Message.java

```
package com.connexia.sqlite;

public class Message {
  private long id;
  private String message;

  public long getId() {
    return id;
  }

  public void setId(long id) {
    this.id = id;
  }

  public String getMessage() {
    return message;
  }

  public void setMessage(String message) {
    this.message = message;
  }

  // Will be used by the ArrayAdapter in the ListView
  @Override
  public String toString() {
    return message;
  }
}
```

MessagesDataSource.java

```
package com.connexia.sqlite;

import java.util.ArrayList;
import java.util.List;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;

public class MessagesDataSource {

  // Database fields
  private SQLiteDatabase database;
  private MySQLiteHelper dbHelper;
  private String[] allColumns = { MySQLiteHelper.COLUMN_ID,
      MySQLiteHelper.COLUMN_MESSAGE };

  public MessagesDataSource(Context context) {
    dbHelper = new MySQLiteHelper(context);
  }

  public void open() throws SQLException {
    database = dbHelper.getWritableDatabase();
  }

  public void close() {
    dbHelper.close();
  }

  public Message createMessage(String message) {
    ContentValues values = new ContentValues();
    values.put(MySQLiteHelper.COLUMN_MESSAGE, message);
    long insertId = database.insert(MySQLiteHelper.TABLE_MESSAGES,
null,
        values);
    Cursor cursor = database.query(MySQLiteHelper.TABLE_MESSAGES,
        allColumns, MySQLiteHelper.COLUMN_ID + " = " + insertId, null,
        null, null, null);
    cursor.moveToFirst();
    Message newMessage = cursorToMessage(cursor);
    cursor.close();
    return newMessage;
  }

  public void deleteMessage(Message message) {
    long id = message.getId();
    System.out.println("Message deleted with id: " + id);
    database.delete(MySQLiteHelper.TABLE_MESSAGES,
MySQLiteHelper.COLUMN_ID
        + " = " + id, null);
  }

  public void deleteAllMessage() {

          database.delete(MySQLiteHelper.TABLE_MESSAGES, null, null);
  }
  public List<Message> getAllMessages() {
    List<Message> messages = new ArrayList<Message>();

    Cursor cursor = database.query(MySQLiteHelper.TABLE_MESSAGES,
        allColumns, null, null, null, null, "_id DESC");

    cursor.moveToFirst();
    while (!cursor.isAfterLast()) {
      Message message = cursorToMessage(cursor);
      messages.add(message);
      cursor.moveToNext();
    }
    // Make sure to close the cursor
    cursor.close();
    return messages;
  }

  private Message cursorToMessage(Cursor cursor) {
    Message message = new Message();
    message.setId(cursor.getLong(0));
    message.setMessage(cursor.getString(1));
    return message;
  }
}
```

MySQLiteHelper.java

```
package com.connexia.sqlite;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

public class MySQLiteHelper extends SQLiteOpenHelper {

  public static final String TABLE_MESSAGES = "messages";
  public static final String COLUMN_ID = "_id";
  public static final String COLUMN_MESSAGE = "message";

  public static final String TABLE_TV = "tv";
  public static final String COLUMN_IP = "_ip";
  public static final String COLUMN_ALIAS = "alias";

  private static final String DATABASE_NAME = "connexiaLite.db";
  private static final int DATABASE_VERSION = 1;

  // Database creation sql statement
  private static final String DATABASE_CREATE = "create table "
      + TABLE_MESSAGES + "(" + COLUMN_ID
      + " integer primary key autoincrement, " + COLUMN_MESSAGE
      + " text not null);";
  private static final String DATABASE_CREATE2 = "create table "
            + TABLE_TV + "(" + COLUMN_IP
            + " string primary key, " + COLUMN_ALIAS
            + " text not null);";

  public MySQLiteHelper(Context context) {
    super(context, DATABASE_NAME, null, DATABASE_VERSION);
  }

  @Override
  public void onCreate(SQLiteDatabase database) {
    database.execSQL(DATABASE_CREATE);
    database.execSQL(DATABASE_CREATE2);
  }

  @Override
  public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
    Log.w(MySQLiteHelper.class.getName(),
        "Upgrading database from version " + oldVersion + " to "
            + newVersion + ", which will destroy all old data");
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_MESSAGES);
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_TV);
    onCreate(db);
  }

}
```

Tv.java

```java
package com.connexia.sqlite;

public class Tv {
  private String ip;
  private String alias;

  public String getIp() {
    return ip;
  }

  public void setIp(String ip) {
    this.ip = ip;
  }

  public String getAlias() {
    return alias;
  }

  public void setAlias(String alias) {
    this.alias = alias;
  }

  // Will be used by the ArrayAdapter in the ListView
  @Override
  public String toString() {
    return alias;
  }
}
```

TvDataSource.java

```
package com.connexia.sqlite;

import java.util.ArrayList;
import java.util.List;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;

public class TvDataSource {

  // Database fields
  private SQLiteDatabase database;
  private MySQLiteHelper dbHelper;
  private String[] allColumns = { MySQLiteHelper.COLUMN_IP,
      MySQLiteHelper.COLUMN_ALIAS };

  public TvDataSource(Context context) {
    dbHelper = new MySQLiteHelper(context);
  }

  public void open() throws SQLException {
    database = dbHelper.getWritableDatabase();
  }

  public void close() {
    dbHelper.close();
  }

  public Tv createTv(String alias) {
    ContentValues values = new ContentValues();
    values.put(MySQLiteHelper.COLUMN_ALIAS, alias);
    long insertIp = database.insert(MySQLiteHelper.TABLE_TV, null,
values);

    Cursor cursor = database.query(MySQLiteHelper.TABLE_TV,
        allColumns, MySQLiteHelper.COLUMN_IP + " = " + insertIp, null,
        null, null, null);
    cursor.moveToFirst();
    Tv newTv = cursorToTv(cursor);
    cursor.close();
    return newTv;
  }

  public void deleteTv(Tv tv) {
    String ip = tv.getIp();
    System.out.println("Tv deleted with id: " + ip);
    database.delete(MySQLiteHelper.TABLE_TV, MySQLiteHelper.COLUMN_IP
        + " = " + ip, null);
  }

  public List<Tv> getAllTv() {
    List<Tv> tvs = new ArrayList<Tv>();

    Cursor cursor = database.query(MySQLiteHelper.TABLE_TV,
        allColumns, null, null, null, null, null);
    cursor.moveToFirst();
    while (!cursor.isAfterLast()) {
      Tv tv = cursorToTv(cursor);
      tvs.add(tv);
      cursor.moveToNext();
    }
    // Make sure to close the cursor
    cursor.close();
    return tvs;
  }

  private Tv cursorToTv(Cursor cursor) {
      Tv tv = new Tv();
      tv.setIp(cursor.getString(0));
      tv.setAlias(cursor.getString(1));
    return tv;
  }
}
```

MainActivity.java

```
package com.connexia.main;

import android.app.ActionBar;
import android.app.FragmentTransaction;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentActivity;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentPagerAdapter;
import android.support.v4.view.ViewPager;
import android.view.Menu;
import android.view.View;

import com.connexia.fragments.ListMessagesFragment;
import com.connexia.fragments.NewMessageFragment;
//import android.widget.TextView;

public class MainActivity extends FragmentActivity implements
ActionBar.TabListener {

    /**
     * The {@link android.support.v4.view.PagerAdapter} that will
provide
     * fragments for each of the sections. We use a
     * {@link android.support.v4.app.FragmentPagerAdapter} derivative,
which
     * will keep every loaded fragment in memory. If this becomes too
memory
     * intensive, it may be best to switch to a
     * {@link android.support.v4.app.FragmentStatePagerAdapter}.
     */
    SectionsPagerAdapter mSectionsPagerAdapter;

    /**
     * The {@link ViewPager} that will host the section contents.
     */
    ViewPager mViewPager;
    NewMessageFragment newMessageFragment;
    ListMessagesFragment listMessagesFragment;

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

        // Set up the action bar.
        final ActionBar actionBar = getActionBar();
        actionBar.setNavigationMode(ActionBar.NAVIGATION_MODE_TABS);

        // Create the adapter that will return a fragment for each of
the three
        // primary sections of the app.
        mSectionsPagerAdapter = new
SectionsPagerAdapter(getSupportFragmentManager());

        // Set up the ViewPager with the sections adapter.
        mViewPager = (ViewPager) findViewById(R.id.pager);
        mViewPager.setAdapter(mSectionsPagerAdapter);
        // When swiping between different sections, select the
corresponding
        // tab. We can also use ActionBar.Tab#select() to do this if
we have
        // a reference to the Tab.
        mViewPager.setOnPageChangeListener(new
ViewPager.SimpleOnPageChangeListener() {
            @Override
            public void onPageSelected(int position) {
                actionBar.setSelectedNavigationItem(position);
            }
        });

        // For each of the sections in the app, add a tab to the
action bar.
        for (int i = 0; i < mSectionsPagerAdapter.getCount(); i++) {
            // Create a tab with text corresponding to the page title
defined by
            // the adapter. Also specify this Activity object, which
implements
            // the TabListener interface, as the callback (listener)
for when
            // this tab is selected.
            actionBar.addTab(
                    actionBar.newTab()

.setText(mSectionsPagerAdapter.getPageTitle(i))
                            .setTabListener(this));

        }

    }


      @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it
is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }


    @Override
    public void onTabSelected(ActionBar.Tab tab, FragmentTransaction
fragmentTransaction) {
        // When the given tab is selected, switch to the corresponding
page in
        // the ViewPager.
        mViewPager.setCurrentItem(tab.getPosition());
    }

    @Override
    public void onTabUnselected(ActionBar.Tab tab, FragmentTransaction
fragmentTransaction) {
    }

    @Override
    public void onTabReselected(ActionBar.Tab tab, FragmentTransaction
fragmentTransaction) {
    }
    public void sendMessage(View v) {
      newMessageFragment.sendMessage(v, listMessagesFragment);
    }

    /**
     * A {@link FragmentPagerAdapter} that returns a fragment
corresponding to
     * one of the sections/tabs/pages.
     */
    public class SectionsPagerAdapter extends FragmentPagerAdapter {

        public SectionsPagerAdapter(FragmentManager fm) {
            super(fm);
        }

        @Override
        public Fragment getItem(int position) {
            // getItem is called to instantiate the fragment for the
given page.
            // Return a NewMessageFragment (defined as a static inner
class
            // below) with the page number as its lone argument.

            Fragment fragment = null;
            //mostrem fragments segons pantalla
            switch (position) {
            case 0:
                  listMessagesFragment = new ListMessagesFragment();
                  fragment             = listMessagesFragment;

                break;
            case 1:
                  newMessageFragment = new NewMessageFragment();
                fragment           = newMessageFragment;
                break;
            }
            return fragment;
        }

        @Override
        public int getCount() {
            // Show 3 total pages.
            return 2;
        }

        @Override
        public CharSequence getPageTitle(int position) {

            switch (position) {
            case 0:
                  return "History";
            case 1:
                  return "New message";
            }
            return null;
        }
    }

}
```

DiscoverDevicesAndServices.java

```
package com.connexia.bluetooth;
import java.util.ArrayList;
import java.util.Set;

import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import com.connexia.main.R;

public class DiscoverDevicesAndServices extends Activity {
  ListView out, out2;
  private final int REQUEST_ENABLE_BT = 1;
  private final int REQUEST_PAIRED_DEVICE = 2;
  private BluetoothAdapter mBluetoothAdapter;
  private ArrayList<BluetoothDevice> btDeviceList = new
ArrayList<BluetoothDevice>();


  public DiscoverDevicesAndServices() {

  }

  /** Called when the activity is first created. */
  @Override
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.listdevicesbt);

    out = (ListView) findViewById(R.id.out);
    out2 = (ListView) findViewById(R.id.out2);

    //out.setMovementMethod(new ScrollingMovementMethod());

    //Register the BroadcastReceiver
    IntentFilter filter = new
IntentFilter(BluetoothDevice.ACTION_FOUND);
    filter.addAction(BluetoothDevice.ACTION_UUID);
    filter.addAction(BluetoothAdapter.ACTION_DISCOVERY_STARTED);
    filter.addAction(BluetoothAdapter.ACTION_DISCOVERY_FINISHED);
    registerReceiver(ActionFoundReceiver, filter); // Don't forget to
unregister during onDestroy

    // Getting the Bluetooth adapter
    mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

    CheckBTState();
  }

  /* This routine is called when an activity completes.*/
  @Override
  protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == REQUEST_ENABLE_BT) {
      CheckBTState();
    }
    if (requestCode == REQUEST_PAIRED_DEVICE){
         if(resultCode == RESULT_OK){
               //emparellar dispositius
         }
      }
  }


  private void CheckBTState() {
    // Check for Bluetooth support and then check to make sure it is
turned on
    // If it isn't request to turn it on
    // List paired devices
    // Emulator doesn't support Bluetooth and will return null
    if(mBluetoothAdapter==null) {
      //out.append("\nBluetooth NOT supported. Aborting.");
      Toast.makeText(this, "Bluetooth no disponible",
Toast.LENGTH_LONG).show();
      return;
    } else {

      if (mBluetoothAdapter.isEnabled()) {
        //out.append("\nBluetooth habilitat...");
      if (mBluetoothAdapter.isDiscovering()){
            mBluetoothAdapter.cancelDiscovery();
        }
        // Starting the device discovery
        mBluetoothAdapter.startDiscovery();
      } else {
        Intent enableBtIntent = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
        startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);
      }

          //Listing paired devices
            ArrayList<String> s = new ArrayList<String>();

            //s.add("\nDISPOSITIUS SINCRONITZATS:");
            Set<BluetoothDevice> pairedDevices  =
mBluetoothAdapter.getBondedDevices();

            for (BluetoothDevice device : pairedDevices ) {
                  s.add("\n" + device.getName() + " (" +
device.getAddress() + ")");
                  //out.append("\nNOM: " + device.getName() + " ADREÇA:
" + device.getAddress());
            }

            ArrayAdapter<String> arrayAdapter =
                     new
ArrayAdapter<String>(this,android.R.layout.simple_list_item_1, s);
                     out.setAdapter(arrayAdapter);
    }
  }



  private final BroadcastReceiver ActionFoundReceiver = new
BroadcastReceiver(){

    @Override
    public void onReceive(Context context, Intent intent) {
           String action = intent.getAction();



           if(BluetoothDevice.ACTION_FOUND.equals(action)) {
             BluetoothDevice device =
intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
             //out.append("\n" + device.getName() + " (" + device+
")");
             btDeviceList.add(device);
             System.out.println("\n\nDins");

//           ArrayAdapter<String> arrayAdapter =
//                   new
ArrayAdapter<String>(this,android.R.layout.simple_list_item_1,
btDeviceList);
//           out2.setAdapter(arrayAdapter);

           } else {

if(BluetoothAdapter.ACTION_DISCOVERY_STARTED.equals(action)) {
                   //out.append("\nCERCANT NOUS DISPOSITIUS...");
                   TextView text = (TextView)
findViewById(R.id.llistaNousBT);
                   text.setText("CERCANT NOUS DISPOSITIUS...");
             } else {

if(BluetoothAdapter.ACTION_DISCOVERY_FINISHED.equals(action)) {
//                       TextView textCercaFinal = (TextView)
findViewById(R.id.cercaFinalitzada);
//                       textCercaFinal.setVisibility(View.VISIBLE);

                         //out.append("\n\nCERCA FINALITZADA");
                   }
             }
           }
    }
  };

  @Override
  protected void onDestroy() {
    super.onDestroy();
    if (mBluetoothAdapter != null) {
      mBluetoothAdapter.cancelDiscovery();
    }
    unregisterReceiver(ActionFoundReceiver);
  }

}
```

ListMessagesFragment.java

```java
package com.connexia.fragments;

import java.util.List;

import android.app.AlertDialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.ListFragment;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemLongClickListener;
import android.widget.ArrayAdapter;
import android.widget.Toast;

import com.connexia.sqlite.Message;
import com.connexia.sqlite.MessagesDataSource;


public class ListMessagesFragment extends ListFragment {
  private MessagesDataSource datasource;
  private List<Message> values;
  private ArrayAdapter<Message> adapter;
  private FragmentManager fr;

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

      fr = this.getFragmentManager();

      datasource = new MessagesDataSource(this.getActivity());
      datasource.open();

      values = datasource.getAllMessages();

      // Use the SimpleCursorAdapter to show the
      // elements in a ListView
      adapter = new ArrayAdapter<Message>(this.getActivity(),
                  android.R.layout.simple_list_item_1, values);
      setListAdapter(adapter);

      adapter.notifyDataSetChanged();

  }

  @Override
  public void onActivityCreated(Bundle savedInstanceState) {

        super.onActivityCreated(savedInstanceState);
        Log.v("DetailFragment", "onActivityCreated()");
        this.getListView().setLongClickable(true);

        this.getListView().setOnItemLongClickListener(new
OnItemLongClickListener() {
```

```
              public boolean onItemLongClick(AdapterView<?> arg0, View
arg1, int pos, long id) {

                    final Message msg = values.get(pos);

                    AlertDialog.Builder builder = new
AlertDialog.Builder(getActivity());
                    builder.setMessage("Què vols fer?");
                    builder.setCancelable(true);

                    builder.setNegativeButton("Reenviar", new
DialogInterface.OnClickListener() {

                              @Override
                              public void onClick(DialogInterface
dialog, int which) {
                                    //Message msg = values.get(pos);
                              Log.v("missatge","msg"+": " +msg);

                              }
                    });
                    builder.setNeutralButton("Esborrar", new
DialogInterface.OnClickListener() {

                          @Override
                          public void onClick(DialogInterface dialog,
int which) {

                              datasource.deleteMessage(msg);
                              adapter.clear();
                        adapter.addAll(datasource.getAllMessages());
                        adapter.notifyDataSetChanged();
                        Toast toast = Toast.makeText(getActivity(),
"Missatge esborrat correctament", Toast.LENGTH_SHORT);
                        toast.show();
                          }
                    });
                    builder.setPositiveButton("Esborrar historial", new
DialogInterface.OnClickListener() {

                              @Override
                      public void onClick(DialogInterface dialog, int
which) {

                                    datasource.deleteAllMessage();
                                    adapter.clear();

adapter.addAll(datasource.getAllMessages());
                              adapter.notifyDataSetChanged();
                              Toast toast =
Toast.makeText(getActivity(), "Historial eliminat correctament",
Toast.LENGTH_SHORT);
                              toast.show();
                      }
                });

                    builder.show();
                    return false;
                  }

        });
        this.getListView().setOnItemClickListener(new
OnItemClickListener() {

            @Override
            public void onItemClick(AdapterView<?> arg0, View arg1, int
arg2,
                        long arg3) {
            //Log.v(" clicked","pos"+" "+69);
            Message msg = values.get(arg2);
            Log.v("missatge","msg"+": " +msg);
            final Bundle arguments = new Bundle();
            arguments.putLong(NewMessageFragment.ARG_SECTION_NUMBER,
msg.getId());
            final NewMessageFragment fragment = new
NewMessageFragment();
            fragment.setArguments(arguments);
            //fr.beginTransaction().replace(R.id.fragment_detail,
fragment).commit();
            }

        });

  }

  @Override
      public void onDestroy(){
      super.onDestroy();
      if (datasource != null) {
            datasource.close();
      }
  }

  @Override
      public void onResume(){
      super.onResume();
      if (datasource != null) {
            datasource.open();
      }
  }
}
```

NewMessageFragment.java

```
package com.connexia.fragments;

import java.io.ByteArrayOutputStream;
import java.io.IOException;

import org.apache.http.HttpResponse;
import org.apache.http.HttpStatus;
import org.apache.http.StatusLine;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.protocol.BasicHttpContext;
import org.apache.http.protocol.HttpContext;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.os.StrictMode;
import android.support.v4.app.Fragment;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.inputmethod.InputMethodManager;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.Toast;

import com.connexia.bluetooth.BluetoothListDevicesActivity;
import com.connexia.main.R;
import com.connexia.sqlite.Message;
import com.connexia.sqlite.MessagesDataSource;
import com.connexia.wifi.LinearLayout9;
//import com.connexia.bluetooth.DiscoverDevicesAndServices;
   /**
 * A new message fragment representing a section of the app, but that
simply
 * displays text.
 */
public class NewMessageFragment extends Fragment {
    /**
     * The fragment argument representing the section number for this
     * fragment.
     */
    public static final String ARG_SECTION_NUMBER = "section_number";
    EditText etNombreParaInsertar;
    private MessagesDataSource datasource;
    EditText etMessage;
    //private String TAG = "TGtracker";

    public NewMessageFragment() {

    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
            Bundle savedInstanceState) {
        View rootView =
inflater.inflate(R.layout.fragment_new_message, container, false);

      datasource = new MessagesDataSource(this.getActivity());
      datasource.open();

      etMessage = (EditText) rootView.findViewById(R.id.etMessage);

      StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
      StrictMode.setThreadPolicy(policy);

        return rootView;
    }

    public void sendMessage (View view, ListMessagesFragment
listMessagesFragment) {

      String missatge = etMessage.getText().toString();
      //Log.d("missatge", missatge);
        int duration = Toast.LENGTH_SHORT;

        System.out.println("missatgePrimer: " + missatge);

        switch (view.getId()) {
        case R.id.btnSend:

            //controlem que hi hagi missatge i no hi hagi un espai
            if(missatge.trim().length()==0) {

                        Toast toast = Toast.makeText(getActivity(),
"ERROR: Missatge buit", duration);
                        toast.setGravity(Gravity.CENTER, 0, 60);
                  toast.show();
                  }
                  else {

                        //afegim missatge a la bbdd i actualitzem
historic
                        Message message =
datasource.createMessage(missatge);
                        ArrayAdapter<Message> listMessageAdapter =
(ArrayAdapter<Message>) listMessagesFragment.getListAdapter();
                        listMessageAdapter.add(message);
                        listMessageAdapter.notifyDataSetChanged();


                        //Amaguem teclat
                        InputMethodManager mgr = (InputMethodManager)
getActivity().getSystemService(Context.INPUT_METHOD_SERVICE);

mgr.hideSoftInputFromWindow(etMessage.getWindowToken(), 0);


                AlertDialog.Builder builder = new
AlertDialog.Builder(getActivity());
                    builder.setMessage("Envia el missatge via: ");
                    builder.setCancelable(false);
                    builder.setPositiveButton("Bluetooth", new
DialogInterface.OnClickListener() {

                          @Override
                          public void onClick(DialogInterface dialog,
int which) {
                                startActivity(new Intent(getActivity(),
BluetoothListDevicesActivity.class));
                          }
                    });
                    builder.setNegativeButton("Wifi", new
DialogInterface.OnClickListener() {

                              @Override
                              public void onClick(DialogInterface
dialog, int which) {

                                    //startActivity(new
Intent(getActivity(), setupWifi.class));
                                    //startActivity(new
Intent(getActivity(), LinearLayout9.class));


//                                  WifiManager wifi = (WifiManager)
getActivity().getSystemService(Context.WIFI_SERVICE);
//                              if (wifi.isWifiEnabled()==true) {
//                                  System.out.println("wifi activat");
//                              } else {
//                                try {
//                                  wifi.setWifiEnabled(true);
//                                } catch (Exception e){
//
//                                }

//                              }


                                    //if (wifi.isWifiEnabled()) {

                                          String missatge =
etMessage.getText().toString();
                                          System.out.println("missatge:
" + missatge);
                                          missatge =
missatge.replaceAll(" ", "%20");
                                          missatge =
missatge.replaceAll("'", "''");
                                          //missatge =
missatge.replaceAll("[\\r\\n]", "");

      System.out.println("missatgeok: " + missatge);

                                          //String URL =
"http://192.168.1.15/WS_TV/webservice.php?accio=insertarMissatge&missa
tge=" + missatge;
                                          //String URL =
"http://192.168.1.69/WS_TV/webservice.php?accio=insertarMissatge&missa
tge=" + missatge;
                                          System.out.println("URL: " +
URL);
```

```
                                         try {

                                               HttpContext
localContext = new BasicHttpContext();

                                               HttpClient httpclient =
new DefaultHttpClient();
                                               HttpGet httpget =new
HttpGet(URL);
                                               HttpResponse response =
httpclient.execute(httpget, localContext);


                                               StatusLine statusLine =
response.getStatusLine();



      if(statusLine.getStatusCode() == HttpStatus.SC_OK){


ByteArrayOutputStream out = new ByteArrayOutputStream();

response.getEntity().writeTo(out);
                                                   out.close();
                                                   //String
responseString = out.toString();

//System.out.println("Response String: " + responseString);

etMessage.setText("");

                                               } else{
                                                   //Close the
connection.

response.getEntity().getContent().close();
                                                   throw new
IOException(statusLine.getReasonPhrase());
                                               }
                                         } catch
(ClientProtocolException e) {
                                               e.printStackTrace();
                                         } catch
(IllegalStateException e) {
                                               e.printStackTrace();
                                         } catch (IOException e) {

                                               e.printStackTrace();
                                         }

//                                  } else {
//                                       Toast toast3 =
Toast.makeText(getActivity(), "Wifi no connectat",
Toast.LENGTH_SHORT);
//                                  toast3.show();
//                                  }


                              }
```

```
                    });

                    builder.show();

                  }
            break;
        }
    }



    @Override
      public void onDestroy(){
      super.onDestroy();
      if (datasource != null) {
            datasource.close();
      }
    }

    @Override
      public void onResume(){
      super.onResume();
      if (datasource != null) {
            datasource.open();
      }
    }
}
```

fragment_new_message.xml

```xml
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/fragment_detail"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity$NewMessageFragment" >

    <EditText
        android:id="@+id/etMessage"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_toLeftOf="@+id/btnSend"
        android:ems="10"
        android:gravity="bottom|left"
        android:hint="@string/hint"
        android:inputType="textMultiLine"
        android:maxLength="130"
         />

    <Button
        android:id="@+id/btnSend"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:onClick="sendMessage"
        android:text="@string/send" />

</RelativeLayout>
```

Main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/wrapper"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical" >

      <android.support.v4.view.ViewPager
             xmlns:tools="http://schemas.android.com/tools"
          android:id="@+id/pager"
          android:layout_width="fill_parent"
          android:layout_height="fill_parent"
          tools:context=".MainActivity" >
      </android.support.v4.view.ViewPager>

    <ListView
          android:id="@android:id/list"
          android:layout_width="match_parent"
          android:layout_height="wrap_content" />
</LinearLayout>
```

Strings.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">Connexia</string>
    <string name="action_settings">Settings</string>
    <string name="title_section1">Section 1</string>
    <string name="title_section2">Section 2</string>
    <string name="title_section3">Section 3</string>


    <string name="hint">Escriu un missatge via Wifi (màx 130
caràcters)</string>
    <string name="send">Send</string>
    <string name="titolSincronitzats">DISPOSITIUS SINCRONITZATS</string>
    <string name="titolCercarNousBT">DISPOSITIUS DISPONIBLES</string>
    <string name="cercaFinalitzada">CERCA FINALITZADA</string>
    <string name="scan">CERCAR NOUS DISPOSITIUS</string>
    <string name="botoAfegirTv">AFEGIR TV</string>

</resources>
```

addtv_popup.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/popup"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Ip:" />

    <EditText
        android:id="@+id/etIp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="text" >

        <requestFocus />
    </EditText>

    <TextView
        android:id="@+id/textView2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="5dp"
        android:text="Àlies:" />

    <EditText
        android:id="@+id/etAlias"
        android:layout_width="242dp"
        android:layout_height="wrap_content"
        android:inputType="text" >

    </EditText>

    <Button
        android:id="@+id/afegirTV"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Afegir" />

</LinearLayout>
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.connexia.main"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="11"
        android:targetSdkVersion="16" />
      <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
    <uses-permission android:name="android.permission.BLUETOOTH" />
      <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
      <uses-permission android:name="android.permission.INTERNET" />
      <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"
/>
      <uses-permission android:name="android.permission.CHANGE_WIFI_STATE"
/>
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.connexia.main.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity

android:name="com.connexia.bluetooth.BluetoothListDevicesActivity"
            android:label="@string/app_name" >
        </activity>
        <activity
            android:name="com.connexia.wifi.LinearLayout9"
            android:label="@string/app_name" >
        </activity>
    </application>

</manifest>
```