon.executeOnExecutor(AsyncTask.SERIAL_EXECUTOR);

         // ATTENTION: This was auto-generated to implement the App Indexing API.
         // See https://g.co/AppIndexing/AndroidStudio for more information.
         client = new GoogleApiClient.Builder(this).addApi(AppIndex.API).build();
     }

     //@Override
     public void onClick(View v) {
         // Perform page changes so they transition just like a swipe.
         int pg;
         switch (v.getId()) {
             case R.id.home:
                 pg = 0;
                 break;
             case R.id.local:
                 pg = 1;
                 break;
             default:
                 pg = 2;
                 break;
         }
         mBinding.pager.setCurrentItem(pg, true);
     }

     /**
      * ATTENTION: This was auto-generated to implement the App Indexing API.
      * See https://g.co/AppIndexing/AndroidStudio for more information.
      */
     public Action getIndexApiAction() {
         Thing object = new Thing.Builder()
                 .setName("YardSaleMain Page") // TODO: Define a title for the content shown
 .
                 // TODO: Make sure this auto-generated URL is correct.
                 .setUrl(Uri.parse("http://[ENTER-YOUR-URL-HERE]"))
                 .build();
         return new Action.Builder(Action.TYPE_VIEW)
                 .setObject(object)
                 .setActionStatus(Action.STATUS_TYPE_COMPLETED)
                 .build();
     }

     @Override
```

```java
    public void onStart() {
        super.onStart();

        // ATTENTION: This was auto-generated to implement the App Indexing API.
        // See https://g.co/AppIndexing/AndroidStudio for more information.
        client.connect();
        AppIndex.AppIndexApi.start(client, getIndexApiAction());
    }

    @Override
    public void onStop() {
        super.onStop();

        // ATTENTION: This was auto-generated to implement the App Indexing API.
        // See https://g.co/AppIndexing/AndroidStudio for more information.
        AppIndex.AppIndexApi.end(client, getIndexApiAction());
        client.disconnect();
    }

    public void onHomeInteraction(View v)
    {

    }

    public void onLocalInteraction(View v)
    {

    }

    @Override
    public void onLogonComplete(Exception e) {
        if (e != null) {
            Toast.makeText(this, "Unable to connect to Azure. Please try again.", Toast.
LENGTH_LONG).show();
            finish();
        } else {
            MySalesController.Refresh();
            LocalController.Refresh();
            String name = FblaLogon.getAccount().getName();
            if (name == null || name.equals("")) {
                startActivity(new Intent(this, AccountEdit.class));
            }
        }
    }
}
```

```java
/* LocalFragment.java
   =============================================================================
                         Josh Talley and Daniel O'Donnell
                                Dulaney High School
                      Mobile Application Development 2016-17
   =============================================================================
   Purpose: This is the second fragment loaded on YardSaleMain. It will display
   all sale items that are in your same zip code, excluding any of your own
   sale items. You can also swipe right to get to the Home fragment.
*/
package com.fbla.dulaney.fblayardsale;

import android.content.Context;
import android.content.Intent;
import android.databinding.DataBindingUtil;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentActivity;
import android.os.Bundle;
import android.support.v7.widget.LinearLayoutManager;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import com.fbla.dulaney.fblayardsale.controller.LocalController;
import com.fbla.dulaney.fblayardsale.databinding.FragmentLocalBinding;

public class LocalFragment extends Fragment implements View.OnClickListener {

    private LocalFragment.OnFragmentInteractionListener mListener;
    private FragmentActivity mParent;
    FragmentLocalBinding mBinding;

    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.comments:
                getActivity().startActivity(new Intent(getActivity(), Comments.class));
                break;
            default:
                break;
        }
    }

    public interface OnFragmentInteractionListener {
        public void onLocalInteraction(View v);
    }

    // Implementation of Fragment
    public static LocalFragment newInstance(String param1, String param2) {
        LocalFragment fragment = new LocalFragment();
        Bundle args = new Bundle();
        fragment.setArguments(args);
        return fragment;
    }

    public LocalFragment() {
        // Required empty public constructor
    }

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        try {
            mListener = (LocalFragment.OnFragmentInteractionListener) context;
        } catch (ClassCastException e) {
            throw new ClassCastException(context.toString()
                    + " must implement OnFragmentInteractionListener");
        }
    }

    @Override
    public void onDetach() {
        super.onDetach();
        mListener = null;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        mBinding = DataBindingUtil.inflate(
                inflater, R.layout.fragment_local, container, false);
        //mBinding.comments.setOnClickListener(this);
        View view = mBinding.getRoot();
        return view;
    }

    @Override
    public void onActivityCreated(Bundle bundle) {
        super.onActivityCreated(bundle);
        // Setup the RecyclerView here because the data changes.
        mBinding.list.setLayoutManager(new LinearLayoutManager(mParent));
        LocalAdapter adapter = new LocalAdapter(this);
        LocalController.AttachAdapter(adapter);
        LocalController.Refresh();
        mBinding.list.setAdapter(adapter);
    }

}

```

```java
/* MySalesAdapter.java
   =============================================================================
                         Josh Talley and Daniel O'Donnell
                                Dulaney High School
                      Mobile Application Development 2016-17
   =============================================================================
   Purpose: This is the recycler view adapter for the MySales fragment.
*/
package com.fbla.dulaney.fblayardsale;

import android.content.DialogInterface;
import android.databinding.DataBindingUtil;
import android.graphics.Bitmap;
import android.os.AsyncTask;
import android.support.v7.app.AlertDialog;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import com.fbla.dulaney.fblayardsale.controller.CommentListController;
import com.fbla.dulaney.fblayardsale.controller.MySalesController;
import com.fbla.dulaney.fblayardsale.databinding.ListItemsBinding;
import com.fbla.dulaney.fblayardsale.model.SaleItem;
import com.microsoft.windowsazure.mobileservices.table.MobileServiceTable;

public class MySalesAdapter extends RecyclerView.Adapter<MySalesAdapter.ViewHolder>
implements View.OnClickListener {
    private View.OnClickListener mParentListener;
    ListItemsBinding mBinding;
    MySales mContext;

    public MySalesAdapter (MySales context, View.OnClickListener onClickListener) {
        mContext = context;
        mParentListener = onClickListener;
    }

    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        ListItemsBinding mBinding = DataBindingUtil.inflate(
                LayoutInflater.from(parent.getContext()), R.layout.list_items, parent, false
);
        mBinding.sold.setOnClickListener(this);
        mBinding.comments.setOnClickListener(this);
        mBinding.layoutAddress.setVisibility(View.GONE);
        mBinding.layoutChapter.setVisibility(View.GONE);
        mBinding.layoutZipcode.setVisibility(View.GONE);
        View view = mBinding.getRoot();

        return new ViewHolder(view, mBinding);
    }

    @Override
    public void onBindViewHolder(ViewHolder holder, int position) {
        if (!FblaLogon.getLoggedOn()) return;
        SaleItem item = MySalesController.getItem(position);
        if (item != null) {
            mBinding = holder.getBinding();
            Log.d("MySalesAdapter", "onBindViewHolder");
            mBinding.comments.setTag(position);
            mBinding.name.setText(item.getName());
            mBinding.price.setText(String.format("$%.2f", item.getPrice()));
            mBinding.description.setText(item.getDescription());
            mBinding.sold.setTag(position);
            Bitmap image = item.getPicture();
            if (image != null) {
                FblaPicture.setLayoutImage(mBinding.layoutPicture);
                FblaPicture.LoadPictureOnView(mBinding.picture, image);
            }
        }
    }

    @Override
    public int getItemCount() {
        return MySalesController.getItemCount();
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.comments:
                if (FblaLogon.getLoggedOn()) {
                    int position = (int)v.getTag();
                    SaleItem item = MySalesController.getItem(position);
                    CommentListController.setItem(item);
                    CommentListController.Refresh();
                    Log.d("MySalesAdapter", "Refreshed for " + position);
                    mParentListener.onClick(v);
                }
                break;
            case R.id.sold:
                final int position = (int)v.getTag();
                AlertDialog.Builder builder = new AlertDialog.Builder(mContext);
                builder.setTitle("Are You Sure?");
                final TextView info = new TextView(mContext);
                info.setText("By Pressing Confirm, The Item Will Be Deleted.");
                info.setPadding(30, 0, 0, 0);
                builder.setView(info);

                builder.setPositiveButton("Confirm", new DialogInterface.OnClickListener()
 {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        Log.d("MySalesAdapter", "delete");
                        deleteItem(position);
                        dialog.dismiss();
                    }
                });

                builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        dialog.cancel();
                    }
                });

                builder.show();
                break;
            default:
                break;
        }
    }

    public class ViewHolder extends RecyclerView.ViewHolder {
```

```java
        private ListItemsBinding mBinding;

        public ViewHolder(View itemView, ListItemsBinding binding) {
            super(itemView);
            mBinding = binding;
        }

        public ListItemsBinding getBinding() {
            return mBinding;
        }
    }

    private void deleteItem(int position) {
        if (!FblaLogon.getLoggedOn()) return;

        final int pos = position;
        final SaleItem item = MySalesController.getItem(position);
        final MobileServiceTable<SaleItem> mSaleItemTable = FblaLogon.getClient().getTable(
SaleItem.class);
        // Delete the comment from the database.
        AsyncTask<Void, Void, Void> task = new AsyncTask<Void, Void, Void>() {
            @Override
            protected Void doInBackground(Void... params) {
                try {
                    mSaleItemTable.delete(item);
                    Log.d("MySales:delete", "Deleted item " + item.getName());
                    mContext.runOnUiThread(new Runnable() {
                        @Override
                        public void run() {
                            MySalesController.removeItem(pos);
                        }
                    });
                } catch (Exception e) {
                    Log.d("MySales:delete", e.toString());
                }
                return null;
            }
        };
        task.executeOnExecutor(AsyncTask.SERIAL_EXECUTOR);
    }
}
```

```java
/* CommentsAdapter.java
   =============================================================================
                             Josh Talley and Daniel O'Donnell
                                    Dulaney High School
                          Mobile Application Development 2016-17
   =============================================================================
   Purpose: This adapter is used by the Comments activity to manage the list of
   comments. It makes use of the CommentListController.
*/
package com.fbla.dulaney.fblayardsale;

import android.content.DialogInterface;
import android.databinding.DataBindingUtil;
import android.os.AsyncTask;
import android.support.v7.app.AlertDialog;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import com.fbla.dulaney.fblayardsale.controller.CommentListController;
import com.fbla.dulaney.fblayardsale.databinding.ListCommentsBinding;
import com.fbla.dulaney.fblayardsale.model.ItemComment;
import com.microsoft.windowsazure.mobileservices.table.MobileServiceTable;

public class CommentsAdapter extends RecyclerView.Adapter<CommentsAdapter.ViewHolder>
implements View.OnClickListener {
    private View.OnClickListener mParentListener;
    ListCommentsBinding mBinding;
    Comments mContext;

    public CommentsAdapter (Comments context, View.OnClickListener onClickListener) {
        mContext = context;
        mParentListener = onClickListener;
    }

    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        ListCommentsBinding mBinding = DataBindingUtil.inflate(
                LayoutInflater.from(parent.getContext()), R.layout.list_comments, parent,
false);
        View view = mBinding.getRoot();
        mBinding.delete.setOnClickListener(this);

        Log.d("CommentsAdapter", "onCreateViewHolder");
        return new ViewHolder(view, mBinding);
    }

    @Override
    public void onBindViewHolder(ViewHolder holder, int position) {
        if (!FblaLogon.getLoggedOn()) return;
        ItemComment comment = CommentListController.getComment(position);
        if (comment != null) {
            mBinding = holder.getBinding();
            Log.d("CommentsAdapter", "onBindViewHolder");
            mBinding.comments.setText(comment.getComment());
            if (comment.getAccount() == null) mBinding.username.setText("{Unknown}");
            else mBinding.username.setText(comment.getAccount().getName());
            mBinding.delete.setTag(position);
        }
    }

    @Override
    public int getItemCount() {
        return CommentListController.getCommentCount();
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.delete:
                final int position = (int)v.getTag();
                AlertDialog.Builder builder = new AlertDialog.Builder(mContext);
                builder.setTitle("Are You Sure?");
                final TextView info = new TextView(mContext);
                info.setText("By Pressing Confirm, The Comment Will Be Deleted.");
                info.setPadding(30, 0, 0, 0);
                builder.setView(info);

                builder.setPositiveButton("Confirm", new DialogInterface.OnClickListener()
 {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        deleteComment(position);
                        dialog.dismiss();
                    }
                });

                builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        dialog.cancel();
                    }
                });

                builder.show();
                break;
            default:
                break;
        }
    }

    private void deleteComment(int position) {
        if (!FblaLogon.getLoggedOn()) return;

        final int pos = position;
        final ItemComment comment = CommentListController.getComment(position);
        final MobileServiceTable<ItemComment> mCommentTable = FblaLogon.getClient().
getTable(ItemComment.class);
        // Delete the comment from the database.
        AsyncTask<Void, Void, Void> task = new AsyncTask<Void, Void, Void>() {
            @Override
            protected Void doInBackground(Void... params) {
                try {
                    mCommentTable.delete(comment);
                    Log.d("Comments:delete", "Deleted comment " + comment.getComment());
                    mContext.runOnUiThread(new Runnable() {
                        @Override
                        public void run() {
                            CommentListController.removeComment(pos);
                        }
                    });
                } catch (Exception e) {
                    Log.d("Comments:delete", e.toString());
                }
                return null;
            }
        };
        task.executeOnExecutor(AsyncTask.SERIAL_EXECUTOR);
    }

    public class ViewHolder extends RecyclerView.ViewHolder {
        private ListCommentsBinding mBinding;

        public ViewHolder(View itemView, ListCommentsBinding binding) {
            super(itemView);
            mBinding = binding;
        }

        public ListCommentsBinding getBinding() {
            return mBinding;
        }
    }
}

```

```java
/* FblaPagerAdapter.java
   =============================================================================
                        Josh Talley and Daniel O'Donnell
                               Dulaney High School
                     Mobile Application Development 2016-17
   =============================================================================
   Purpose: This simply loads the appropriate fragment onto the YardSaleMain activity.
*/
package com.fbla.dulaney.fblayardsale;

import android.content.Context;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentStatePagerAdapter;

public class FblaPagerAdapter extends FragmentStatePagerAdapter {
    protected Context mContext;

    public FblaPagerAdapter(FragmentManager fm, Context context)
    {
        super(fm);
        mContext = context;
    }

    @Override
    public Fragment getItem(int position) {
        Fragment fragment;
        switch (position)
        {
            case 0:
                fragment = new HomeFragment();
                break;
            default:
                fragment = new LocalFragment();
                break;
        }
        Bundle args = new Bundle();
        args.putInt("page_position", position);

        fragment.setArguments(args);

        return fragment;
    }

    @Override
    public int getCount() {
        return 2;
    }
}

```

```java
/* Account.java
   ============================================================================
                         Josh Talley and Daniel O'Donnell
                                Dulaney High School
                      Mobile Application Development 2016-17
   ============================================================================
   Purpose: Model of the Azure database table for user account information.
*/
package com.fbla.dulaney.fblayardsale.model;

public class Account {
    /*
    User Id
     */
    @com.google.gson.annotations.SerializedName("id")
    private String mId;

    /*
    User Name
     */
    @com.google.gson.annotations.SerializedName("name")
    private String mName;

    /*
    State
     */
    @com.google.gson.annotations.SerializedName("state")
    private String mState;

    /*
    Region
     */
    @com.google.gson.annotations.SerializedName("region")
    private String mRegion;

    /*
    FBLA Chapter
     */
    @com.google.gson.annotations.SerializedName("chapter")
    private String mChapter;

    /*
    Address
     */
    @com.google.gson.annotations.SerializedName("address")
    private String mAddress;

    /*
    Zip Code
     */
    @com.google.gson.annotations.SerializedName("zipcode")
    private String mZipCode;

    public Account() {
        mId = "";
        mName = "";
        mAddress = "";
        mChapter = "";
        mRegion = "";
        mState = "";
        mZipCode = "";
    }

    @Override
    public String toString() {
        return getId();
    }

    // Getters and Setters
    public String getId() { return mId; }
    public final void setId(String id) { mId = id; }
    public String getName() { return mName; }
    public final void setName(String name) { mName = name; }
    public String getState() { return mState; }
    public final void setState(String state) { mState = state; }
    public String getRegion() { return mRegion; }
    public final void setRegion(String region) { mRegion = region; }
    public String getChapter() { return mChapter; }
    public final void setChapter(String chapter) { mChapter = chapter; }
    public String getAddress() { return mAddress; }
    public final void setAddress(String address) { mAddress = address; }
    public String getZipCode() { return mZipCode; }
    public final void setZipCode(String zipCode) { mZipCode = zipCode; }

    @Override
    public boolean equals(Object o) {
        return o instanceof Account && ((Account)o).mId == mId;
    }
}

```

```java
/* SaleItem.java
   =============================================================================
                         Josh Talley and Daniel O'Donnell
                                Dulaney High School
                      Mobile Application Development 2016-17
   =============================================================================
   Purpose: Model of the Azure database table for sale item information.
*/
package com.fbla.dulaney.fblayardsale.model;

import android.graphics.Bitmap;

import com.fbla.dulaney.fblayardsale.FblaPicture;

public class SaleItem {
    /*
    Item Id
     */
    @com.google.gson.annotations.SerializedName("id")
    private String mId;

    /*
    User Id
     */
    @com.google.gson.annotations.SerializedName("userid")
    private String mUserId;


    /*
    User Id
     */
    @com.google.gson.annotations.SerializedName("name")
    private String mName;

    /*
    Item Description
     */
    @com.google.gson.annotations.SerializedName("description")
    private String mDescription;

    /*
    Item Price
     */
    @com.google.gson.annotations.SerializedName("price")
    private float mPrice;

    /*
    Picture (Base64 Encoded String)
     */
    @com.google.gson.annotations.SerializedName("picture")
    private String mPictureBase64;

    @com.google.gson.annotations.Expose(serialize = false)
    private Account mAccount;

    public SaleItem() {
        mAccount = null;
        mName = "";
        mId = "";
        mUserId = "";
        mDescription = "";
        mPictureBase64 = "";
        mPrice = 0;
    }

    @Override
    public String toString() {
        return getId();
    }

    // Getters and Setters
    public String getId() { return mId; }
    public final void setId(String id) { mId = id; }
    public String getUserId() { return mUserId; }
    public final void setUserId(String userId) { mUserId = userId; }
    public String getName() { return mName; }
    public final void setName(String name) { mName = name; }
    public String getDescription() { return mDescription; }
    public final void setDescription(String description) { mDescription = description; }
    public float getPrice() { return mPrice; }
    public final void setPrice(float price) { mPrice = price; }
    public Bitmap getPicture() {
        return FblaPicture.DecodeFromBase64(mPictureBase64);
    }
    public final void setPicture(Bitmap image) {
        mPictureBase64 = FblaPicture.EncodeToBase64(image);
    }
    public Account getAccount() { return mAccount; }
    public final void setAccount(Account account) { mAccount = account; }

    @Override
    public boolean equals(Object o) {
        return o instanceof SaleItem && ((SaleItem)o).mId == mId;
    }
}

```

```java
/* ItemComment.java
   =============================================================================
                        Josh Talley and Daniel O'Donnell
                                Dulaney High School
                     Mobile Application Development 2016-17
   =============================================================================
   Purpose: Model of the Azure database table for item comment information.
*/
package com.fbla.dulaney.fblayardsale.model;

public class ItemComment {
    /*
    Item Id
     */
    @com.google.gson.annotations.SerializedName("id")
    private String mId;

    /*
    User Id
     */
    @com.google.gson.annotations.SerializedName("userid")
    private String mUserId;

    /*
    Item Id
     */
    @com.google.gson.annotations.SerializedName("itemid")
    private String mItemId;

    /*
    Comment
     */
    @com.google.gson.annotations.SerializedName("comment")
    private String mComment;

    @com.google.gson.annotations.Expose(serialize = false)
    private Account mAccount;

    public ItemComment() {
        mAccount = null;
        mId = "";
        mUserId = "";
        mItemId = "";
        mComment = "";
    }

    @Override
    public String toString() {
        return getId();
    }

    // Getters and Setters
    public String getId() { return mId; }
    public final void setId(String id) { mId = id; }
    public String getUserId() { return mUserId; }
    public final void setUserId(String userId) { mUserId = userId; }
    public String getItemId() { return mItemId; }
    public final void setItemId(String itemId) { mItemId = itemId; }
    public String getComment() { return mComment; }
    public final void setComment(String comment) { mComment = comment; }
    public Account getAccount() { return mAccount; }
    public final void setAccount(Account account) { mAccount = account; }

    @Override
    public boolean equals(Object o) {
        return o instanceof ItemComment && ((ItemComment)o).mId == mId;
    }
}

```

```java
/* LocalController.java
   =============================================================================
                           Josh Talley and Daniel O'Donnell
                                    Dulaney High School
                       Mobile Application Development 2016-17
   =============================================================================
   Purpose: Used by LocalFragment to control access to the list of Sale Items in
   the user's local area (by zip code). Attaching a recycler view to the class
   so that when the list of items is refreshed or changed, the recycler view is
   notified of that change.
*/
package com.fbla.dulaney.fblayardsale.controller;

import android.os.AsyncTask;
import android.support.v7.widget.RecyclerView;
import android.util.Log;

import com.fbla.dulaney.fblayardsale.FblaLogon;
import com.fbla.dulaney.fblayardsale.model.Account;
import com.fbla.dulaney.fblayardsale.model.SaleItem;
import com.microsoft.windowsazure.mobileservices.MobileServiceList;
import com.microsoft.windowsazure.mobileservices.table.MobileServiceTable;

import java.util.ArrayList;

public class LocalController {
    private static ArrayList<SaleItem> mSaleItems = new ArrayList<>();
    private static ArrayList<RecyclerView.Adapter> mAdapters = new ArrayList<>();

    public static void AttachAdapter(RecyclerView.Adapter adapter) {
        mAdapters.add(adapter);
    }

    public static int getItemCount() {
        return mSaleItems.size();
    }

    public static SaleItem getItem(int position) {
        if (mSaleItems.size() > position) return mSaleItems.get(position);
        else return null;
    }

    public static void addItem(SaleItem item) {
        mSaleItems.add(item);
        for (RecyclerView.Adapter adapter : mAdapters) {
            adapter.notifyDataSetChanged();
        }
    }

    public static void removeItem(int position) {
        mSaleItems.remove(position);
        for (RecyclerView.Adapter adapter : mAdapters) {
            adapter.notifyDataSetChanged();
        }
    }

    private static MobileServiceTable<Account> mAccountTable;
    private static MobileServiceTable<SaleItem> mSaleItemTable;
    public static void Refresh() {
        if (!FblaLogon.getLoggedOn()) return;
        mSaleItems.clear();

        mAccountTable = FblaLogon.getClient().getTable(Account.class);
        mSaleItemTable = FblaLogon.getClient().getTable(SaleItem.class);
        new AsyncTask<Void, Void, Void>() {
            @Override
            protected Void doInBackground(Void... params) {
                try {
                    Account myAccount = FblaLogon.getAccount();
                    // First get all of the accounts in the same zip code.
                    final MobileServiceList<Account> accounts =
                            mAccountTable.where().field("zipcode").eq(myAccount.getZipCode(
)).execute().get();
                    for (Account account : accounts) {
                        // Now get all the items for those accounts (excluding your own)
                        if (!account.getId().equals(myAccount.getId())) {
                            final MobileServiceList<SaleItem> result =
                                    mSaleItemTable.where().field("userid").eq(account.getId
()).execute().get();
                            for (SaleItem item : result) {
                                item.setAccount(account);
                                mSaleItems.add(item);
                            }
                        }
                    }
                } catch (Exception exception) {
                    Log.e("MySalesController", exception.toString());
                }
                return null;
            }
            @Override
            protected void onPostExecute(Void v) {
                for (RecyclerView.Adapter adapter : mAdapters) {
                    adapter.notifyDataSetChanged();
                }
            }
        }.execute();
    }
}
```

```
/* MySalesController.java
   ==============================================================================
                            Josh Talley and Daniel O'Donnell
                                   Dulaney High School
                         Mobile Application Development 2016-17
   ==============================================================================
   Purpose: Used by MySalesFragment to control access to the list of Sale Items owned
   by the user. Attaching a recycler view to the class so that when the list of
   items is refreshed or changed, the recycler view is notified of that change.
*/
package com.fbla.dulaney.fblayardsale.controller;

import android.os.AsyncTask;
import android.support.v7.widget.RecyclerView;
import android.util.Log;

import com.fbla.dulaney.fblayardsale.FblaLogon;
import com.fbla.dulaney.fblayardsale.model.SaleItem;
import com.microsoft.windowsazure.mobileservices.MobileServiceList;
import com.microsoft.windowsazure.mobileservices.table.MobileServiceTable;

import java.util.ArrayList;

public class MySalesController {
    private static ArrayList<SaleItem> mSaleItems = new ArrayList<>();
    private static ArrayList<RecyclerView.Adapter> mAdapters = new ArrayList<>();

    public static void AttachAdapter(RecyclerView.Adapter adapter) {
        mAdapters.add(adapter);
    }

    public static int getItemCount() {
        return mSaleItems.size();
    }

    public static SaleItem getItem(int position) {
        if (mSaleItems.size() > position) return mSaleItems.get(position);
        else return null;
    }

    public static void addItem(SaleItem item) {
        mSaleItems.add(item);
        for (RecyclerView.Adapter adapter : mAdapters) {
            adapter.notifyDataSetChanged();
        }
    }

    public static void removeItem(int position) {
        mSaleItems.remove(position);
        for (RecyclerView.Adapter adapter : mAdapters) {
            adapter.notifyDataSetChanged();
        }
    }

    private static MobileServiceTable<SaleItem> mSaleItemTable;
    public static void Refresh() {
        if (!FblaLogon.getLoggedOn()) return;
        mSaleItems.clear();

        mSaleItemTable = FblaLogon.getClient().getTable(SaleItem.class);
        new AsyncTask<Void, Void, Void>() {
            @Override
            protected Void doInBackground(Void... params) {
                try {
                    final MobileServiceList<SaleItem> result =
                            mSaleItemTable.where().field("userid").eq(FblaLogon.getUserId()
).execute().get();
                    for (SaleItem item : result) {
                        mSaleItems.add(item);
                    }
                } catch (Exception exception) {
                    Log.e("MySalesController", exception.toString());
                }
                return null;
            }
            @Override
            protected void onPostExecute(Void v) {
                for (RecyclerView.Adapter adapter : mAdapters) {
                    adapter.notifyDataSetChanged();
                }
            }
        }.execute();
    }
}
```

```java
/* CommentListController.java
   ============================================================================
                         Josh Talley and Daniel O'Donnell
                                Dulaney High School
                      Mobile Application Development 2016-17
   ============================================================================
   Purpose: Used by CommentList to control access to the list of comments for
   a selected item. Attaching a recycler view to the class so that when the list of
   items is refreshed or changed, the recycler view is notified of that change.
*/
package com.fbla.dulaney.fblayardsale.controller;

import android.os.AsyncTask;
import android.support.v7.widget.RecyclerView;
import android.util.Log;

import com.fbla.dulaney.fblayardsale.FblaLogon;
import com.fbla.dulaney.fblayardsale.model.Account;
import com.fbla.dulaney.fblayardsale.model.ItemComment;
import com.fbla.dulaney.fblayardsale.model.SaleItem;
import com.microsoft.windowsazure.mobileservices.MobileServiceList;
import com.microsoft.windowsazure.mobileservices.table.MobileServiceTable;

import java.util.ArrayList;

public class CommentListController {
    private static ArrayList<ItemComment> mComments = new ArrayList<>();
    private static ArrayList<RecyclerView.Adapter> mAdapters = new ArrayList<>();
    private static MobileServiceTable<ItemComment> mItemCommentTable;
    private static SaleItem mItem;

    public static void AttachAdapter(RecyclerView.Adapter adapter) {
        mAdapters.add(adapter);
    }
    public static void RemoveAdapter(RecyclerView.Adapter adapter) { mAdapters.remove(
adapter); }

    public static int getCommentCount() {
        return mComments.size();
    }

    public static ItemComment getComment(int position) {
        if (mComments.size() > position) return mComments.get(position);
        else return null;
    }

    // Add a comment and notify the adapter of the change
    public static void addComment(ItemComment comment) {
        mComments.add(comment);
        for (RecyclerView.Adapter adapter : mAdapters) {
            adapter.notifyDataSetChanged();
        }
    }

    // Remove a comment and notify the adapter of the change
    public static void removeComment(int position) {
        mComments.remove(position);
        for (RecyclerView.Adapter adapter : mAdapters) {
            adapter.notifyDataSetChanged();
        }
    }

    public static SaleItem getItem() { return mItem; }
    public static void setItem(SaleItem item) { mItem = item; }

    // Refresh all comments and notify the adapter of the change
    public static void Refresh() {
        if (!FblaLogon.getLoggedOn()) return;
        mComments.clear();

        mItemCommentTable = FblaLogon.getClient().getTable(ItemComment.class);
        final MobileServiceTable<Account> mAccountTable = FblaLogon.getClient().getTable(Account.class);
        new AsyncTask<Void, Void, Void>() {
            @Override
            protected Void doInBackground(Void... params) {
                try {
                    final MobileServiceList<ItemComment> result =
                            mItemCommentTable.where().field("itemid").eq(mItem.getId()).execute().get();
                    for (ItemComment comment : result) {
                        Account account = mAccountTable.lookUp(comment.getUserId()).get();
                        comment.setAccount(account);
                        mComments.add(comment);
                    }
                } catch (Exception exception) {
                    Log.e("CommentListController", exception.toString());
                }
                return null;
            }
            @Override
            protected void onPostExecute(Void v) {
                for (RecyclerView.Adapter adapter : mAdapters) {
                    adapter.notifyDataSetChanged();
                }
            }
        }.execute();
    }
}
```