|  |
| --- |
|  |
| Career Portfolio |
| ***Dongchao Feng*** |

|  |
| --- |
| 3/18/2014 |

Contents

[Resume 2](#_Toc386535636)

[Academic Certificates 4](#_Toc386535637)

[Reference Letter 6](#_Toc386535638)

[Reference List 7](#_Toc386535639)

[Sample of My Work 8](#_Toc386535640)

## Resume

**Career Objective**

A Software Program graduate with a keen interest in practical applications and information systems is seeking a career where a background in problem-solving would be advantageous.

**Qualifications**

* Strong understanding in Object-Oriented Analysis/ Design
* Easy to communicate with team mate
* Good experience in Java development and C# .NET Frame development
* Good experience in Android development and Web service development
* Good experience in learning new technologies
* High energy, flexible and achievement-oriented individual

**Technical Skills and Development Tools**

**Programming Languages**: C#, C, Java, HTML/XML, SQL

**Platforms/Operating Systems**: Linux, Windows 2008 Server, Microsoft's .NET, MS-DOS

**Programs**: MS Access, MS Office Suite, Crystal Reports, Photoshop CS, Flash CS, Netscape Navigator, MS Web Expression Web, MS Project, MS Visio, MS Visual Studio

**Hardware**: Experienced with installation of mother boards, SIMM chips, internal/external modems, NICs, SCSI and IDE hard disks, SCSI floppy drives, SCSI I/O ports, and various printer configurations.

**Database**: MySQL, ORACLE, Microsoft Access

**Education**

* **Diplomas: Software Engineering Technician**

**Centennial College** 2011 to 2013 941 Progress Ave., Toronto, ON, M1G 3T8

* **Diplomas: High School Diploma** 2010 to 2011

**Boren-Sino Canadian School**

No.65 ShuangLong Road, Jiangmen Guangdong P.R.C

Postcode: 529030

**Volunteer Experience:**

Boren-Sino Canadian School (2010 to 2011)

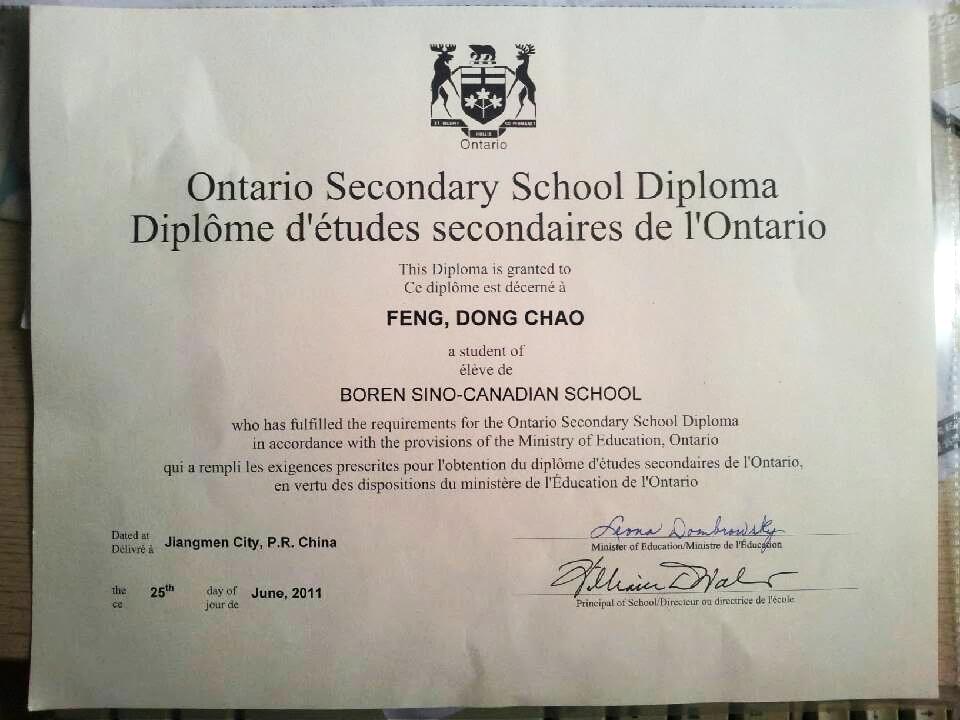
Library Server, Communicate with student and support the services. Learn the skill of communication and management.

**Languages:** English, Mandarin, Cantonese

**References: Available upon request.**

## Academic Certificates





## Reference Letter

# Reference List

Ilir Dema

Professor, SETAS, Centennial College

Email: [idema@my.centennialcollege.ca](mailto:idema@my.centennialcollege.ca)

Cell Phone: (647) 299 – 5349

Paula McMillan

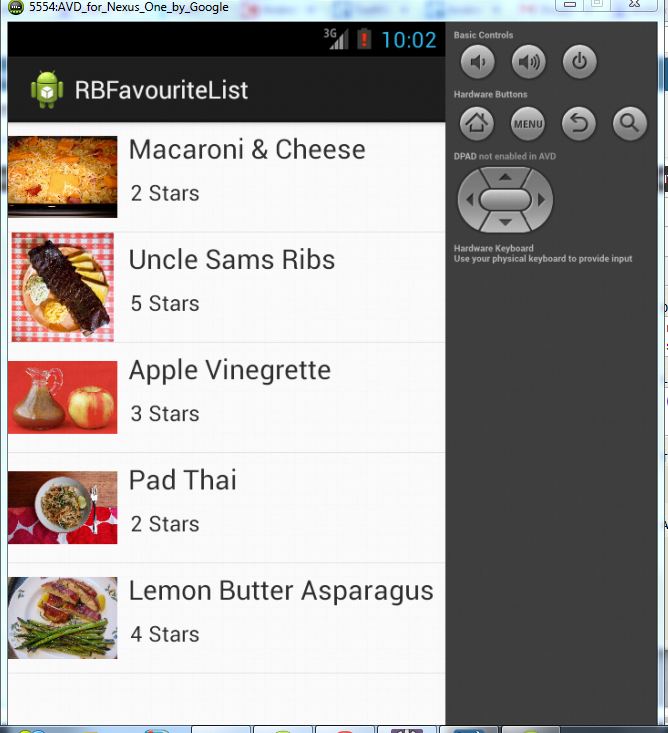
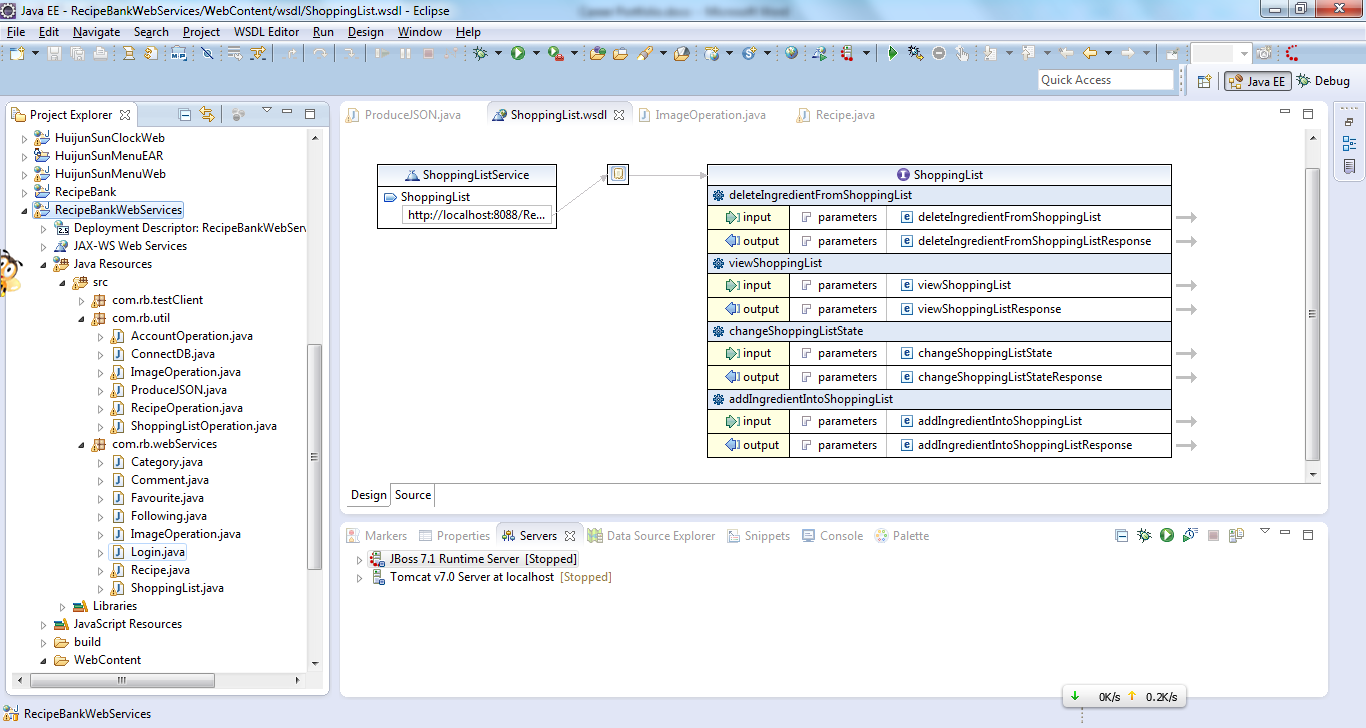
Professor, SETAS, Centennial College

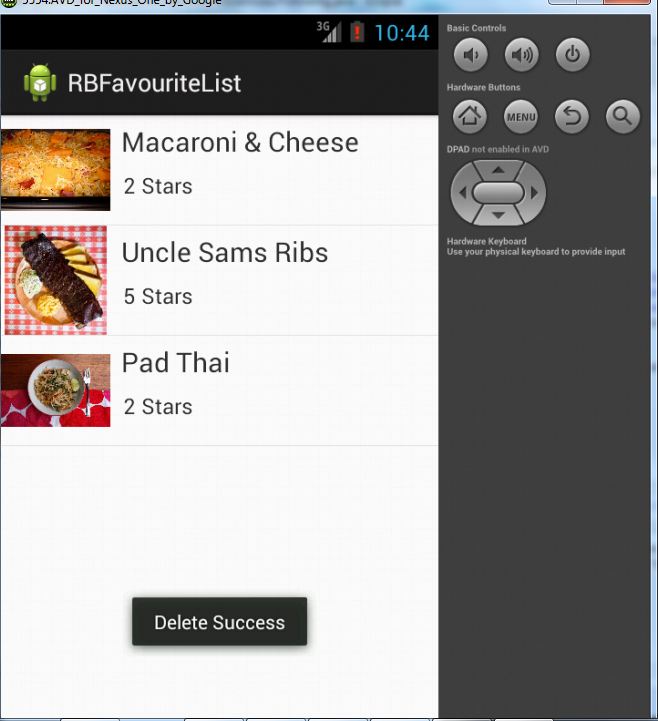
Email: [pmcmillan@centennialcollege.ca](mailto:pmcmillan@centennialcollege.ca)

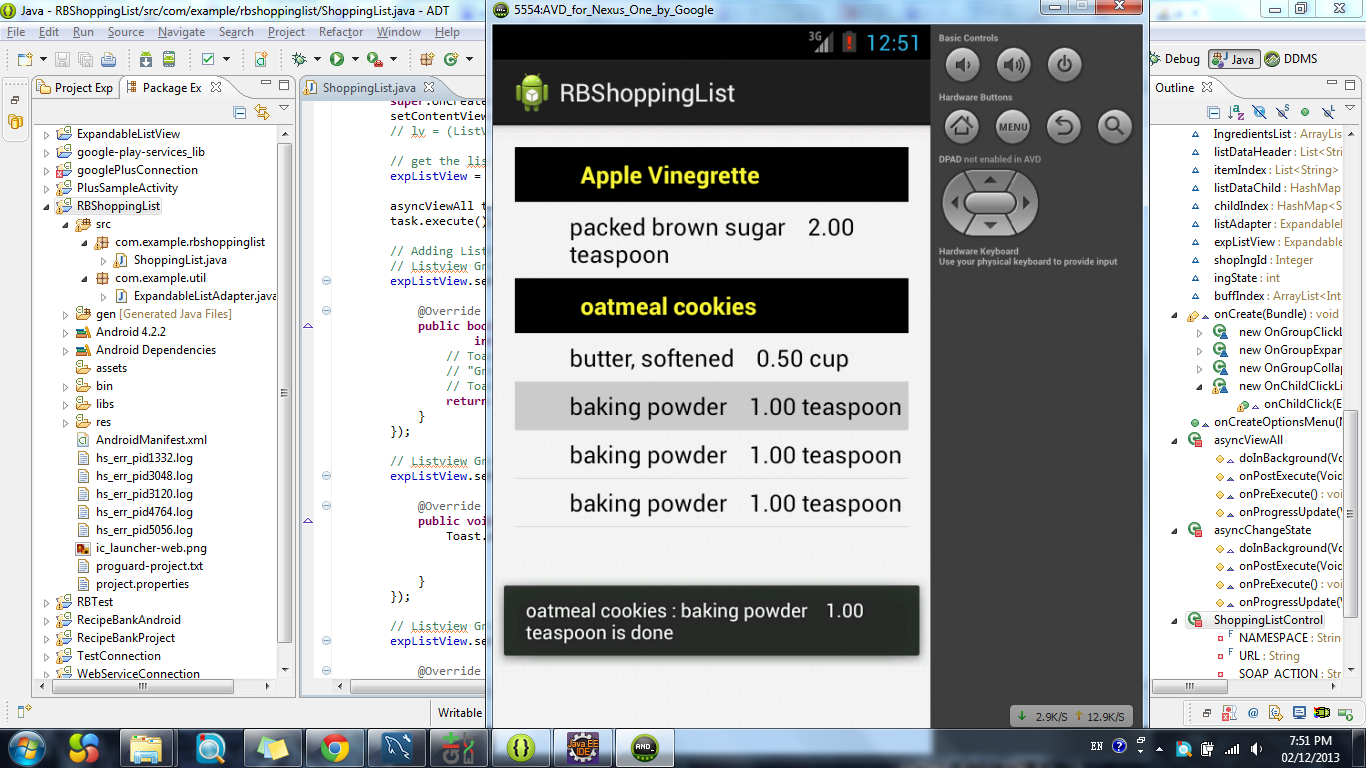
# Sample of My Work

Recipe Bank

* Java + wsdl + Json + SOAP + Android
* Group Project
* Using Json and KSoap to communicate between App and Web Service







Part of Code

* Image Operation.java

**package** com.rb.webServices;

**import** java.sql.Connection;

**import** java.sql.PreparedStatement;

**import** com.rb.util.ConnectDB;

**public** **class** ImageOperation {

**public** ImageOperation() {

// **TODO** Auto-generated constructor stub

}

**public** **int** insertImage(String bString)

{

**int** status=0;

Connection conn = **null**;

PreparedStatement statement = **null**;

System.*out*.println(bString);

**try** {

conn=ConnectDB.*getConnection*();

String sql = "update recipebank.recipe set photo=? where Recipeid=1";

statement = conn.prepareStatement(sql);

//pstmt.setBinaryStream(1, b, (int) file.length());

**byte**[] b=bString.getBytes();

statement.setBytes(1, b);

**if**(statement.executeUpdate()>0)

{

status=1;

}

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**finally**

{

ConnectDB.*closeConnection*(conn);

}

**return** status;

}

}

* ProduceJson.java

package com.rb.util;

import java.sql.ResultSet;

import java.sql.ResultSetMetaData;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Iterator;

import java.util.Set;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

public class ProduceJSON {

public static String resultSetToJsonArray(ResultSet rs) throws SQLException,

JSONException {

JSONArray array = new JSONArray();

ResultSetMetaData metaData = rs.getMetaData();

int columnCount = metaData.getColumnCount();

while (rs.next()) {

JSONObject jsonObj = new JSONObject();

for (int i = 1; i <= columnCount; i++) {

String columnName = metaData.getColumnLabel(i);

String value = rs.getString(columnName);

jsonObj.put(columnName, value);

}

array.put(jsonObj);

}

System.out.println(array.toString());

return array.toString();

}

public static ArrayList<HashMap<String, String>> parseJsonArrayToArrylist(

String jsonString) {

ArrayList<HashMap<String, String>> al = new ArrayList<HashMap<String, String>>();

try {

JSONArray jsonArray = new JSONArray(jsonString);

for (int i = 0; i < jsonArray.length(); i++) {

JSONObject jsonObject = jsonArray.getJSONObject(i);

String[] keysStrings = JSONObject.getNames(jsonObject);

System.out.println(jsonObject.toString());

HashMap<String, String> hMap = new HashMap<String, String>();

for (int j = 0; j < keysStrings.length; j++) {

hMap.put(keysStrings[j], jsonObject.get(keysStrings[j])

.toString());

}

al.add(hMap);

}

} catch (JSONException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return al;

}

public static HashMap<String, String> parseJsonObjectToHashMap(String jsonString) {

HashMap<String, String> hMap = new HashMap<String, String>();

try {

JSONObject jsonObject = new JSONObject(jsonString);

String[] keysStrings = JSONObject.getNames(jsonObject);

for (int i = 0; i < keysStrings.length; i++) {

hMap.put(keysStrings[i], jsonObject.get(keysStrings[i])

.toString());

}

} catch (JSONException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return hMap;

}

public static String arrayListToJsonArray(ArrayList<HashMap<String, String>> al) {

String jsonString = "";

JSONArray jsonArray = new JSONArray();

for (int i = 0; i < al.size(); i++) {

HashMap<String, String> hMap = al.get(i);

Set<String> keys = hMap.keySet();

Iterator iterator = hMap.keySet().iterator();

JSONObject jsonObject = new JSONObject();

while (iterator.hasNext()) {

String key = iterator.next().toString();

try {

jsonObject.put(key, hMap.get(key));

} catch (JSONException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

//System.out.println(key + ":" + hMap.get(key));

}

jsonArray.put(jsonObject);

}

jsonString=jsonArray.toString();

return jsonString;

}

public static String hashMapToJsonObject(HashMap<String, String> hashMap)

{

JSONObject jsonObject = new JSONObject(hashMap);

return jsonObject.toString();

}

}