

Hong Kong Institute of Vocational Education (Tsing Yi)
Department of Information and Communications Technology

Higher Diploma in Software Engineering

Final Year Project (ITD3309)

Final Report - Code

(2011/2012)

Distance Learning on Android Tablet – LearnAble!

Supervisor: Clarence Lau
Esther Yuen
Student: Tai Ching Kiu
Fung Ka Yiu
Wong Ho Fung

We declare that this is a group project and that no part of this submission has been copied from any other student's work or from any other source except where due acknowledgement is made explicitly in the text, nor has any part been written for us by another person.

Student	Contribution to the project (%) (Total 100%)	Signature
Tai Ching Kiu	40	
Fung Ka Yiu	40	
Wong Ho Fung	20	

LessonMain.java

```
package com.curtis.fyp.lesson;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.InputStream;

import org.apache.http.HttpResponse;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.util.EntityUtils;
import org.jivesoftware.smack.XMPPException;
import org.jivesoftware.smackx.muc.MultiUserChat;
import org.json.JSONArray;
import org.json.JSONObject;

import com.asmack.ActivityMain;
import com.curtis.fyp.Login;
import com.curtis.fyp.R;

import android.app.Activity;
import android.app.FragmentManager;
import android.app.FragmentTransaction;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.widget.AdapterView;
import android.widget.Toast;

public class LessonMain extends Activity {
    Uri outputFileUri;
    private static final int ACTIVITY_SELECT_IMAGE = 100;
    private static final int ACTIVITY_TAKE_IMAGE = 101;
    public static String lessonID = "";
    public static String live = ""; // 1- live 2 -record
    public static String position = "";
```

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    // Set Information  
    lessonID = getIntent().getStringExtra("lessonID");  
    live = getIntent().getStringExtra("live");  
    // position = getIntent().getStringExtra("position");  
    position = Login.loginPosition;  
  
    // Create View  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.lesson_main);  
  
    // Control Fragment Display  
    FragmentManager fm = getFragmentManager();  
    FragmentTransaction ft = getFragmentManager().beginTransaction();  
    ft.setCustomAnimations(android.R.animator.fade_in,  
        android.R.animator.fade_out);  
    ft.hide(fm.findFragmentById(R.id.LessonMainRightAskQuestion));  
    ft.hide(fm.findFragmentById(R.id.LessonMainRightChat));  
    ft.show(fm.findFragmentById(R.id.LessonMainRightPowerpoint));  
    ft.hide(fm.findFragmentById(R.id.LessonMainRightWhiteBoard));  
    ft.hide(fm.findFragmentById(R.id.LessonMainRightAttendance));  
    ft.commit();  
  
    markAttend();  
    sendLessonStatus();  
}
```

```
public static boolean isTeacher() {  
    // return true; // TODO Testing Purpose  
    return position.equals("teacher");  
}
```

```
public static boolean isLive() {  
    return live.equals("1");  
}
```

@Override

```
public boolean onCreateOptionsMenu(Menu menu) {  
    MenuInflater inflater = getMenuInflater();
```

```

if (isTeacher()) {
    inflater.inflate(R.menu.teacherlesson, menu);
} else {
    inflater.inflate(R.menu.lesson, menu);
}
return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    FragmentManager fm = getFragmentManager();
    FragmentTransaction ft = getFragmentManager().beginTransaction();
    ft.setCustomAnimations(android.R.animator.fade_in,
        android.R.animator.fade_out);

    switch (item.getItemId()) {
        case R.id.askQuestion:
            ft.show(fm.findFragmentById(R.id.LessonMainLeft));
            ft.show(fm.findFragmentById(R.id.LessonMainRightAskQuestion));
            ft.hide(fm.findFragmentById(R.id.LessonMainRightChat));
            ft.hide(fm.findFragmentById(R.id.LessonMainRightWhiteBoard));
            ft.hide(fm.findFragmentById(R.id.LessonMainRightPowerpoint));
            ft.hide(fm.findFragmentById(R.id.LessonMainRightAttendance));
            ft.commit();
            return true;

        case R.id.Chat: // Cast to individual Chat
            ft.show(fm.findFragmentById(R.id.LessonMainLeft));
            ft.hide(fm.findFragmentById(R.id.LessonMainRightAskQuestion));
            ft.show(fm.findFragmentById(R.id.LessonMainRightChat));
            ft.hide(fm.findFragmentById(R.id.LessonMainRightWhiteBoard));
            ft.hide(fm.findFragmentById(R.id.LessonMainRightPowerpoint));
            ft.hide(fm.findFragmentById(R.id.LessonMainRightAttendance));
            ft.commit();
            return true;

        case R.id.whiteBoard:
            ft.show(fm.findFragmentById(R.id.LessonMainLeft));
            ft.hide(fm.findFragmentById(R.id.LessonMainRightAskQuestion));
            ft.hide(fm.findFragmentById(R.id.LessonMainRightChat));
            ft.show(fm.findFragmentById(R.id.LessonMainRightWhiteBoard));
            ft.hide(fm.findFragmentById(R.id.LessonMainRightPowerpoint));
            ft.hide(fm.findFragmentById(R.id.LessonMainRightAttendance));
            ft.commit();
    }
}

```

```

        return true;
    case R.id.powerPoint:
        ft.show(fm.findFragmentById(R.id.LessonMainLeft));
        ft.hide(fm.findFragmentById(R.id.LessonMainRightAskQuestion));
        ft.hide(fm.findFragmentById(R.id.LessonMainRightChat));
        ft.hide(fm.findFragmentById(R.id.LessonMainRightWhiteBoard));
        ft.show(fm.findFragmentById(R.id.LessonMainRightPowerpoint));
        ft.hide(fm.findFragmentById(R.id.LessonMainRightAttendance));
        ft.commit();
        return true;
    case R.id.attendance:
        ft.show(fm.findFragmentById(R.id.LessonMainLeft));
        ft.hide(fm.findFragmentById(R.id.LessonMainRightAskQuestion));
        ft.hide(fm.findFragmentById(R.id.LessonMainRightChat));
        ft.hide(fm.findFragmentById(R.id.LessonMainRightWhiteBoard));
        ft.hide(fm.findFragmentById(R.id.LessonMainRightPowerpoint));
        ft.show(fm.findFragmentById(R.id.LessonMainRightAttendance));
        ft.commit();
        return true;
    default:
        return super.onOptionsItemSelected(item);
    }
}

```

```

public void onDestroy() {
    markLeave();
    live = "2";
    sendLessonStatus();
    super.onDestroy();
}

```

@Override

```

public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    Log.d("hello", "Get Result");

    switch (requestCode) {
    case ACTIVITY_SELECT_IMAGE:
        if (resultCode == RESULT_OK) {

```

```

            Uri selectedImage = data.getData();

```

```

        InputStream imageStream = null;
        try {
            imageStream = getContentResolver().openInputStream(
                selectedImage);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
        LessonMainRightWhiteBoard myFragment = (LessonMainRightWhiteBoard)
getFragmentManager()
                .findFragmentById(R.id.LessonMainRightWhiteBoard);
        myFragment.changeBackground(imageStream);
        // TODO
    }
    break;
case ACTIVITY_TAKE_IMAGE:
    if (resultCode == RESULT_OK) {
        File imgFile = new File("/sdcard/image.jpg");

        if (imgFile.exists()) {

            Bitmap myBitmap = BitmapFactory.decodeFile(imgFile
                .getAbsolutePath());
            // drawingSurface.setBackground(BitmapFactory
            // .decodeFile(imgFile.getAbsolutePath()));
            LessonMainRightWhiteBoard myFragment = (LessonMainRightWhiteBoard)
getFragmentManager()
                .findFragmentById(R.id.LessonMainRightWhiteBoard);
            myFragment.changeBackground2(imgFile);

        }
    }
    break;

}

private void markAttend() {
    sendAttendent(1);
}

private void markLeave() {
    sendAttendent(2);
}

```

```

    }

    private void sendAttendent(int status) {
        if (isTeacher()) {
            // No need to Send Attendance
        } else {
            try {
                HttpPost httppost = new HttpPost(
                    Login.webServerIp
                        +
"/Controller/studentAttendLessonMarkdown.php?myusername="
                        + Login.loginID + "&mylesson=" + lessonID
                        + "&attendstatus=" + status);
                HttpResponse httpResponse = new DefaultHttpClient()
                    .execute(httppost);
                String retSrc = EntityUtils.toString(httpResponse.getEntity());

            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }

    private void sendLessonStatus() {
        if (isTeacher()) {
            try {
                HttpPost httppost = new HttpPost(Login.webServerIp
                    + "/Controller/setlessonOn.php?mylesson=" + lessonID
                    + "&status=" + live);
                HttpResponse httpResponse = new DefaultHttpClient()
                    .execute(httppost);
                String retSrc = EntityUtils.toString(httpResponse.getEntity());

            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}

```

LessonMainLeft.java

```
package com.curtis.fyp.lesson;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Date;
import java.util.Iterator;

import java.util.List;

import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.util.EntityUtils;
import org.jivesoftware.smack.PacketListener;
import org.jivesoftware.smack.XMPPException;
import org.jivesoftware.smack.packet.Message;
import org.jivesoftware.smack.packet.Packet;
import org.jivesoftware.smack.packet.Presence;
import org.jivesoftware.smackx.Form;
import org.jivesoftware.smackx.FormField;
import org.jivesoftware.smackx.muc.MultiUserChat;
import org.jivesoftware.smackx.packet.DelayInformation;
import org.json.JSONArray;
import org.json.JSONObject;

import com.asmack.ActivityMain;
import com.asmack.MessageReceiver;
import com.asmack.OnMessageListener;
import com.curtis.fyp.Login;
import com.curtis.fyp.R;

import android.app.Fragment;
import android.app.FragmentManager;
import android.app.FragmentTransaction;
import android.net.Uri;
import android.os.Bundle;
import android.os.Handler;
```



```

import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.ViewGroup.LayoutParams;
import android.widget.AdapterView;
import android.view.View.OnClickListener;

import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.MediaController;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.ToggleButton;
import android.widget.VideoView;

public class LessonMainLeft extends Fragment implements OnClickListener,
    OnMessageListener, AdapterView.OnItemClickListener {
    VideoView videoview;
    ListView listview;
    String[] listItems;
    ToggleButton toggleButton;
    int toggleClick = 1;
    String url = Login.mediaServerIp + "/live/" + LessonMain.lessonID;
    String vodUrl = Login.vodServerIP + "/" + LessonMain.lessonID + ".mp4";

    // Chat
    private String TAG = "chat";
    private Thread mThread;

    private final int RECEIVE = 1;

    private Button send;
    private EditText record, etMessage;

    ArrayAdapter<String> buddylist;
    ArrayList<String> occpantsArray;

    public static MultiUserChat muc;
    private MessageReceiver mUpdateMessage;

```

```
public static String multiUserRoom = LessonMain.lessonID + "@conference."
    + Login.imDomain;// TODO to be confirm
```

@Override

```
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    // url = Login.mediaServerIp + "/live/" + LessonMain.lessonID;
    // vodUrl = Login.vodServerIP + "/" + LessonMain.lessonID + ".mp4";
    View result = inflater.inflate(R.layout.lesson_main_left, container);
    videoview = (VideoView) result
        .findViewById(R.id.lesson_main_left_videoView);
    listview = (ListView) result
        .findViewById(R.id.lesson_main_left_buddyList);
    send = (Button) result.findViewById(R.id.lesson_main_left_send);
    record = (EditText) result.findViewById(R.id.lesson_main_left_record);
    etMessage = (EditText) result
        .findViewById(R.id.lesson_main_left_message);
    toggleButton = (ToggleButton) result
        .findViewById(R.id.lesson_main_left_toggleButton);

    // / listview.setAdapter(ActivityMain.getRosterToListView(getActivity()
    // .getApplicationContext()));
    // ActivityMain.setUp();

    record.setVisibility(result.GONE);
    send.setOnClickListener(this);
    toggleButton.setOnClickListener(this);
    occpantArray = new ArrayList<String>();
    try {
        muc = new MultiUserChat(ActivityMain.connection, multiUserRoom);
        if (LessonMain.isTeacher()) {
            muc.create(LessonMain.lessonID);
            // Send an empty room configuration form which indicates that we
            // want
            // an instant room
            muc.sendConfigurationForm(new Form(Form.TYPE_SUBMIT));
        }
        muc.join(Login.mCurrentAccount);
        occpantArray.add("No People");
        buddylist = new ArrayAdapter<String>(getActivity(),
            android.R.layout.simple_list_item_1, occpantArray);
```

```

listview.setAdapter(buddylist);
listview.setOnItemClickListener(this);
PacketListener peopleListener = new PacketListener() {

    @Override
    public void processPacket(Packet packet) {
        // System.out.println("People PacketData:" +
        // packet.toXML());
        if (packet
            .getClass()
            .toString()
            .equalsIgnoreCase(
                "class org.jivesoftware.smack.packet.Presence")) {
            Presence p = (Presence) packet;

            try {
                System.out.println("Presence from: " + p.getFrom());
                System.out.println("Presence :" + p.toString());

                String[] username = p.getFrom().split("/");
                String t[] = username[0].split("@");
                if (p.toString().equalsIgnoreCase("available")) {
                    getActivity().runOnUiThread(new Runnable() {
                        public void run() {
                            record.setText("** Welcome new Student attend the
room **"
                                + "\n" + record.getText());
                        }
                    });

                } else if (p.toString().equalsIgnoreCase(
                    "unavailable")) {
                }

            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
};
muc.addParticipantListener(peopleListener);
Log.v(TAG, "join success");

```

```

    } catch (XMPPException e) {
        e.printStackTrace();
    }

    ChatPacketListener chatListener = new ChatPacketListener(muc);
    muc.addMessageListener(chatListener);
    updateListView(); // Update User list in the chat room
    loadVideo();
    return result;
}

private void loadVideo() {
    // TODO to be confirm
    if (LessonMain.live.equals("1")) { // Live
        videoview.setVideoURI(Uri.parse(url));
    } else { // Vod
        videoview.setVideoURI(Uri.parse(vodUrl));
        MediaController mediaController = new MediaController(getActivity());
        mediaController.setAnchorView(videoview);
    }
    videoview.requestFocus();
    videoview.start();
}

public Handler handler = new Handler() {
    @Override
    public void handleMessage(android.os.Message msg) {
        switch (msg.what) {
            case RECEIVE: {
                Bundle bd = msg.getData();
                String from = bd.getString("from");
                String body = bd.getString("body");
                receiveMsg(from, body);
            }
            break;
            default:
                break;
        }
    }
};

```

```

@Override
public void onClick(View view) {
    switch (view.getId()) {
        case R.id.lesson_main_left_send: {
            String msg = etMessage.getText().toString();
            if (!"".equals(msg)) {
                try {
                    muc.sendMessage(msg);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
            break;
        case R.id.lesson_main_left_toggleButton: {
            if (toggleClick % 2 == 0) {
                record.setVisibility(view.GONE);
                listview.setVisibility(view.VISIBLE);
                toggleClick = 0;
            } else {
                record.setVisibility(view.VISIBLE);
                listview.setVisibility(view.GONE);
            }
            toggleClick++;
        }
            break;
        default:
            break;
    }
    updateListView();
}

```

```

class ChatPacketListener implements PacketListener {
    private String _number;
    private Date _lastDate;
    private MultiUserChat _muc;
    private String _roomName;

    public ChatPacketListener(MultiUserChat muc) {
        _number = "0";
    }
}

```

```

        _lastDate = new Date(0);
        _muc = muc;
        _roomName = muc.getRoom();
    }

```

@Override

```

public void processPacket(Packet packet) {
    Message message = (Message) packet;
    String from = message.getFrom();

    if (message.getBody() != null) {
        DelayInformation inf = (DelayInformation) message.getExtension(
            "x", "jabber:x:delay");
        Date sentDate;
        if (inf != null) {
            sentDate = inf.getStamp();
        } else {
            sentDate = new Date();
        }

        Log.w(TAG,
            "Receive old message: date="
                + sentDate.toLocaleString() + " ; message="
                + message.getBody());

        android.os.Message msg = new android.os.Message();
        msg.what = RECEIVE;
        Bundle bd = new Bundle();
        bd.putString("from", from);
        bd.putString("body", message.getBody());
        msg.setData(bd);
        handler.sendMessage(msg);
    }
}

```

}

}

@Override

```

public void processMessage(Message message) {
    receiveMsg(message.getFrom(), message.getBody());
}

```

```

private void receiveMsg(String from, String msg) {
    Log.v(TAG, "receiveMsg():msg=" + msg);
    String[] fromString = from.split("/");
    if (fromString.length == 1) {
        // Filter out Useless Message
        // record.setText(record.getText() + "System Message" + " : " + msg
        // + "\n");
    } else if (!(msg.length() < 5)) {
        // Json Handler
        if (msg.substring(0, 5).equals("<ppt>")) { // Powerpoint
            try {
                LessonMainRightPowerpoint ppt = (LessonMainRightPowerpoint)
getFragmentManager()
                    .findFragmentById(R.id.LessonMainRightPowerpoint);
                ppt.handleJson(msg.substring(5));
            } catch (Exception e) {
                System.out
                    .println("PowerPoint cannot handle since it was not created");
            }

        } else if (msg.substring(0, 5).equals("<whb>")) {
            try {
                LessonMainRightWhiteBoard whb = (LessonMainRightWhiteBoard)
getFragmentManager()
                    .findFragmentById(R.id.LessonMainRightWhiteBoard);
                whb.loadJsonFromIM(msg.substring(5));
            } catch (Exception e) {
                System.out
                    .println("White Board cannot handle since it was not created");
            }

        } else {
            record.setText(fromString[1] + " : " + msg + "\n"
                + record.getText());
        }
    } else {
        record.setText(fromString[1] + " : " + msg + "\n"
            + record.getText());
    }
}
}

```

```

private void updateListView() {
    occupantsArray.clear();
    Iterator<String> occupants = muc.getOccupants();

    String name = "";
    System.out.println("O count" + muc.getOccupantsCount());
    while (occupants.hasNext()) {
        String tempName = occupants.next();
        System.out.println("occupants name : " + name);
        if (tempName.equals(Login.loginID)) {
            continue;
        }
        occupantsArray.add(tempName.substring(tempName.indexOf("/") + 1));
    }
    try {
        getActivity().runOnUiThread(new Runnable() {
            public void run() {
                buddylist.notifyDataSetChanged();
            }
        });
    } catch (NullPointerException e) {
        occupantsArray.add("No People.");
        // buddylist.notifyDataSetChanged();
    }
}

```

@Override

```

public void onItemClick(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
    LessonMainRightChat lessonMainRightGroupChat = (LessonMainRightChat)
getFragmentManager()
        .findFragmentById(R.id.LessonMainRightChat);
    lessonMainRightGroupChat.startByLessonMainLeft(occupantsArray.get(arg2),
        muc);
    FragmentManager fm = getFragmentManager();
    FragmentTransaction ft = getFragmentManager().beginTransaction();
    ft.setCustomAnimations(android.R.animator.fade_in,
        android.R.animator.fade_out);
    ft.show(fm.findFragmentById(R.id.LessonMainLeft));
    ft.hide(fm.findFragmentById(R.id.LessonMainRightAskQuestion));
}

```



```
        ft.show(fm.findFragmentById(R.id.LessonMainRightChat));  
        ft.hide(fm.findFragmentById(R.id.LessonMainRightWhiteBoard));  
        ft.hide(fm.findFragmentById(R.id.LessonMainRightPowerpoint));  
        ft.commit();  
    }
```

```
    public void onDestroy() {  
        muc.leave();  
        super.onDestroy();  
    }
```

```
}
```