

ITIS/ITCS 4180/5180 Mobile Application Development
Homework 3

Basic Instructions:

1. In every file submitted you MUST place the following comments:
 - a. Assignment #.
 - b. File Name.
 - c. Full name of all students in your group.
2. Each group should submit only one assignment. Only the group leader is supposed to submit the assignment on behalf of all the other group members.
3. Your assignment will be graded for functional requirements and efficiency of your submitted solution. You will lose points if your code is not efficient, does unnecessary processing or blocks the UI thread.
4. Please download the support files provided with this assignment and use them when implementing your project.
5. Export your Android project and create a zip file which includes all the project folder and any required libraries.
6. Submission details: The file name should follow the following format:
Group#_HW03.zip
7. **Failure to follow the above instructions will result in point deductions.**

Homework 3 (100 Points)

In this assignment you will develop “How Geeky Are You?” personality test. It is Trivia Quiz where you will be given a set of questions and based on the answers that you provide, the quiz will show your “geekiness” level. All support files used in this app will be provided.

Part A: Splash Screen Activity (15 Points)

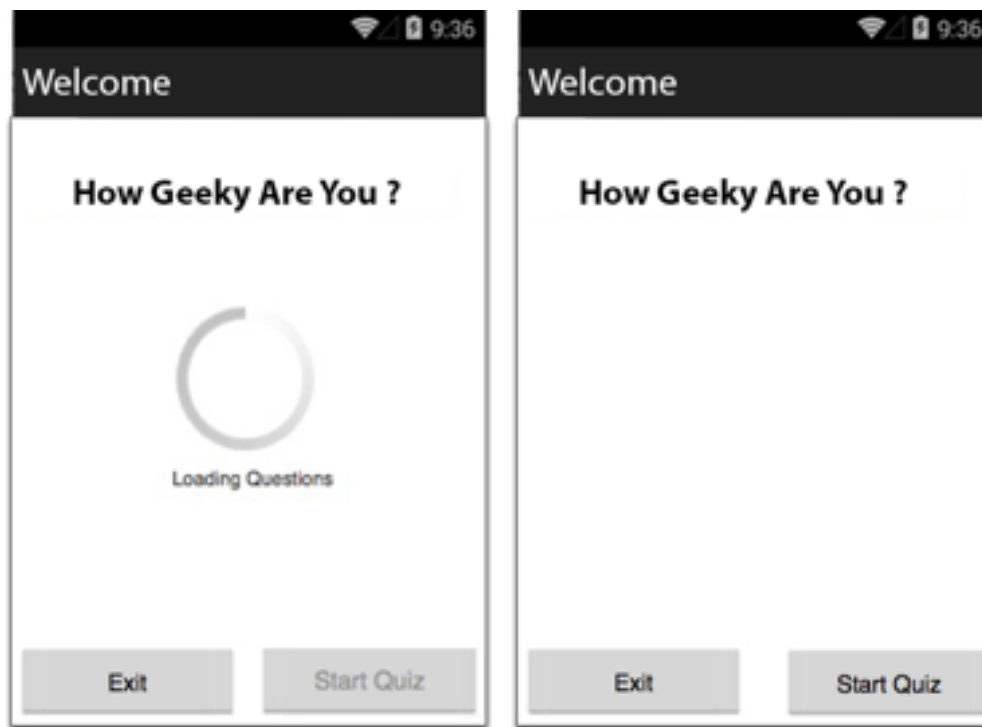
Figure 1 shows the launcher activity (Splash), it contains:

1. Image (**splash-screen.jpg**)
2. The screen automatically navigates to next activity (Welcome) after 8 seconds unless user taps (Start Quiz) Button
3. Button Start Quiz: Tapping this button finishes the current activity (Splash) & start the next activity (Welcome).



Figure 1, Splash Activity

Part B: Welcome Activity: Loading and Parsing Questions (35 Points)



(a) Downloading the questions

(b) Finished downloading the questions

Figure 2, Welcome Activity

The questions and answers for this app can be retrieved from the URL:

http://dev.theappsdr.com/apis/spring_2016/hw3/index.php?qid=<id>

The URL requires one parameter, **qid**. There are 7 questions with IDs from 0-6. For example to retrieve the **4th** question, you need to use this URL:

http://dev.theappsdr.com/apis/spring_2016/hw3/index.php?qid=3

Each question is formatted in a separate line with entries separated by the semicolon “;”. An example question is listed below:

```
3;Its time to upgrade your mobile phone and you have the
budget for it. How do you choose the make and model?;Get
whatever the sales person says is best for my needs;0;Get a
model identical or similar to that of my friends;2;Ask a
friend or read reviews online;5;I already know what mobile
I want as I follow mobile industry updates;10;http://
dev.theappsdr.com/apis/spring\_2016/hw3/photos/3.jpg
```

Which follows the following syntax:

```
<ID>;<Question Text>;<Option Text>;<Option value>;  
...;<Question Image URL>
```

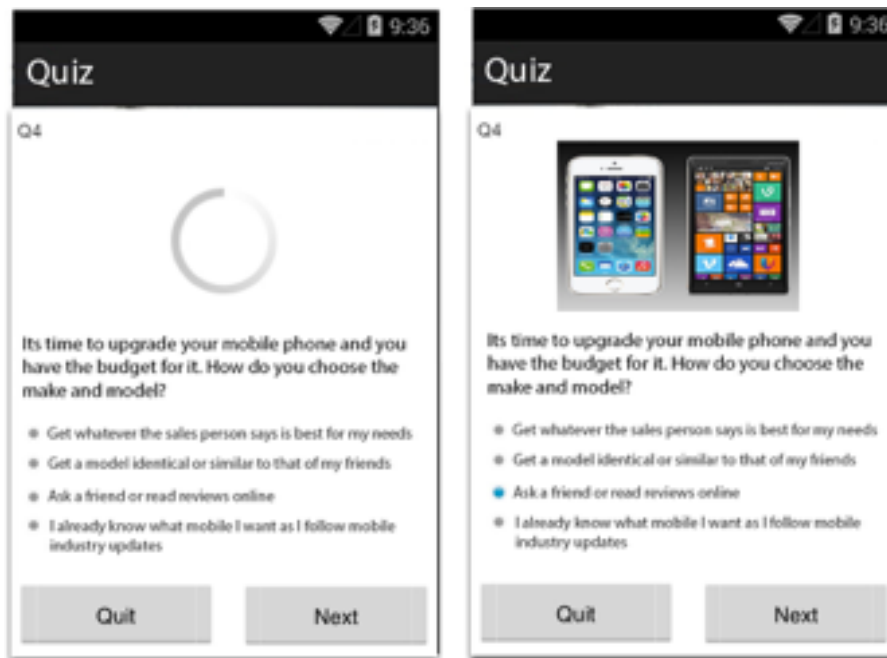
Key:Can be repeated 1 or more times,Optional

Values after option text represent the amount that will be added to the score if the user selects that option. For example, here the third option is “Ask a friend or read reviews online” and the next value after the delimiter is “5”. So 5 will be added to the score if this option is selected.

Activity requirements are :

1. The Welcome Activity should use an AsyncTask or Thread to retrieve the questions and answers from the URL provided above.
2. While the questions are being retrieved the Welcome Activity should display a progress dialogue as indicated in Figure 2(a)
3. User should not be able to start quiz before the quiz retrieval is finished. Thus, the “Start Quiz” button should be disabled while the questions are being retrieved and enabled when done retrieving.
4. You should create a Question Class that holds question attributes such as, question text, options and their associated scores, image URL and any other needed information.
5. The AsyncTask or Thread should return a List of Question objects. Make sure the Question class should implement the Parcelable interface. **Note:** Do not pre-download any of the images and do not store the images bitmap in the Question Class.
6. Figure 2(b) shows the activity after the loading and parsing are completed. The “Start Quiz” button is enabled, the progress dialogue is dismissed indicating that app is ready to view quiz questions.
7. Tapping the “Start Quiz” button should do the following:
 - (a) Pass Question list to next activity (Quiz)
 - (b) Finish current activity
 - (c) Start the Quiz Activity
 - (d) Tapping the “Exit” button should exit the application

Part C: Quiz Activity (40 Points)



(a) While loading the image

(b) After Loading the image

Figure 3, Quiz Activity Wireframe

The Quiz activity should receive the list of questions from the Welcome activity. Figures 3(a) and 3(b), show the wireframe of the Quiz activity. The activity shows the question number, question text, Image and the set of answer options. This activity is used display quiz questions. UI elements and Layout should be properly updated to display the current question. The requirements are as follows:

- Options should not keep the original order as parsed, they should be shuffled randomly when populated to the user. Note also that the number of choices for each question varies, so the views representing the choices should be handled dynamically.
- In regard to question image:
 - If there is no image, the url entry will be left as "" to indicate no image. Only show the questions text and options.
 - If the current question has an image, then the image should be downloaded using the corresponding image URL indicated in the question. Downloading the image should be done using a separate thread (or AsyncTask), not using the main thread.
- Your activity should ensure that the downloaded image displayed corresponds to the currently question and not when other questions are displayed. **Hint: use setTag()**
- While the question image is loading you should display a progress dialogue indicating the image is loading, as shown in Figure 3(a).

5. When the user answers a question, the accumulative score should be maintained to reflect user selection. So if the user selects the 3rd option as shown in Figure 3(b), he gets 5 points for that particular question).
6. Users cannot skip to the next question without answering. If the user clicks on next button without selecting an option, an error Toast message should be displayed indicating "Please select an option..".
7. If the user clicks the "Quit" button the activity is finished and the user is sent back to the Welcome activity.
8. Upon answering the last question, the Quiz activity should start the Result activity and send it the accumulative score.

Part D: Result Activity (10 Points)

Score	Result	Description
0-10	Non-Geek	There isn't a single geeky bone in your body. You prefer to party rather than study, and have someone else fix your computer, if need be. You're just too cool for this. You probably don't even wear glasses!
11-50	Semi-Geek	Maybe you're just influenced by the trend, or maybe you just got it all perfectly balanced. You have some geeky traits, but they aren't as "hardcore" and they don't take over your life. You like some geeky things, but aren't nearly as obsessive about them as the uber-geeks. You actually get to enjoy both worlds
51-72	Uber-Geek	You are the geek supreme! You are likely to be interested in technology, science, gaming and geeky media such as Sci-Fi and fantasy. All the mean kids that used to laugh at you in high school are now begging you for a job. Be proud of your geeky nature, for geeks shall inherit the Earth!

Table1: Results



Figure 4, Results Activity

Figure 4 shows the wireframe for the Result activity. This activity shows the user the the result according to value of the accumulative score of his answers in the previous activity. The result should be in the range of 0 – 72. This range is divided as shown in Table1 into 3 categories, check the rage of the score and display the result accordingly. The Activity contains:

1. TextView indicating the Geekiness level

2. Geekiness level image. You will be provided with corresponding images in SupportFiles.
3. TextView showing a description of the geekiness level(See Table1).
4. Clicking the “Quit” button should finish current activity and send the user to the Welcome activity.
5. Clicking the “Try Again” button should finish the current activity and send the user back to the Quiz activity to Re-take the quiz.