

Institute of Vocational Education
Department of Information and Communications Technology
HDSE (IT114105)
ITP4501 Programming Techniques for Mobile Systems
Summer Semester 2019-2020
Assignment

Submission Guidelines

- This is an individual assignment.
- The submission deadline of the assignment to is **11:55pm, 12 July 2020 (Sunday)**.
- You need to submit all program sources (in a single zip file) to the Moodle website <http://moodle.vtc.edu.hk> assignment dropbox before the deadline. You are advised to upload your work at a time reasonably earlier than the cut-off date and time. Moodle allows multiple submissions, however, only the latest copy will be retained. You will receive **NO MARKS for LATE SUBMISSION**.
- You are also required to give a demonstration. 40% of total marks will be deducted if demonstration is not done.
- If you do **NOT** meet 70% attendance requirement of IVE Higher Diploma Programme, your mark in this assignment will be **ZERO**!

1 Aims and Objectives

- To gain experience in mobile application UI and program design.
- To gain practical skill of Android application development.
- To understand the constraints and limitation of mobile application and the ways to overcome them.
- To obtain knowledge on connecting the mobile device to the internet services and building a multi-tier distributed system.

2 Introduction

In this assignment, you are required to develop an Android Application to play a **Numerical Reasoning IQ Test** by downloading the questions from a server and then request player to solve them. This app will also record the result and corresponding time required to complete a game and use charts to show the history records.

You can go to following web to try the Numerical Reasoning IQ Test:

<http://books.tamilcube.com/career/aptitude-test/numerical-reasoning/number-series/number-series-questions-001.aspx>

3 Functional Requirements

Listed below are the basic requirements of your application. You need to refer to the Local Database section for the database schema.

1. An activity which contain a button “Start”. When a user touches this button, your app will download ten IQ Test questions from a server and then ask user to **randomly answer five of them** one by one. **You are required to use radio buttons which randomly create three incorrect choices and are close to the value of correct one.**
2. Once a user answers a question, your app will show user’s answer is correct or wrong. You are not required to show the answer on the screen.
3. When a user finishes to answer all questions, your app will show following figures:
 - a. the time spent on the test,
 - b. the average time on each question and
 - c. the number of answered questions are correct.
4. A database which contains two tables. One table stores the questions, answers and your corresponding answers. Another table stores the date and time you play a test, corresponding duration to complete a test and how many answers questions are correct. (You are required to build and initialise the database on Android mobile phone.)
5. Two activities (one for **QuestionsLog**, one for **TestsLog**) properly show the data stored in the local database.
6. Get questions from the remote server and store it to the local database.
7. Display a bar chart to show the result on different tests (how many answers are correct on each test).

Note: You are encouraged to design and implement extra features. 10% of the total mark will be allocated on such additional functions. Refer to section 7 Marking Guidelines for more details.

4 Local Database

The database scheme described here is an extremely simple one. Many fields are intended not to be included in order to reduce the complexity of this assignment. You are free to add columns and tables to the database to fit for your own needs.

QuestionsLog (questionNo, question, yourAnswer, isCorrect(yes or no))

TestsLog (testNo, testDate, testTime, duration, correctCount)

5 Questions JSON Server

You can obtain the questions from the server:

<https://ajtdbwzbzh.execute-api.us-east-1.amazonaws.com/default/201920ITP4501Assignment>
and the data returned is in JSON format.

The sample JSON string returned is shown below:

```
{
  "questions": [
    {
      "question": "11, 13, 17, 19, 23, 29, 31, 37, 41, ? ",
      "answer": 43
    },
    {
      "question": "11, 10, ?, 100, 1001, 1000, 10001",
      "answer": 101
    },
    {
      "question": "20, 19, 17, ?, 10, 5",
      "answer": 14
    },
    {
      "question": "9, 12, 11, 14, 13, ?, 15",
      "answer": 16
    },
    {
      "question": "4, 6, 12, 14, 28, 30, ?",
      "answer": 60
    },
    {
      "question": "36, 34, 30, 28, 24, ?",
      "answer": 22
    },
    {
      "question": "1, 4, 27, 16, ?, 36, 343",
      "answer": 125
    },
    {
      "question": "6, 11, 21, 36, 56, ? ",
      "answer": 81
    },
    {
      "question": "2, 3, 5, 7, 11, ?, 17",
      "answer": 13
    },
    {
      "question": "2, 7, 14, 23, ?, 47",
      "answer": 34
    }
  ]
}
```

Note: The JSON string returned is not formatted. You may use the online JSON formatter tool such as <http://json.parser.online.fr/> or <http://jsonformatter.curiousconcept.com/> to read it.

6 Additional Constraints

- The UI of the mobile application must be produced with Android widgets such as **TextView**, **CheckBox**, and **Spinner** etc. Web-based UI is **NOT** allowed.
- The statistical charts must be produced using Android built-in graphics API such as **drawRect()** and **drawText()**. Using any other external drawing packages or libraries is **NOT** allowed.

7 Marking Guidelines

Your project will be assessed according to the items below.

- Database initialisation
- Level of completion
- Correctness
- UI design
- Program design and implementation
- Program style and comments

10% of marks will be allocated to extra features not described in section 3. Each student can develop at most 3 additional functions such as animation effect or sound effect on the Android device or any other relevant and useful functions.

40% of total marks **will be deducted** if demonstration is not done.

END