Matthias MELLOULI

PROJECT REPORT

Android Application

Sendai college supervisor : Mr Takatoshi SUENAGA

IUT A supervisor : Mr Patrick Lebegue

Sendai national college of technology, from 30th of march to 23th of june
Acknowledgements

First, I would like to thanks Mr Suenaga, my tutor during this internship, who gave me this interesting project, and gave me his help, his time and the tools to make my project properly.

I also would like to thanks Mr Patrick Lebegue, my french tutor, because it is thanks to him that I had the possibility, to do my internship in Japan.

Next I would like to thanks Mr Ousaka, Mr Kaino, Mr Yajima, and all the staff from the SNCT, for their kindness and for their welcome.

Finally, I would like to thanks the students of my laboratory, especialy Takeasumi Sato, for their help, for their friendship and for helping me to discover The best parts of Japanese culture.
Abstract

I finished my studies at iut ‘a’ of Lille with an internship abroad, an internship with the aim to improve my technical abilities in computer science, English, but also in Japanese, because this course is held in Japan, a country that I love for many years.

My topic for this internship was to learn how to create an application in android, the Google operating system for mobile devices. The design of such an application is made in slightly modified Java. The purpose of the final application is to help students to speak easily with the Japanese. It is a system of databases that allows the user to have English phrases and their translations in Japanese. So I also used my knowledge of databases during my internship.

This internship allowed me to increase my knowledge in Java, a language with which I had many difficulties, but also to discover different aspects of Japanese culture and wonderful people who made this internship really enriching for me.
**Introduction**

As part of my DUT (equivalent to the OTC), in the 4th semester an internship is required. We can either do it in French company, or in a university abroad. This is the option I chose when the opportunity was given to me to be able to go to Japan, a country which I am passionate about for many years. So it is in the Sendai National College of Technology (SNCT) that I did my internship. I was under the tutelage of Mr. Suenaga, who offered me as project to do an Android application. Android is an operating system for mobile devices.

This project was conducted in two stages: the first is to familiarize myself with the development in Android (development tools, coding instructions, setting up the display). During this first step I tried to do a Task Killer, because there is not task manager under Android. For the second part of my project, I needed a topic that can be useful and accessible for me because of my low skills in Java. Mr. Suenaga proposed the idea of a mobile application to help foreign students to speak Japanese. The application is consist therefore in a database of translations of English / Japanese that the student can view and edit in order to have in any situations the chance to express themselves.

The problematic throughout this course will therefore: how to create an Android application that can be useful for foreign students?

I intend to improve the application once that my internship is ended, because there was lack of time to finish my application, and so the graphics are quite neglected, and I would like to add support for the camera in certain situations, but all this will be detailed in the section on my application.
# Contents

Acknowledgements.......................................................................................................................... 3  
Abstract.................................................................................................................................................. 4  
Introduction.............................................................................................................................................. 5  
1. Presentation of the work environment. ............................................................................................. 7  
   1.1 Japan.............................................................................................................................................. 7  
   1.2 SNCT.................................................................................................................................................. 8  
   1.3 Laboratory....................................................................................................................................... 9  
      1.3.1 Presentation................................................................................................................................. 9  
      1.3.2 Equipment available .................................................................................................................. 9  
2. My project: How to create Android application. ................................................................................. 10  
   2.1 Android........................................................................................................................................... 10  
      2.1.1 What is android? ......................................................................................................................... 10  
      2.1.2 Specifications: ............................................................................................................................ 10  
      2.1.3 Characteristic of the market: ....................................................................................................... 11  
      2.1.4 Why Android is better? ............................................................................................................... 13  
   2.2 First project: task killer .................................................................................................................. 14  
      2.2.1 The basics of creating applications ............................................................................................ 14  
      2.2.2 The project ................................................................................................................................ 17  
      2.2.3 Improvements ............................................................................................................................. 21  
   2.3 Second project: .................................................................................................................................. 22  
      2.3.1 Databases .................................................................................................................................... 22  
      2.3.2 The project .................................................................................................................................. 22  
      2.3.3 Improvements .............................................................................................................................. 33  
3. What did I learn? .................................................................................................................................... 34  
   3.1 Technically ....................................................................................................................................... 34  
   3.2 Humanly .......................................................................................................................................... 34  
Conclusion................................................................................................................................................ 34  
Appendixes............................................................................................................................................... 35
1. Presentation of the work environment.

1.1 Japan

Since the beginning of my studies, I wanted to do my internship abroad, but I didn't know if I could do it. Then I discovered that there was opportunity for me to go to Japan, so I did not hesitate, it is a country that I love for many years now. A country with a culture so special, bathed in a contradiction between high technology and persistent traditions.

Japan is an archipelago of 6,852 islands. The four largest islands are Honshū, Hokkaidō, Kyūshū and Shikoku, together accounting for 97% of Japan’s land area which is 377,944 square kilometers. The capital of Japan is Tokyo, which is considered as one of the three command center for the world economy, with New York and London.

Japan has the world’s tenth-largest population, estimated at around 127.3 million. Japanese society is linguistically and culturally homogeneous with small populations of foreign workers. In 2003, there were about 136,000 Western expatriates in Japan. Japan has one of the highest life expectancy rates in the world, at 81.25 years of age as of 2006. The Japanese population is rapidly aging, the effect of a post-war baby boom followed by a decrease in births in the latter part of the twentieth century. In 2004, about 19.5% of the population was over the age of 65.
Today, Japan is one of the leading nations in the fields of scientific research, particularly technology, machinery and biomedical research. Nearly 700,000 researchers share a US$130 billion research and development budget, the third largest in the world. Japan is a world leader in fundamental scientific research, having produced thirteen Nobel laureates in either physics, chemistry or medicine, three Fields medalists and one Gauss Prize laureate.

1.2 SNCT

With rapid progress and remarkable technological developments in Japan during the early 1960’s, every field of our society felt the need for trained technical experts. This led to the establishment of a new type of a higher educational institution, the college of technology. A college of technology admits students from junior high school, who then study one organized curriculum for five years.

The establishment that welcomed me for my placement is one of them. It is called SNCT that means Sendai National College of Technology. It educate junior high school graduates who will work as engineers. Students are here for at least five years, and can obtain an associate degree. They can also continue for two more years in advanced courses to have a bachelor’s degree. You can find the choices that a student has for being engineer in the appendixes. The college mottoes are spirit of independence, self-respect and exploration of possibilities.

There are several departments in this college which are : General Sciences, Information and Communication Engineering, Electronic Engineering, Electronic Control Engineering, Information
Engineering. The advanced courses are also divided in two categories which are Electronic Systems and Information Systems

1.3 Laboratory

1.3.1 Presentation

I am in the laboratory of Professor Suenaga. The laboratory is part of Electronic Control Engineering Department, but since this year, a new perspective of work was put in place, which is Department of Intelligent Systems and Electronics. The professor is currently working on human interface, and more particularly on mobile devices interfaces because it's more efficient and more convenient than big computers. That's why my subject for this placement was to study the way to create android application.

1.3.2 Equipment available

To do my internship in good conditions, I had at my disposal the tools necessary for my advancement. First I worked with my own laptop, which allowed me to work effectively immediately. Then the teacher Suenaga brought me a Mobile Development Platform, which serves to developers. This device can emulate any mobile OS, and has all the features of the latest mobile phones. This device is called ZOOM OMAP 2 and is manufactured by Texas. I didn't used a lot this device, because it was mostly for the camera, but I didn't work quickly enough to code the use of the camera.

Finally, the documentation I first I got a book called Android Application Development, but the problem with this book is that it is in Japanese. The only thing I could do was to look at the sample codes and try to understand. The second book is "Programming Bible for Android 2.1". There was two problems with this book. The first was that it was in japanese too, and the second problem was that it was as its name suggests focused on Android 2.1, whereas I was working on Android 1.5 so, many possibilities that Android 2.1 does offer could not assist me in Android 1.5. The last book is "Android Wireless Application Development". This book talked about wireless application but it had all the basics that I needed. The only problem with this book was that I received it in the last weeks.
2. My project: How to create Android application.

2.1 Android

2.1.1 What is android?

Android is basically an operating system for smartphones. But we find now integrated into PDAs, touch pads or televisions, even cars (trip computer) or netbooks. The OS was created by the start-up of the same name, which is owned by Google since 2005.

2.1.2 Specifications:

This operating system is based on version 2.6 of Linux, so it has a monolithic system kernel, what means that all system functions and drivers are grouped into one block of code.

- Architecture:

  Android consists of five layers:

  - The Linux kernel 2.6—which includes useful drivers that allow for example WiFi or Bluetooth.

  - The library written in C and C++ that provide higher level functionality such as an HTML engine, or a database (SQLite).

  - A runtime environment for applications based on a virtual machine, made for inefficient machines such as telephones. The aim is to translate JAVA in machine language understood by Android.

  - A JAVA framework that allows applications running on the virtual machine to organize and cooperate.
The user applications written in Java (Web browser, contact manager etc..)

• Current Version

Today android is in its 5th version, Android 2.1. Each version is designed to gradually correct the lack of APIs, to enhance the user interface and add functionality. The latest version adds such things as support in HTML5 in the browser, it allows multitouch or it brings new Contact API, which defines a database for contact management.

2.1.3 Characteristic of the market:

• Competitors

-The principal competitor is iPhone OS. It is mainly for competing with Apple that Android has been
created.

- Palm OS devices on PDA.
- Blackberry: which team the same name smartphones
- Windows Mobile: which team smartphones and PDAs.
- Sybian: Current Market Leader

• Key partners
  To help launch Android, Google has created an alliance of thirty companies in order to develop standards for mobile devices. There is, among others:
  - Operators such as NTT Dokomo, T-Mobile or Bouygues Telecom
  - Of-equipment manufacturers like Sony Ericsson or Samsug
  - Manufacturers of semiconductors, including Intel and Nvidia
  - Corporate businesses.

• market share

  The android market share continues to increase since its inception, and is likely to continue climbing because it is favored by big players like HTC, Sony Ericsson, Samsung, LG, Motorola, Dell, Acer. Moreover, according to IDC, android will be the 2nd mobile operating system used of the market in 2013. Here is the state of the market from 2006 to 2009. You have to know that the first mobile phone appeared in android date October 2008.
2.1.4. Why Android is better?

- **Applications**
  - google applications

  Android includes most of the time many Google applications like Gmail, YouTube or Maps. These applications are delivered with the machine most of the time, except in certain cases, such as some phones running android on which the provider has replaced Google applications by its own applications.

  - widgets

  With android, it is possible to use widgets which are small tools that can most often get information. These widgets are directly visible on the main window.

  - Android Market

  This is an online software store to buy applications. Developers who created applications can add them into the store, and these applications can be downloaded by users, they can be both free and paid.

- **Multitasking**

  Android allows multitasking in the sense that multiple applications can run simultaneously. With Task Manager it is possible view all running tasks and to switch from one to another easily.

- **SDK**

  A development kit has been put at disposal of everybody. Accordingly, any developer can create their own applications, or change the android platform. This kit contains a set of libraries, powerful tools for debugging and development, a phone emulator, thorough documentation, FAQs and tutorials.
• **Modifiability:**

This allows everyone to use, improve or transform the functions of Android for example transform the interface in function of uses, to transform the platform in a real system embedded Linux.

### 2.2 First projet : task killer

#### 2.2.1 The basics of creating applications

To begin to program for Android I needed some basics, because some elements are very different, even if programming an application in Android uses the Java language, therefore, an object oriented language. Firstly, in an Android application, there is no main method:

```java
public static void main(String[] args){...}
```

This method that allows to launch a program in java is not present in an application android. This example is only the first of a long list. So I'll try to explain what I had to learn to use to create my first application which is the Task Killer.

**Activity:**

An activity is a user interface that allows the user to interact with the screen, to perform actions. For example, a text messaging application could have an activity that displays a list of contacts to send messages. Once the contact is selected, activity could send information to a second activity that could serve to send the message to the contact.

When an application is launched, what it displays is the result of an activity. At the code level, for create an activity, you must create a class that extends the Activity class. An activity has a required `onCreate()` method. It is the main method. To interact with the program, through the activity, there must be something displayed, that is why the activity, contains what is called views.
**View:**

A View is the basic building block for user interface components. A View occupies a rectangular area on the screen. View is the base class for widgets, which are used to create interactive UI components (buttons, text fields, etc.). There are different kinds of views, for example a ListView is able to display only an interactive list of what you want to display, while a WebView allows you to display a web page. As said before, a View occupies a rectangular area on the screen. To organise these rectangles on the screen, there is a text file written in XML for every different screen.

**Xml:**

Xml means Extensible Markup Language. Android provides a straightforward XML vocabulary that corresponds to the View classes and subclasses. The goal of using Android’s XML vocabulary is to quickly design UI layouts and the screen elements they contain, in the same way that creating web pages in HTML:

with a series of nested elements.

Here is an example:

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView"/>
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button"/>
</LinearLayout>
```
In the previous example, here's an XML layout that uses a vertical LinearLayout to hold a TextView and a Button. It will be possible to modify attributes the elements in the activity class that use this XML file. For example to change the text.

**Intent:**

An activity can of course start another one, even if it but to do this, it will need a special object called Intent. An intent is basia description of an operation to be performed. It can launch an Activity, send a broadcastIntent to any interested BroadcastReceiver components, and communicate with a background Service. An Intent performs binding between the code in different applications. It can be thought of as the link between activities. It is possible to add some informations to an Intent, thanks to an object called bundle, that you add to the intent thanks to the method:

```
Intent.putExtras(Bundle objetbunble);
```

**Android Manifest:**

AndroidManifest.xml file is necessary for all android applications and must have this name in its root directory. In the manifest you can find essential informations about the application for the Android system, informations that the system must have before it can run any of the application's code. Here is what you can find in the Android manifest:

- The name of the Java package for the application. The package name serves as a unique identifier for the application.

- The description of the components of the application: the activities, services, broadcast receivers, and content providers that the application is composed of and under what conditions they can be launched.

- The processes that will host application components.

- The permissions the application must have in order to access protected parts of the API and interact with other applications.

- The permissions that others are required to have in order to interact with the application's
components.

- The list of the Instrumentation classes that provide profiling and other information as the application is running. These declarations are present in the manifest only while the application is being developed and tested; they're removed before the application is published.

- The minimum level of the Android API that the application requires.

- The list of the libraries that the application must be linked against.

With all these elements, an application can be created. So I'll explain my first application which was a Task Killer.

2.2.2 The project

For my first application, I was not really comfortable with android, so in this application there is only two different screens, and so two different activities, but it was a good training to try how to communicate between the activities, how to display elements (TextView, Buttons ...) on the screen and how to interact with them.
The first screen is drawn in a linear layout which is the simplest layout mechanism available on Android. You can add components horizontally or vertically. This is a pretty simple way of getting components assembled on a screen. You can find the XML code for this screen in the Appendixes. As you will see, there is one TextView, two buttons and again one TextView.

The two buttons have been declared in the XML file, but it is not enough for them for being active. I must declare them in the code:

```java
private Button okButton;
private Button quitButton;
```

Then you instantiate the buttons: they are linked with those of the xml file as follows:

```java
okButton = (Button) findViewById(R.id.oK);
quitButton = (Button) findViewById(R.id.quitter);
```

In this way it is now possible to interact with the buttons, thanks to the next method:

```java
Button.setOnClickListener(OnClickListener listener)
```

The aim of the 'OK' button is to access the process list, so to a new screen and thus a new activity: we will have a new Intent as explained previously. The action on the button code is as follow:

```java
okButton.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        try{
            Intent i = new Intent(TaskManager.this, ProcessList.class);
            startActivity(i);
            finish();
        }
        catch(Exception e)
```

18
To start an activity it is necessary to declare an intent and then implement it with what is called the context (current activity) and the name of the class that defines the next event (this one extends the Activity class). The next screen is a list of processes currently running on the device.

you must know that when the first of an application’s components needs to be run, Android starts a Linux process for it with a single thread of execution. By default, all components of the application run in that process.

To view this list several things are essential. Firstly, activity that allows the display of the list is a
ListActivity, which allows combined with the XML file, the display of a screen with a list. For the XML file it is simply a TextView. For the code itself there are several things to explain.

First, to retrieve information on the processes, you must declare the object ActivityManager. It must then instantiate it so that it retrieves the system datas, and especially datas about activities. Then, the goal is to recover all the processes running, so we need a list to store data of each process. This gives:

```java
android.app.ActivityManager activityManager;
activityManager = (ActivityManager) this.getSystemService(ACTIVITY_SERVICE);
List<RunningAppProcessInfo> procInfos;
procInfos = activityManager.getRunningAppProcesses();
```

To then display the names of the process (which will be strings), I created a list of strings, and I picked the name of each process. It only remains then to show it on screen, I needed for this a SetListAdapter, which converts the list into a viewable list. This gives:

```java
ArrayList<String> stringList = new ArrayList<String>();
for (int i = 0; i < procInfos.size(); i++)
{
    stringList.add(procInfos.get(i).processName);
}
setListAdapter(new ArrayAdapter<String>(this, R.layout.list, stringList));
```

The last step is to define the action to perform when the user clicks on a process. Here that will stop the process. In order to stop the process, we must define the permission in the AndroidManifest.xml.

```java
ListView lv = findViewById(R.id.listView);
lv.setTextFilterEnabled(true);
```

```java
lv.setOnItemClickListener(new OnItemClickListener() {
    public void onItemClick(AdapterView<?> parent, View view,
        int position, long id) {
        
```
android.os.Process.killProcess(procInfos.get(position).pid);
procInfos.clear();
procInfos = activityManager.getRunningAppProcesses();
if(!stringList2.isEmpty())
{
    stringList2.clear();
}
for (int i = 0 ; i< procInfos.size() ; i++)
{
    stringList2.add(procInfos.get(i).processName);
}

setListAdapter(new ArrayAdapter<String>(ProcessList.this, R.layout.
    list, stringList2));
ListView lv = getListView();
lv.setTextFilterEnabled(true);
}

Here what I do is that I conect the ListView with the XML file, and I kill the process in the given position of the list, and After i just display again the list.

2.2.3 Improvements

The improvements I could do would be first to add options to see this application such as the percentage of resources that each application takes, to see the relationship between processes. Finally a graphical look more presentable would have been preferable, but I had not the time to do it so I could not do better.
2.3 Second project :

2.3.1 Databases

For the second project, I needed to learn a new thing: how to use databases in Android. I can use databases for JEE, but in android is different. In fact there are ready-made functions to manipulate sqlite which is the database built into android. In this part of the report, I will not re-explain the mechanisms already explained that were resolved such as how to display a textview or a list.

To use a database, we will create a class called Helper. This class will allow us to manipulate the database from any other class that has instantiated the object Helper. This class has elements and methods very specific. First specific objects: a SQLiteDatabase, and a class called OpenHelper that we will also create.

```java
private SQLiteDatabase db;
OpenHelper openHelper = new OpenHelper(this.context);
this.db = openHelper.getWritableDatabase();
```

The class OpenHelper extends SQLiteOpenHelper. This class is used to actually create one table or several tables in a database, and fill the table in the moment of its creation, all this in the method onCreate(). This class allows to update the version of the table with the method onUpgrade(). The method onCreate will be called only once, after that the table is created this method will no longer serve. For the class Helper, you can add all the methods used to select, add, edit or delete entries in the table.

2.3.2 The project

The goal of the project is to help students to communicate with the Japanese people. For that there has 3 sections in the application:
For these three sections, I will need to use 3 tables. For each situation I'll use a table. The dictionary section, offers a table of English words, translated into Japanese. The table has three columns which are ID, word1, word2:
For this table, I made a class DictionaryHelper that works as explained above. In the method onCreate of the class OpenHelper I created the table and added all the entries. This happens in the following way:

```java
public void onCreate(SQLiteDatabase db) {
    db.execSQL("CREATE TABLE " + TABLE_NAME + " (id INTEGER PRIMARY KEY autoincrement, word1 TEXT, word2 TEXT)");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (word1,word2) VALUES('hello','kon nichi wa')");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (word1,word2) VALUES('spicy','karai')");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (word1,word2) VALUES('sweet','amai')");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (word1,word2) VALUES('bitter','nigai')");
    db.execSQL("INSERT INTO " + TABLE_NAME + " (word1,word2) VALUES('sour tart','suppai')");
    ...
    ...
}
```

The user can choose between the letters of the alphabet and when he press a letter, the onClick method of the button in question is called. In the case that the user selected the letter S, it will call the following method:

```java
sButton.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        try{
            List<String> names = Dictionary.this.dh.select("s");
            StringBuilder sb = new StringBuilder();
            ...
        }
    }
```

24
The class has an object DictionaryHelper (dh), which has a select method that takes as parameters the letter of the dictionary that the user want to see: here 's'.

```java
public List<String> select(String letter){
    List<String> list = new ArrayList<String>();
    Cursor cursor = this.db.query(TABLE_NAME, new String[] { "word1", "word2" },
        "word1 like "+letter.toLowerCase()+"'" or word1 like "+letter.toUpperCase()+"'" + null, null, "word1 asc");
    if (cursor.moveToFirst()) {
        do {
            list.add(cursor.getString(0)+" = "+cursor.getString(1));
        } while (cursor.moveToNext());
    } if (cursor != null && !cursor.isClosed()) {
        cursor.close();
    }
    return list;
}
```
The function query(TABLE_NAME, new String[] { "word1", "word2" }, "word1 like "+letter.toLowerCase()+"%" or word1 like "+letter.toUpperCase()+"%", null, null, null, "word1 asc") is equivalent as the sql request: "select word1, word2 in TABLE_NAME (here it's dictionary) where word1 like 's %' or word1 like 'S%' order by word1 asc". The select function returns all the translations in a list, shown on the screen as followed.
The situations section allows the user to be able to find translated sentences depending on the situation in which he is. For example, if he is in a store, he has a choice of phrases such as "how much does it cost?". At the interface level, this works in much the same way as the dictionary except that the table is different:

and that the select function now corresponds to "select word1, word2 in situations WHERE status = 'shop' (This Is An example) order by word1 asc; selection screen is different but the screen with the words is the same. The selection screen corresponds to the following:
Regarding the 3rd section, there is a table where the user can add, edit, delete, and of course consult his own translations. The table used has the same pattern as the dictionary table. The mechanism of the Helper class is always the same. The screen of choice in "Own translations" is the following:
If the user wants to see the translations, it'll be like the dictionary and for the request it's a simple "SELECT word1, word2 from translations"

Regarding the addition, the user is faced with this screen:
To add, the user must enter an English word and a Japanese translation in the corresponding fields. There is a test on the fields, so when the user let a field blank or let 'phrase 1' for the first field or 'phrase 2' for two fields, he must repeat the process and a message error appears:
If the fields are properly filled, the application execute the following sql: `insert into situations values ('<phrase 1>', '<phrase 2>');`. the translation is added to the table and the user is redirected to the main screen.

To change a translation, the user arrives on a list with all traductions. All he will have to do is click on the translation that has to be amended, he will be redirected to a modification screen similar to the addition screen, but with fields pre-filled with the words of the translation. He will only have to change them. Once done, he will be redirected to the screen of choices.
Finally, for the delete part, the user arrives on a list of translations, and pick the translation that he wants to delete. There is of course a confirmation screen to be sure that the user makes no error. If the user cancels, he returns to the list, if the query is executed and he is redirect to the translations menu. The sql request is: "delete from translations WHERE word1 = '<phrase 1>' and word2 = '<Phrase 2>'.

2.3.3 improvements

For this project, I had not time to do a better application. In fact it lacks first interface easier to use with anything other than simple lists and text boxes. The navigation could be improved, because right now it is only repetitive actions that the user has the choice to make. Finally, a part of the project was supposed to do but I had not time to do was to manage the camera to be able to, in the case of a situation, take a photo and use it to be understood. For example take the situation where the user is in a store, it could take a picture, choose the sentence that corresponds to his need, show the person to whom he wants to talk with, camera screen. There would be the photo and below by example the phrase "how much does it cost?" In Japanese.
3. What did I learn ?

3.1. Technicaly

First of all I gained additional skills in the Java programming language that was difficult for me before. I learned also how to use a lot of the components in eclipse such as the debugger. I also learned a new language that is XML, a language I didn't know at all earlier. Finally, this project allowed me to use my sql skills acquired during my studies.

3.2. Humanly

This placement and this project brought me a lot of things on a human level. First in relation to it myself. I had to acquire a large autonomy. Indeed I knew nothing about android before, and I had to learn how to use, and create things on android by myself. This brought me a lot. This autonomy is also linked to a research spirit. I had to try to find examples by myself in the book written in Japanese, try to test new things, for having nowhere to find answers. Finally I was able to acquire through this course and also thanks to the project a better english.

Conclusion

Thanks to this internship, which was also planned for this, I discovered a country, a culture, and beautiful people also learned new skills, and used all the things that I learned for two years. I do not regret anything in this experience and I thank all those who have allowed such a thing feasible. The Japanese will remain forever engraved in my memory and I hope from my heart to come back one day and discover still other aspects of this country so singular that is Japan. The people I met here are not just classmates, roommates and professors, they are friends, and once again I thank you all.
Appendixes

*Choices for students to become engineers*
Texas Instruments OMAP ZOOM2
Code XML for the first screen of the task killer

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout android:orientation="vertical"
    android:layout_width="fill_parent"
    <TextView
        android:id="@+id/text"
        android:layout_width="fill_parent"
        android:layout_height="280dp"
        android:gravity="center"
        android:text="@string/welcome_message"
        android:textStyle="bold"/>
    <Button
        android:id="@+id/ok"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
</LinearLayout>
```
android:layout_width="100dip"
android:layout_height="wrap_content"
android:layout_below="@id/text"
android:layout_gravity="center"
android:text="Enter"/>

(Button
android:id="@+id/quitter"
android:layout_width="100dip"
android:layout_height="wrap_content"
android:layout_below="@id/oK"
android:layout_gravity="center"
android:text="Quit")

<TextView
    android:id="@+id/text2"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_below="@id/oK"
    android:text="@string/rights"
    android:gravity="bottom"/>
</LinearLayout>