

Baze University
COM313 Application Programming with Java and C#
Level 300 May 2020
Examination Paper

In addition to this paper, you will find an archive (WinRar) file containing an **ExamStarterKit** folder.

When you receive the Exam Starter Kit, expand the folder “**ExamStarterKit**” into the class folder **COM313_AppProgramming2020Q1**. Looking inside the **ExamStarterKit** folder, you should see the following folders: **AppTesterDB2020Q1**

This app serves as
the single case
study for this exam.

Before you start **the exam** CLOSE ALL
open IDEs (Android Studio,
Visual Studio 2019, IntelliJ). Also
close ANY open Emulators.

READ BEFORE YOU START

With so many graduates in Nigeria, getting hired to work in technology might require that you take an **online** entry exam using your computer from your home.


This exam is usually constructed to weed out B.Sc. in Computer Science (IT, CS, SE) graduates who cannot code or learned programming on paper or crammed to pass the exams. The exam is usually designed to test your competency in reasoning, problem solving, basic Java, C++, C#, et cetera, Object Oriented Programming and simple application of this competency.

This exam attempts to simulate a pre-hire **online** entry exam. The key to this exam is **(1)** Attention to details **(2)** Reasoning and problem solving **(3)** Providing as correct an answer as you can **(4)** The recollection/application of what you studied in the different programming modules that you have taken.

The App is Case Studies for All the questions

Answer all 4 questions.
Total Score 50

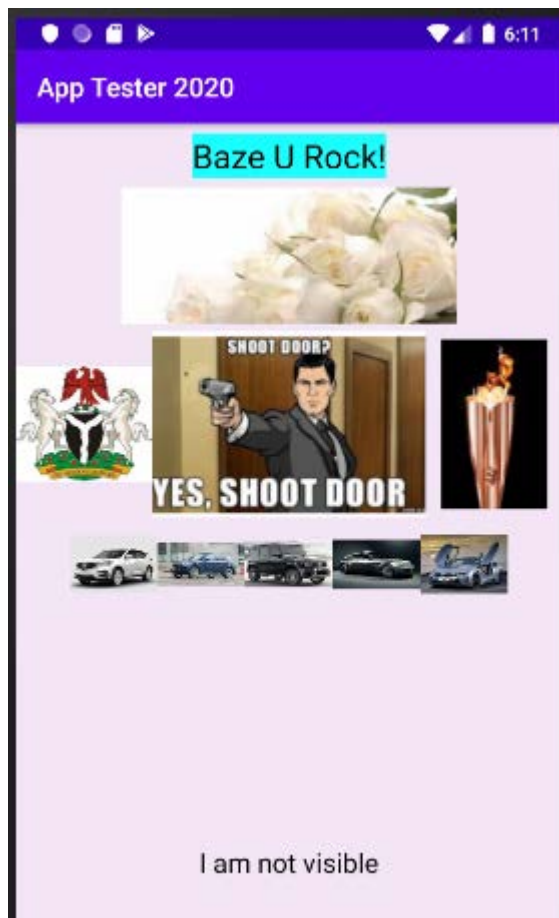
Using Android Studio 3.6.1 and above, navigate to open the

 AppTesterDB2020Q1 app

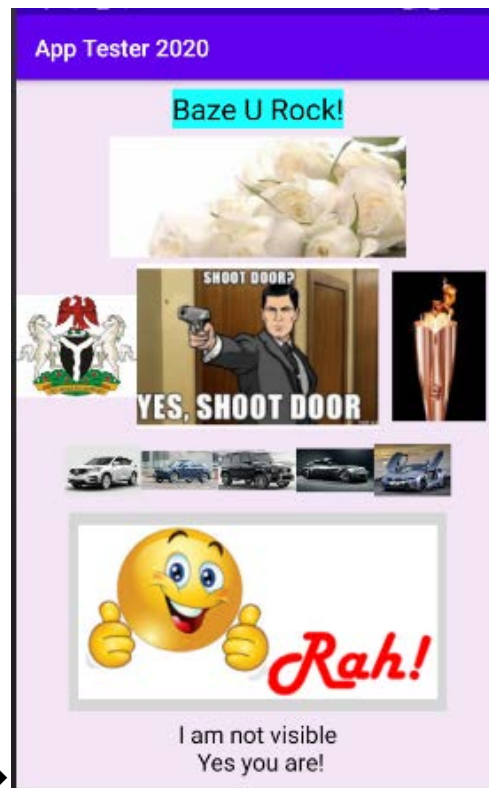
To stabilize the app for your environment, invalidate cache and the rebuild the app. You can see how to accomplish this on page 3 of the document: Week 11 (Online Question and Answer)-V2 – also in the archive containing the ExamStarterKit.

Now run the app and let us step through the sequence and activities or expectations in the app. It is important that you pay **attention to the sequence and flow of events as discussed below**:

When the app is launched in the emulator, we should see the Main activity UI:



Clicking on shoot the door, display the Rah button and clicking on the Rah button navigates to the HelloMe UI.



Note #1

- The expectation is that a tool bar will be added at the top of the main page that will enable the user to login by navigating to and from the Login UI.
- If the user clicks on any of the cars, a pop-up UI will be used to display the image of the car that was clicked - with the image name (as coded in the drawable folder) displayed below the Image.

Now then clicking on the Rah button navigates to the HelloMe UI.

On display of the **Hello Me** UI, if user had previously logged in, his or her info is displayed from the database.

Otherwise the user must 1st login and on return the info will be displayed.

The user can change some of the displayed info and use the Update button to write the update to the database

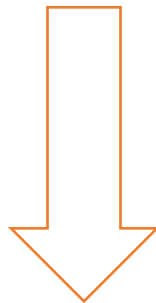
CRUD

Clicking the Login button navigates to the Login UI

A **ME** button here will navigate to **Me Things** UI. The **Me Things** UI enables the user to add, edit, delete and display "stuff" using a RecyclerView with database I/O

CRUD

On Navigation to the Login UI, we see:



Login

ENTER

☐ Show password

LOGIN

Register FEEDBACK

Login

ENTER

Person Email

Password

☐ Show password

LOGIN

Register FEEDBACK

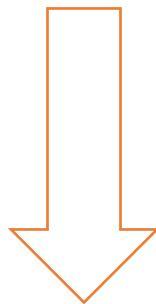
While coding the credentials to login, the user can use the Show password checkbox to uncloak the password. If the LOGIN fails, a message directs the user to verify credentials & try again or register

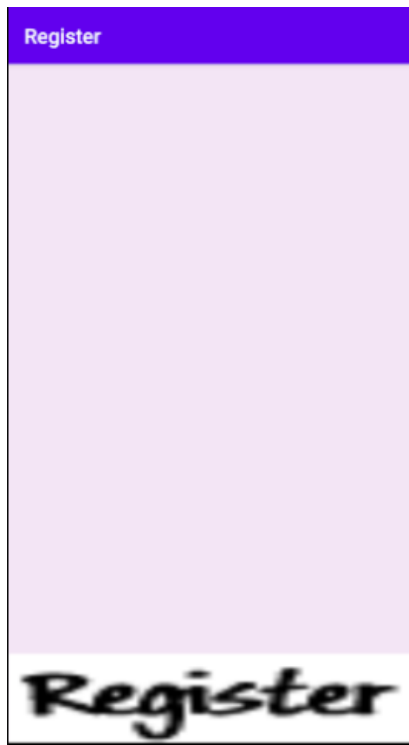
CRUD

Click ENTER button to display the login credentials EditView controls

The funny looking button is for future enhancement on login options. Feedback takes the user to/from the UI for feedback and comment. User must be logged in to leave feedback. Clicking the Register button, navigates the user to the Register UI

On Navigation to the Register UI, we see:





Basic Register UI user entry fields.

The Register enables the user to enter login credentials as well as some of the data displayed on the Hello Me UI – such as 14 digit phone number, a check box as to whether the phone number should be used for WhatsApp.

On completion the user is navigated back to the Login UI to complete the login process.

It is your responsibility to unit test after any change you make. Not doing so might prevent you from completing question #2 which has 15 points.

Question #1 [15 points]

- A. Design the Register UI to show the Basic Register UI user entry fields and enable Register button to navigate to an empty method named: `processRegisterButtonClick ()`.
- B. In the Login.java activity file, add a public static string variable named `loginUser`. At the bottom of the `Oncreate ()` method initialize this variable to “Not Logged In”. In the empty `validateUserCredential()` method, set the value in the variable to your name.

Now implement the following method at the bottom of the MainActivity.java class file.

```

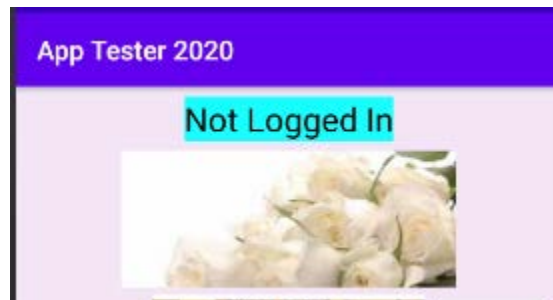
protected void onResume()
{
    super.onResume();
    //
    checkIfUserIsLoggedIn();
    //
    Toast.makeText(MainActivity.this,
        "From Hello_Me or Login, ON RESUME",
        Toast.LENGTH_LONG).show();
}

private void checkIfUserIsLoggedIn()
{
    //Address Baze u rock TextView control @id/myNameIs
    TextView myNameIs = findViewById(R.id.myNameIs);
    //retrieve loggedIn user name from Login.java class field
    if (Login.loginUser != null && Login.loginUser.length() > 0)
    {
        myNameIs.setText(Login.loginUser);
    }
}

```



Start the app, navigate to the Login UI. Click the Login button and then use the hardware back button to return to the main page. Your name should now show instead of:




- C. Add a Feedback activity and enable the FEEDBACK button to navigate to the Feedback UI. Change the title of the UI page accordingly. Change the background to be same as the Login UI background

- D. Add the MeThings Activity and change the title of the related UI to “Me Things”. Add the **ME** button on the HelloMe UI and enable navigation to the MeThings UI. Change the background color to be same as HelloMe. Add a button with text “Add Me Things” at the bottom of the Me Things UI bounded by left of parent, bottom of parent, right of parent and horizontal guideline as H.93

[15 marks]

End Question #1

Question #2 [15 Points]

In the ExamStarterKit, is a folder named:  **Popups** . Using the content of folder as reference or any other source, implement a Popup UI that is triggered when a user clicks any of the 5 cars shown on the main page. The popup on activation should display the image of the clicked car using a size like the Rah ImageView. Also the name of the car should be displayed at the bottom of the popup using a TextView – where the name of the car is retrieved (using code) from the ImageView attribute `android:src="@drawable/acura_rdx_2019"` used in coding the name of image to be displayed.


Critical Notes

- The reference folder shows how to implement a popup using a UI design technology known as LinearLayout or RelativeLayout. If you choose to use this to implement your popup, **the maximum score that you can get is 10 points.**
- If you can use ConstraintLayout UI design technology, as is used in all the UIs in the app, to implement your popup, **the maximum score that you can get is 15 points.**

[15 Marks]

End Question #2

Question #3 **[10 Points]**

In the ExamStarterKit, is a Visio file named:  Question#2-Fatima-Jaguar-Nana-1760 .
Change the name “Fatima-Jaguar-Nana” to your name and the number 1760 to the last 4 digits of your matric number.

Using the Visio file, create an Application Architecture that correctly depicts the activities (both active and planned) as well as database access highlighted in Question #1 and the UIs screen shots shown in the case study.


Note:

- This question must be answered in the file provided, to assure that your solution is submitted together when returning the ExamStarterKit.

[10 Marks]

End Question #3

Question #4 **[10 Points]**

In the ExamStarterKit, is an Erwin file named:  Question#3[ERD]-Fatima-Jaguar-Nana-1760 .
Change the name “Fatima-Jaguar-Nana” to your name and the number 1760 to the last 4 digits of your matric number.

Using this file, design the database model that you feel might be used to adequately support the activities (both active and planned) as well as database access highlighted in Question #1 and the UIs screen shots shown in the case study.

Your ERD should focus on the Entities, primary keys and relationships (with appropriate cardinalities) as well as the key attributes for each entity.

Note:

- This question must be answered in the file provided, to assure that your solution is submitted together when returning the **ExamStarterKit**.
- The Me Things UI will require the use of categories to record the stuff of interest such as cars, music, friends, gossip tidbits, fashion, sports, food, health and wellbeing, movies, TV, et cetera. The Me Things UI will also enable user to be able send delayed SMS and email to be automatically launched at a future date and time.

[10 Marks]

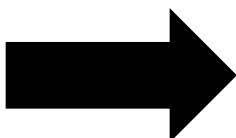
End Question #4

SUBMISSION OF THE EXAMINATION

When done:

- Compress the ExamStarterKit**
- Copy the Compressed file to OneDrive folder
- In OneDrive, **test** the archive by clicking to be sure that the archive opens
- Generate a OneDrive link.
- Email the OneDrive link (**using your Baze email**) to eu.kvac@bazeUniversity.edu.ng **on or before** 6p.m. Sunday May 10, 2020.

Note: This is an exam, hence submission after the dateline will not be accepted for whatever reason. Start early to package and submit if you suspect (or you usually have) network problems. You will not be called or contacted if your archive is faulty. The university has set a dateline for processing the exams. You are being allowed to submit on April 10 because this exam is usually held on the last day of the exam or the day after.



END OF EXAM

