

# IM2073 Introduction to Design and Project

## Module IM203 Mobile Programming

### 1. Objective

This project module lets you “experience” mobile programming on Android platform.

### 2. Preparation

I STRONGLY encourage that you do the project using your own laptop. You MUST install the following software before the class:

- JDK (@ [http://www3.ntu.edu.sg/home/ehchua/programming/howto/JDK\\_HowTo.html](http://www3.ntu.edu.sg/home/ehchua/programming/howto/JDK_HowTo.html))

### 3. Schedule

You have four practical sessions (of 3 hours each) to complete the project.

**Session #1 and Session #2 (Individual or in pair if you cannot install Android Studio):**

- Do “How to Install Android SDK and Get Started” @ [http://www3.ntu.edu.sg/home/ehchua/programming/android/Android\\_HowTo.html](http://www3.ntu.edu.sg/home/ehchua/programming/android/Android_HowTo.html).
- Do “Android Basics and User Interfaces” @ [http://www3.ntu.edu.sg/home/ehchua/programming/android/Android\\_BasicsUI.html](http://www3.ntu.edu.sg/home/ehchua/programming/android/Android_BasicsUI.html).
- Do “Android Network Programming” @ [https://www.ntu.edu.sg/home/ehchua/programming/android/Android\\_Networking.html](https://www.ntu.edu.sg/home/ehchua/programming/android/Android_Networking.html).

**Session #3 and Session #4 (in groups of 2 persons): Mobile/Web Project**

In this project, we shall combine webapp and mobile app to develop a mobile “Clicker” system.

#### Use Case

1. Instructor shows a MCQ with options.
2. Students select an option from their mobile devices.
3. Instructor displays the statistics on his console.

#### **Hints:**

1. Refer to your “ebookshop” case study, in particular, the section on “Processing the Order”.
2. Create a MySQL table to capture the responses from the students.

Assume that there are many questions with any number of responses (a, b, c,...). Create two columns: questionNo, and choice.

Table: responses	
questionNo (int)	choice (VARCHAR(1))
...	...
...	...

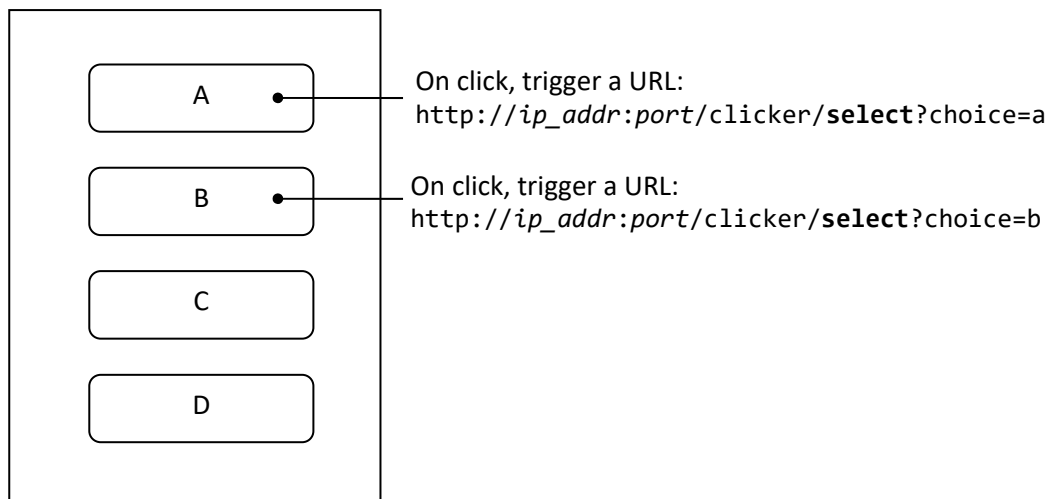
Whenever a response is received, you shall insert a new record into the table for that questionNo and choice, via SQL statement:

```
INSERT INTO responses (questionNo, choice) VALUES (8, '?');
```

You shall get the selected choice ('?') from the GET request parameter (to be explained later). Hardcode the questionNo (says 8), to begin with.

To count the number of 'b' response for a particular questionNo, use SQL statement:  
`SELECT count(*) FROM responses WHERE questionNo=8 AND choice='b';`

3. Under Tomcat, setup up a webapp called "clicker".
4. Write an Android app (as client) for the users:



Take note that each of buttons trigger the same URL, but differentiated by the GET request parameter "choice=x".

5. Write a servlet, which maps to URL `"/select"` (in `web.xml`), to capture the choice and store in the database table.

```
// Step 3 & 4 of the database servlet
// Assume that the URL is http://ip-addr:port/clicker/select?choice=x
// Assume that the questionNo is 8

String choice = request.getParameter("choice");
String sqlstr = "INSERT INTO responses (questionNo, choice) VALUES (8, '"
    + choice + "')";

int count = stmt.executeUpdate(sqlStr);    // run the SQL statement
```

#### Testing:

First, we shall test the servlet from a browser (before triggering it from an Android device). Issue URL `http://localhost:port/clicker/select?choice=a` from a *browser*. Check your MySQL database to ensure that a record inserted. Check the tomcat console for error messages, if needed.

Next, trigger the servlet from the Android emulator/device by pushing a button (the app that you wrote in Step 4), which issues URL `http://ip_addr:port/clicker/select?choice=a`. You cannot use `localhost` on Android emulator/device as it refers to the emulator/device. Find the IP address of your tomcat server via command "ipconfig" (Windows) or "ifconfig" (Mac OS X). Observe the response and check your MySQL database to ensure that a record inserted.

6. Provide a URL (e.g., `http://ip_addr:port/clicker/display`) for instructor to display the statistics. You need to write a servlet, which maps to URL `"/display"` (in `web.xml`), to query for the numbers of each response.

You could display the statistics in a table. But it would be nicer to show it in a bar chart (google "HTML bar char").

Try this query on your `mysql` client:

```
SELECT choice, COUNT(*) AS count FROM responses
WHERE questionNo=8 GROUP BY choice;
```

7. Of course, the instructor shall be able to show the question on a web browser, before the students click the choices. This completes the use case.
8. Extra Credits:
  - Create an activity for students to log-in to the system.
  - Keeping track of questions, e.g., in the MCQ panel, create two buttons "Start" and "Stop" and capture all responses between the start and stop times, for that particular question.
  - User (mobile phone) registration.
  - Students can also send additional comment.
  - Many others.