|  |  |  |  |
| --- | --- | --- | --- |
| Student Information | | | |
| **Student ID:** | 100576422 | **Student name:** | Timothy Owens |
| **Unit code** | ICTPRG601  ICTPRG532 | **Unit title** | Apply advanced object-oriented language skills  Develop advanced mobile multi-touch applications |

**Part 1. Task Instructions**

In this task, you will be presented with a graphic design of an Android mobile app, and detailed software design specification document (Part 2 in this document). Your job is to develop and publish the Android mobile app that meet all the design requirements. Also, you are required to answer all the questions listed in Part 3 in this document.

The whole software development process can be break down into the following phases

1. Plan & Preparation
2. Client-side Android app user interface design
3. Client-side Local database design and connect to the UI
4. Server-side database design
5. Server-side Web API design
6. Connect front-end Android app to back-end web APIs
7. Software test and performance analysis
8. Documentation and quality control
9. Publish mobile app and handover project

P1

P2

P4

P3

P5

P6

P7

P9

BEGIN

END

P8

You are required to follow the same order as the above 9 phases to complete developing the software. Part 3 questions are also designed following the same logic. So, for each phase of the development, you should walkthrough and answer the corresponding questions as you work on programming. After you finished one phase of programming and questions, you are required to present your work to the assessor; you assessor will also confirm if you can move on to the next phase by signing off the section.

Your assessor also plays the role of the client. Speak with your assessor for requirement clarification if needed.

For final submission, you are required to submit the following:

|  |  |
| --- | --- |
| 1. A signed assessment document |  |
| 1. This instruction document with all questions answered and signed off by assessor for each phase. |  |
| 1. Android app project (source code) |  |
| 1. Server-side project (source code) |  |
| 1. Client-side database snapshots |  |
| 1. Server-side database snapshots |  |
| 1. Web API design document |  |
| 1. User document (html file) |  |
| 1. All test plan documents |  |
| 1. All test report documents |  |
| 1. Performance analysis report |  |
| 1. Application files for deployment (e.g. Android APK) |  |
| 1. Inter device communication program project (if programmed separately) |  |

**Part 2. Software Design Specification**

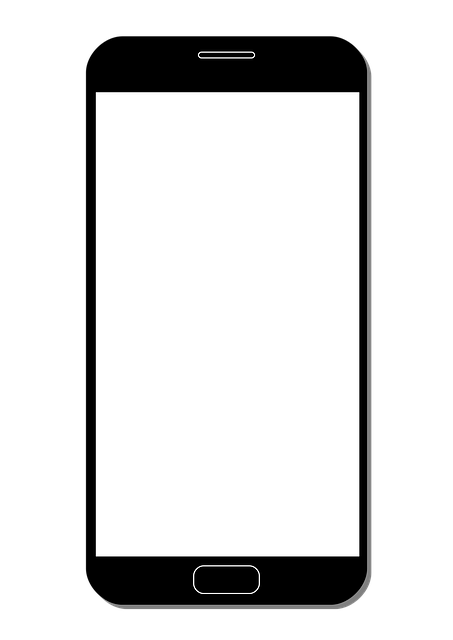
You are required to design an Android mobile app that meet the requirements given by graphic design and this specification. Graphic design resources are provided separately; requirements from the software perspective are explained in this document. Consult the client for further clarification as required.

1. **Overview of the software**

The application provides an interface to perform CRUD functionalities for data stored in databases, on both client-side and server-side. The client-side application is an Android mobile App with local DB (e.g. SQLite), the server-side service is HTTP-based RESTful API with remote DB (e.g. MySql). The overall design is illustrated below.

**Server-side**

**Client-side**



Android App

SQLite DB

Server-side API

MySql DB

DB Sync API

CRUD

CRUD

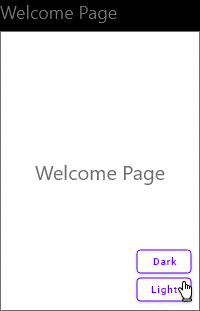
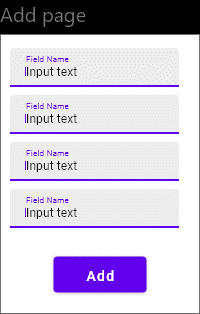
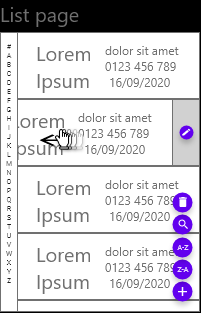
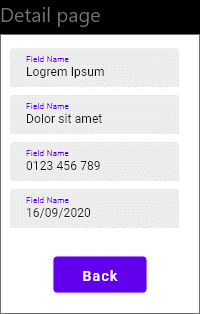
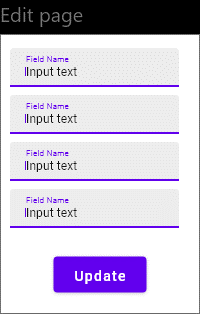
Each component and its functionalities are explained in the following sections.

The database, which is an RDS DB, shall have at least one table with **at least 4 columns besides the ID column, with at least one column of DataTime type.**

1. **Sub-systems requirements**

**Android application**

1. Android App page navigation

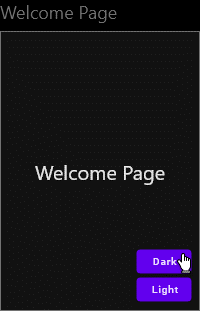
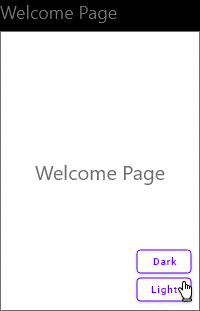


The mobile application consists of 4 pages. The navigation relationships of the four pages are as follow.

* “Welcome Page” displays welcome information. A right swipe gesture on this page navigate user to the “List Page”
* “List Page” displays a list of items and provide following navigations to other pages
  + Floating Action Button  navigates user to “Add Page” to add an entry to the table in database.
  + Clicking on an individual item navigates user to the “Detail Page”.
  + Swiping right an individual item makes appear the edit button  , clicking which navigates user to “Edit Page”
  + Drag-drop an individual item to the bin icon delete the corresponding row in database table.
* In “Add Page”, clicking “Add” button navigates user back to “List Page”.
* In “Edit Page”, clicking “Update” button navigates user back to “List Page”.
* In “Detail Page”, clicking “Back” button navigates user back to “List Page”.

1. Support two colour themes

In the “Welcome Page”, user can select colour theme from two pre-defined colour themes, Light and Dark. Once selected, the colour preference shall be stored in device file system and to be used when next time app is started. Light theme is used if no preference stored.



1. CRUD functionalities

The mobile application provides CRUD functionalities to both client-side Database and server-side database.

* **Create**: “Add Page” is used for creation. By clicking “Add” button, a new entry will be added to server-side DB and / or *\** client-side DB.
* **Read**: “List Page” requires all entries to be read. “Detail Page” requires one entry to be read.
* **Update**: “Edit Page” is used for updating. By clicking “Update” button, the corresponding entry will be updated on server-side DB and / or *\** client-side DB.
* **Delete**: In “List Page”, drag-drop action allows deleting one entry on server-side DB and / or *\** client-side DB.

*\* Depending on network availability, server-side DB may not be reachable, sync up may be required when network is back*

1. Hash, sort, and search

In the list page the following functionalities shall be implemented to improve usability

* **Hash**: A hash table shall be implemented to grouping the data read from DB table. The hash table has 27 indexes from 0~26 representing “#” (meaning non-alphabetic character) and A(a)~Z(z). The hash function will take into account the first character of a given “String” and determine Hash index. For example, string “Adam” maps to hash index 1 (meaning A or a), “1Joe” maps to hash index 0 (meaning non-alphabetic character).

Use double-linked list to resolve hash collision, and items shall be sorted alphabetically for each list.

For UI, in “List Page” a left-side navigation bar with 27 buttons shall be created for quick navigation. That is to say, for example, when user click on button “K”, the list will auto scroll to the items that start with “K” or “k”.

* **Sort**: When user click on button  or , all items in the main list shall be order correctly in ascending or descending alphabetical order.
* **Search**: After search is performed, a list of the items that meet search criterial shall be presented in the “List Page”.

1. Databases synchronization

All data in both server-side DB and client-side DB need to be synchronized any time when the application is running. However, when network is not available, CRUD functions can be performed in client-side DB only, and data will not be in synchronized status before network is back. The mobile application shall be able to synchronize the two databases when network is restored, and **user shall be informed when synchronization status changed**.

1. Advanced features

You are required study and apply Android API to implement at least **one** of the following features for the app.

* Click on a phone to make a call
* Send SMS
* Using camera

**Server-side API**

Server-side HTTP-based API shall provide CRUD functions for accessing server-side database. The API shall be designed in RESTful manner.

1. **Technologies and Algorithms**

The following technologies and algorithms should be adopted for each part of the system.

|  |  |  |
| --- | --- | --- |
| **Android** | **Server-side** | **Others** |
| Java or Kotlin | PHP or C# with or without framework like Laravel and ASP.net | MVC design pattern |
| Activity (Fragment if necessary) | RESTful API structure | Git |
| ModelView | MySQL or SQL Server | Search, Sort, Hash algorithm |
| Room + LiveData |
| SQLite |
| Animation |
| Touch, gesture, drag-drop |

1. **Software testing and performance analysis**

A test plan shall be created to test the functionalities of the software.

Software shall be tested following the test plan, and results to be reported.

Performance analysis shall be performed, and result shall be documented and reported.

1. **User document**

A user level of document shall be created and published to a website with public access.

1. **Publish mobile app**

Android Package (APK) file shall be generated and published to a website with public access.

1. **Documentation requirements**
2. Code following recommended code conversions.
   * Java: <https://google.github.io/styleguide/javaguide.html>
   * Kotlin: <https://kotlinlang.org/docs/reference/coding-conventions.html>
3. Use source code version control system to manage project. You are free to choose any one of the online system like [Github](https://github.com/), [Bitbucket](https://bitbucket.org/), or [Gitlab](https://gitlab.com/)
4. Source code must be fully documented, according to code conversion guides provided in “A” above.

**Part 3. Questions**

You are required to finish all the questions in this part for final submission. The questions in this part are grouped into 8 phases which map to the same 8 phases of the implementation.

For each phase, it is recommended to read the questions and Part 2 specification first before starting programming.

**Phase 1 – Plan & Preparation**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| In this phase, you are required to review the requirements provided in Part 2 and then finish the following tasks:   * Identify the development environment * Setup and verify development environment * Complete an implementation plan and reach agreement with client. | | | | | | |
| **Q1.1** Identify development environment and complete the following table. | | | | | | |
| |  |  | | --- | --- | | (A) OOP language chosen for Android app | The OOP language that is chosen is Java for android side | | (B) IDE for Android app development | The IDE that is used for app development is Android Studio | | (C) Test environment for Android app |  | | (D) Another OOP language for server-side system (must be different from the one chosen in (A)) | C# will be used for the designing of the Web-API | | (E) IDE for server-side system development | For server-side system development is Visual Studio | | (F) Test environment for server-side system |  | | (G) Source code version control system | Git Hub will be used for version control | | | | | | | |
| **Q1.2** Setup the chosen development environment and provide screenshot for each | | | | | | |
| |  |  | | --- | --- | | IDE for Android app development | */\* create a test project and screenshot the IDE to show the project structure \*/* | | Test environment for Android app | */\* run the test project and screenshot the result \*/* | | Run debugger in the Android IDE with test code.  *\*You are required to create test code with loop and “if” conditions, setup breakpoints, navigate code in debug mode, watch variable values* | */\* either screenshot the debug process \*/* | | IDE for server-side system development | */\* create a test project and screenshot the IDE to show the project structure \*/* | | Test environment for server-side system | */\* run the test project and screenshot the result \*/* | | Source code version control system | */\* create a repository on selected git server and screenshot the empty repository \*/* | | | | | | | |
| **Q1.3** Review the requirement and write down how do you plan to implement the required system; then present to your client for approval. | | | | | | |
| */\* use chart, diagram, or table to facilitate the planning \*/*    Client signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | | | | |
| **Checklist (To be completed by the learner’s facilitator)** | | | | | Yes | No |
| 1. Learner has correctly identified the development environment. | | | | |  |  |
| 1. Learner has successfully setup the development environment. | | | | |  |  |
| 1. Learner has written down an implementation plan. | | | | |  |  |
| 1. Learner has presented the plan to client and agreement has been reached. | | | | |  |  |
| **Assessor Name** |  | **Assessor Signature** |  | **Date** |  | |

**Phase 2 – Client-side Android app user interface design**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| In this phase, you are required to analysis the UI requirements and complete building the “view” part of the Android app. The following Android components should be used to build up the app in this phase:   * [Activity](https://developer.android.com/guide/components/activities/intro-activities?hl=en) with layout resource * [Fragment](https://developer.android.com/guide/components/fragments?hl=en) with layout resource (optional) * [ViewModel](https://developer.android.com/topic/libraries/architecture/viewmodel?hl=en)   You are also free to choose any other Android components (e.g. [RecyclerView](https://developer.android.com/guide/topics/ui/layout/recyclerview?hl=en) to create a list of something) and technologies (e.g. [support different screen size](https://developer.android.com/training/multiscreen/screensizes)) for the app where needed to meet the UI requirements.  In addition, you are also required to practice the following:   * Develop according to code conversion document * Follow the MVC design pattern to layer source code properly. * Document source code as you program * Manage source code with version control software   Database is not required and should NOT be involved in this phase, but interface to database should be considered. In order to feed data to the UI, static variables can be used to hold the data and play a role of database. After database is implemented in next stage, you can simply switch the data source from static variables to actual database interfaces. | | | | | | |
| **Q2.1** Design test plan for UI functionalities. Paste the test plan file name below.  Note that CRUD function with static variables need to be implemented and verified in this phase. | | | | | | |
| */\* Test plan file name \*/* | | | | | | |
| **Q2.2** Review “Android application / Android App page navigation” requirements. Briefly explain how you use Android “Activity” and / or “Fragment” components to implement multiple view pages and navigations between them? | | | | | | |
| */\* How to use Activity / Fragment to create multiple pages \*/*  */\* How to implement navigation between pages \*/*  With multiple views being created a button can be established on both pages one on the first which will have an onclicklistener which will establish an intent and the second page having the same except will redirect back to the first page. Using intents and passing in the current instance of an activity and the target activity as parameters can be used to redirect a user to different pages. To initiate this action a startActivity method is required which will pass in the intent which was created as an argument. | | | | | | |
| **Q2.3** Build all pages and correctly implement page navigations as required.   * Screenshot all pages created in layout editor * Screenshot the code sample that for page navigation | | | | | | |
| */\* Screenshot all pages* *\*/*    */\* Screenshot page navigation code \*/* | | | | | | |
| **Q2.4** What is Activity lifecycle? In terms of lifecycle, explain what happens to activity when auto-rotating and auto-resizing the device? | | | | | | |
| */\* Explain Activity lifecycle* *\*/*  An activity lifecycle in android refers a series of methods which are called at various stages of an activities life cycle for example at the beginning of an activity the OnCreate method is called which creates and sets up the view, the OnStart method is called after this which makes the app display to the user. At the end of an activity’s lifesycle is the OnDestroy method which at this point allows the system to garage collect the activity ensure that any redundant data is removed to free up space for the application.  <https://www.javatpoint.com/android-life-cycle-of-activity>  */\* What happens to activity when auto-rotating and auto-resizing the device \*/*  When an activity auto rotates the dimension of the app change to accommodate for the size alteration for example when an app is displayed vertically the Y dimension is much larger, however when the rotation occurs the size will favour the X dimension whilst the Y is reduced. When auto sizing, for example with text views within an activity, similar to auto-rotation the dimensions of the elements will change however the size of the text may also be adjusted.  <https://developer.android.com/guide/topics/ui/look-and-feel/autosizing-textview> | | | | | | |
| **Q2.5** Explain what lifecycle events are triggered when state changed? Screenshot your code where life cycle events are responded. | | | | | | |
| */\* Explain lifecycle events* *\*/*  When an activity first starts, the onCreate activity is called allowing the creation and setting up the UI of the activity, another example is when the onStart activity is called which is done after the OnCreate activity which makes the activity display to the user.  <https://abhiandroid.com/programming/onstart-method-in-android.html>  */\* Screenshot your code* *\*/* | | | | | | |
| **Q2.6** Discuss the options in Android to save app data in file system, for example save user configuration. Discuss at least two options | | | | | | |
| */\* Option No.1* *\*/*  One option is using internal storage to File class which receives a file directory and file name as arguments, with using FileOutputStream object you can utilize this to store values files which have data associated with them. For example using:  FileOutputStream fos = openFileOutput(fileName);  Fos.write(internalStorageBinding.saveFileEditText.getText().toString().getBytes());  This binds the file with internal storage of an application.  */\* Option No.2* *\*/*  Another option is to utilize shared preferences however it is useful if your storing data as key-value pairs. Android stores this data in a private directory within an xml file, typically these are used to store application preferences. For the creation of a Shared Preferences you must declare a SharedPreferences object, and use getSharedPreferences method when instantiating which requires passing an a file name and a MODE\_PRIVATE. For modifying values you must use the edit() method by using SharedPreferences.Editor and assigning a name such as editor and instantiating it with sharedpreferences.edit() with this you can insert values and keys. This is done by using:  editor.putString(key, value) | | | | | | |
| **Q2.7** How do you save colour theme configuration in file system when user make a selection in “Welcome Page”? Screenshot code sample. | | | | | | |
| */\* Explain how do you save configuration in file system* *\*/*  The color configurations are saved utilizing the SharedPreferences class, when the theme is switched the preferences are set to edit allowing the changing of the true or false associated with the isDarkThemeKey, which determined whether the theme changes or not. If set to false, dark mode is not selected, if true dark mode is selected. The editor of shared preferences will commit the data, then the recreate method id called to remake the activity to apply the selected them.  */\* Screenshot your code* *\*/* | | | | | | |
| **Q2.8** In Android what technologies are used to auto adjust position of UI elements when screen size changed? Discuss at least two technologies and their use case | | | | | | |
| */\* Technology No.1* *\*/*  One technology which can be used to adjust positioning of UI elements can be a constraint layout, this binds ui elements together using constraints which sets the distance between by a specified value. For example if you had a textview and a button by setting the bottom constraint of the textview to the top of the button this sets the distance between these elements, these constraints of these elements can also be applied to parent setting the end, start or top to match that of the parent component.  */\* Technology No.2* *\*/*  Another technology can be DP which is a dimension value, which is also called device-indepeent pixels, what these do is define the line density of dp objects. Specifically, DP is defined by a screens dpi value, dp defines the pixels of an element by multiplying the dpi which divides by 160pixels (dp \* (dpi/160)).  <https://medium.com/@ramtintoosi/android-dp-fully-explained-46052e5b6aa> | | | | | | |
| **Q2.9** In your app, how do you auto adjust position of UI elements in response to the change of screen size? Screen code sample as well. | | | | | | |
| */\* Explain how you auto adjust UI position* *\*/*  In order to adjust screen size a constraint layout is used, this enables elements within the layout to be constrained to one another with the use of constraints which are located on the top, bottom, left and right of an element. Constraints are either pulled from their origins to the target element or they are programmed in XML, these adjust the positioning of the elements relative to the targeted constraints.  */\* Screenshot your code* *\*/* | | | | | | |
| **Q2.10** Data saved in Activity class may lost when Activity is rebuilt, i.e. during auto-rotation. In Android, what technology is used to prevent losing data? Explain how does it work? | | | | | | |
| */\* Name the technology* *\*/*  The technology that is used for data loss is called the ViewModel  */\*Explain how it works \*/*  ViewModel Works by creating a Model which will inherit the ViewModel Class, this will allow the establishing of get and set methods of the given object it is meant to represent. A method will need to be created which will act as a constructor as it passes in parameters of the object if said object is null then a new instance of the object is created with the parameters given. Within the activity class the ViewModel declared which need to be instantiated with a ViewModelProvider, this requires the context of this class as well as a get method which will receive a ViewModel.class. Once this created you can implement the view model in an activity by using the current instance of the ViewModel for example Model.getId will retrieve the id of the object. | | | | | | |
| **Q2.11** Explain and screenshot the code to show that you have used proper technology (separating the model and view) to prevent losing data when Activity is rebuilt (e.g. rotate device). | | | | | | |
| */\* Explain your implementation* *\*/*  */\* Screenshot your code* *\*/* | | | | | | |
| **Q2.12** Explain how to build a list view in Android app? | | | | | | |
| In order to build a list a activity which holds the list which will be a recycler view component will be required as well as a list item. The list item holds the specified data items that will be used in a recylerview adapter to interact or manipulate data. An example can be a contact which textviews which hold contact information such as name, phone numbers as well as birthdates. | | | | | | |
| **Q2.13** Android components, for example RecyclerView, that used for creating a list view often requires a Data Adapter. Use the list view component you selected as example and explain how Dataset (as model), Adapter and List View component work together. You may explain by using code samples from you project. | | | | | | |
| A dataset holds the list of data items that will be displayed on each list item, the adapter provides the ability to display each item on each list element by binding these pieces of data to them. The data set also used by the adapter to establish the size of the list. The adapter also contains an item view holder which represents a single element of a list item, the values of these are determined in the onBindViewHolder method. | | | | | | |
| **Q2.14** It is required that left swipe gesture navigate user from “Welcome Page” to “List Page”. Explain and screenshot your implementation in source code. | | | | | | |
| */\* Explain how you implement swipe gesture* *\*/*  */\* Screenshot your code* *\*/* | | | | | | |
| **Q2.15** In “List Page”, left drag an individual item will move the top layer to left and bottom layer with edit button will appear. When the top layer changes position, animation is used to smooth the transition. Explain and screenshot your implementation in source code. | | | | | | |
| */\* Explain how you move UI position in response to figure touch and movement* *\*/*  With an onClick method this can be used to implement functionality to move UI elements position  */\* Screenshot your code* *\*/*    */\* Explain how you implement changing UI position with animation* *\*/*  Regarding animations, translations were used to move UI elements, for example with the moveCard method it receives the x dimension and the offset of x, the card view and btnedit as arguments. Within the is method it gets the current x position plus the offset to determine how much of the x axis it should move, the snimation is sone by assigning the value of X to a PropertyValuesHolder class which will be used in an ObjectAnimator to initiate the animation. The edit button is also revealed before the animation occurs.  */\* Screenshot your code* *\*/* | | | | | | |
| **Q2.16** In “List Page”, drag-drop an item to “bin” icon will delete the item. Explain and screenshot your implementation in source code. | | | | | | |
| */\* Explain how you implement drag-drop* *\*/*  By applying an onLongPressListener on the viewholder within the ContactListAdapter, the creation of a ClipData.Item object was used to set the data of the current view selected which was passed to the method startDragAndDrop which also received the DragShadow and the view itself. Within the ListActivity a DragonListener was applied on the btnDelete which detects whether the dragged object touches it, if so delete else do nothing.  */\* Screenshot your code* *\*/* | | | | | | |
| **Q2.17** In “List Page”, click or tap an item will navigate user to detail page. Explain and screenshot your implementation in source code. | | | | | | |
| */\* Explain how do you response a tap on the item view \*/*  Using an onClickListener when an item is selected, then the S(Detail) button on the bottom right, the data is applied to a Bundle and passed into an intent when is then passed into the startActivity method which initiates the new activity. The data passed is assigned to their appropriate fields.  */\* Screenshot your code* *\*/* | | | | | | |
| **Q2.18** There are multiple instances that functions are triggered by clicking button. Use an example in the app to explain how do you implement button click. Provide screenshot for that as well. | | | | | | |
| */\* Explain how do you implement button click \*/*  A button click can be established by first initializing a Button object with an instance of a button which already exists in an activity. An onClickListener must then be created on the button object which is then assigned a new View.OnClickListener, an overridden onClick method will appear and within whatever action that one needs accomplished.  */\* Screenshot your code* *\*/* | | | | | | |
| **Q2.19** A left navigation system is required in “List Page”. And you required to use Hash technology to implement the feature. Explain how do you implement this using Hash technology and provide code screenshot | | | | | | |
| */\* Explain how the navigation with Hash technology \*/*  Whenever a button is clicked with the given letter, the navBtnClick is called determining whether the key is less than 0 or greater than 26, if so, return null. Else the offset is created based upon the key from the button received, whichever key is chosen is passed into the scrollToPositionWithOffset method. Whenever the button is clicked the associated onCLickListener is fired which will call the navBtnClick method, this passes the key which will scroll to the associated alphabetical character.  */\* Screenshot your code* *\*/* | | | | | | |
| **Q2.20** You are required to use double-linked list to resolve collision for the Hash table in the app. Explain how do you implement it and screenshot the code. | | | | | | |
| */\* Explain using double-linked list for resolving Hash table collision \*/*  The List is used to pass all data that resides within the hash table to a list which is then passed into the ContactListAdapter. The hash table contains an ArrayList of lists, this is used to hold values which share the same key for example having a firstname of Red and Ride, the hash sorting of the alphabetical character of R, without the list storing these values there would be a collision, either an error would ensue or the data would be overwritten.  */\* Screenshot your code* *\*/* | | | | | | |
| **Q2.21** In “List Page” sort functions are required. Explain the sorting algorithm you implemented and provide screenshot for that. | | | | | | |
| */\* Explain how the sorting algorithm your select \*/*  The algorithm that was used was a bubble sort, this was implemented to compare the last names of contacts and sort them in a descending manner. The list was filled with data from the database, and the passed into the sort method which was passed as an argument into when creating a new ContactListAdapter. This was all initiated within an onClickListener provided by the sort button. The new Adapter was set and applied as well as a new layout manager thus refreshing the recycler list.  */\* Screenshot your code* *\*/* | | | | | | |
| **Q2.22** In “List Page” search functions are required. Explain the search algorithm you implemented for Hash table and provide screenshot for that. | | | | | | |
| */\* Explain how the search algorithm your select \*/*  The algorithm that was used is a binary sort, this is incorporated by the middle index acting as the pointer, which is set by the index + (endIndex - index) / 2, The vaue of mid is determined for every iteration that the sort goes through, if the value of the string to search is less than the value in the list, the middle index will be decreased by 1, if greater middle index is increase by 1. If the value us found returns the contact corresponding with the given lastname.  */\* Screenshot your code* *\*/* | | | | | | |
| **Q2.23** Binary search tree can be an alternative approach for storing and searching. Create a program with selected OOP language to implement “binary search tree” with sample data and perform search. Explain how do you do it and screenshot the code. | | | | | | |
| */\* Explain how do you implement binary search tree for data storing and searching \*/*  For the storing of values, if the input value is greater than or equal to the current node of data then the value will be inserted on the left node. If the value of the right nide is equal to null then the null will be filled with the given piece of data, the same applies to the left node. For the search if nothing is found return null, if the key is found then return the value of the key stored, if the key is less than the current node of data, search left node, else the right node will be searched.  */\* Screenshot your code* *\*/* | | | | | | |
| **Q2.24** Explain what is nested class in chosen programming language for Android app? Screenshot example in your code where a nested class is implemented. | | | | | | |
| */\* Explain nested class \*/*  A nested class, refers to classes which can be created within each other. The purpose of this is for a group classes in which they have only a single which may be exclusive to the class in which its declared in. For example within a recycler view adapter class there must be a ViewHolder which contains details about the view that is to be displayed to the UI, the view holder only has a use within the adapter.  */\* Screenshot your code* *\*/* | | | | | | |
| **Q2.25** Test the UI according to test plan using emulator or android mobile device, report test result using template. | | | | | | |
| */\* Test report file name* *\*/* | | | | | | |
| **Q2.26** Address any issue found in the test report. Use a software related issue as an example, explain how do you use IDE embedded debugger for troubleshooting and fix the bug. | | | | | | |
| */\* Problem Description* *\*/*  */\* Troubleshooting process* *\*/*  */\* Cause of the issue* *\*/*  */\* Solution Description* *\*/* | | | | | | |
| **Checklist (To be completed by the learner’s facilitator)** | | | | | Yes | No |
| 1. Test plan for phase 2 has been created | | | | |  |  |
| 1. Colour theme configuration can be saved and loaded from file system | | | | |  |  |
| 1. Page navigation works as expected | | | | |  |  |
| 1. CURD function works as expected with static variable as data source | | | | |  |  |
| 1. Tap, gesture, touch, drag-drop functions works as expected | | | | |  |  |
| 1. Animation function works as expected | | | | |  |  |
| 1. App correctly responds to device auto-rotating and auto-resizing | | | | |  |  |
| 1. Hash navigation function works as expected | | | | |  |  |
| 1. Sorting function works as expected | | | | |  |  |
| 1. Searching function works as expected | | | | |  |  |
| 1. Test is finished according to plan and results have been reported | | | | |  |  |
| 1. Issues identified during test have been addressed | | | | |  |  |
| 1. Software debugging process has been demonstrated and recorded | | | | |  |  |
| **Assessor Name** |  | **Assessor Signature** |  | **Date** |  | |

**Phase 3 – Client-side Local database design and connect to the UI**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Before starting Phase 3, you are required to have **Phase 2 CRUD** functionality programming completed, and verified. In this phase you are required to introduce Android local SQL database facility, the SQLite, into the application to replace the phase 2 data source, which is implemented with static variables as temporary solution.  For better practice, [Architecture Components](https://developer.android.com/topic/libraries/architecture/viewmodel?hl=en) should be leveraged for more robust SQLite database access. You should consider using the following Architecture Components from the [Jatpack](https://developer.android.com/jetpack) library:   * [LiveData](https://developer.android.com/topic/libraries/architecture/livedata) * [Room](https://developer.android.com/topic/libraries/architecture/room) ([DAO](https://developer.android.com/training/data-storage/room/accessing-data), [Room Entity](https://developer.android.com/training/data-storage/room/defining-data))   SQLite management tool is also required to manage local database. [*DB Browser for SQLite*](https://sqlitebrowser.org/) is recommended.  In addition, you are also required to practice the following:   * Develop according to code conversion document * Follow the MVC design pattern to layer source code properly. * Document source code as you program * Manage source code with version control software | | | | | | |
| **Q3.1** Design test plan for UI + SQLite functionalities. Paste the test plan file name below.  Note that a database management tool is required to verify the data in database. You should design at least one test case for using DB tool to verify data. | | | | | | |
| */\* Test plan file name \*/*  SQLLite Test Plan | | | | | | |
| **Q3.2** Explain how to use “[Room Entity](https://developer.android.com/training/data-storage/room/defining-data)” to define an entity (or table) for SQLite? | | | | | | |
| Room entity must be applied on top of an object name for example on top of “public class Contact” will be @Entity(tableName = “contacts”). The member variables must be assigned a column name which will be applied on top of these variables such @ColumnInfo(name = “variableName”) on top of an id of the entity must @PrimaryKey(autogenerate = true) | | | | | | |
| **Q3.3** Screenshot in your code that you have convert Data Model built in phase 2 to Room Entity. | | | | | | |
|  | | | | | | |
| **Q3.4** Explain how to use “[Room DAO](https://developer.android.com/training/data-storage/room/accessing-data)” to design database access interface for operations like CRUD. | | | | | | |
| For the creation of a Room DAO requires an interface with a room annotation of @Dao above the interface name, interface methods must be established with their appropriate Query annotations for example having an insertContact method will have @Insert, an updateContacts method will have an @Update annotation however for a specific functionality for example select all contacts you must have a query annotation such as @Query(“SELECT \* FROM contacts”). | | | | | | |
| **Q3.5** Screenshot in your code that you have coded all DAO interfaces for the app | | | | | | |
|  | | | | | | |
| **Q3.6** Explain the synchronisation technique adopted to synchronise data between the Android app and SQLite data source? That means if data is modified in database, corresponding data in UI elements is updated automatically and instantly. | | | | | | |
| A synchronisation technique that is used is that database singleton which will allow the calling of database functions throughout the application. By using this we call different functions which will add, update or delete contacts from the database dynamically. With the database being connected to the dataset of a recycler adapter the contacts are loaded into the list view and their properties displayed in each element to the user. | | | | | | |
| **Q3.7** Screenshot the synchronisation technique you implemented from your code. | | | | | | |
|  | | | | | | |
| **Q3.8** Explain what need to be done in order to connect the UI built in Phase 2 to the Room interfaces you have done in Phase 3. Screenshot code samples. | | | | | | |
| */\* Explain how to connect UI to Room interfaces \*/*  A way to connect UI to room interfaces is to use a Singleton for the database which will allows this to be used across the application. By calling the Database class and using the getInstance method to get the existing class, this will allow the utilizing of the interface methods for CRUD functionality.  */\* Screenshot your code* *\*/* | | | | | | |
| **Q3.9** Explain how do you use a SQLite management tool to manage the database for this app | | | | | | |
| By using the database inspector under app insector in the view tab, you can see the database’s structure and all the data that is within it. You can also change the values of each entities row, such as changing the first or last name of a contact, you can also view an entities id. Whenever a change happens on the UI end the change is reflected within the database, such adding or editing. | | | | | | |
| **Q3.10** Test the UI according to test plan using emulator or android mobile device, report test result using template. | | | | | | |
| */\* Test report file name* *\*/* | | | | | | |
| **Q3.11** Address any issue found in the test report. Use a software related issue as an example, explain how do you use IDE embedded debugger for troubleshooting and fix the bug. | | | | | | |
| */\* Problem Description* *\*/*  */\* Troubleshooting process* *\*/*  */\* Cause of the issue* *\*/*  */\* Solution Description* *\*/* | | | | | | |
| **Checklist (To be completed by the learner’s facilitator)** | | | | | Yes | No |
| 1. Test plan for Phase 3 has been created | | | | |  |  |
| 1. Android development features for accessing local database have been identified. | | | | |  |  |
| 1. Database interfaces for the mobile app have been implemented | | | | |  |  |
| 1. Database operations including CRUD functionalities have been verified | | | | |  |  |
| 1. Using DBMS management tool for local database has been demonstrated | | | | |  |  |
| 1. Test has been finished according to plan and results are reported | | | | |  |  |
| 1. Issues identified during test have been addressed | | | | |  |  |
| 1. Software debugging process has been demonstrated and recorded | | | | |  |  |
| **Assessor Name** |  | **Assessor Signature** |  | **Date** |  | |

**Phase 4 – Server-side database design**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Although it is recommended, but you don’t need to finish Phase 2 & 3 before starting Phase 4 & 5. Client-side applications depend on server-side services, but server-side services are designed to work independently in a “passive” mode.  In phase 4 you are required to design the server-side database for the system to store the data.  You also need to confirm the server-side technology stacks for Phase 4 and 5.  You are free to select the server-side technology for the system. But you must select a different language used in Android app. That means, if you used Java for Android app, server-side Java technology is not an option. Common server-side stacks are:   * [XAMPP](https://www.apachefriends.org/index.html) or [Uniform Server](https://www.uniformserver.com/) * [ASP.Net](https://dotnet.microsoft.com/apps/aspnet) * [Node JS](https://nodejs.org/en/) * [Java](https://www.java.com/)   In addition, you are also required to practice the following:   * Develop according to code conversion document * Follow the MVC design pattern to layer source code properly. * Document source code as you program * Manage source code with version control software | | | | | | |
| **Q4.1** Confirm server-side stack for Phase 4 & 5. | | | | | | |
| |  |  | | --- | --- | | **Name of server-side stack** | ASP.NET Core | | **OOP language** | C# | | **Database** | Relational Database | | **DBMS management tool** | Microsoft SQL Server | | **HTTP(s) server** | IIS | | **IDE for development** | Visual Studio | | | | | | | |
| **Q4.2** Setup and verify the target sever-side environment. Screenshot that the server-side setup is ready. | | | | | | |
| */\* E.g. screenshot that the related services are running, and test page is working* *\*/* | | | | | | |
| **Q4.3** Design server-side database structure and screenshot the design below. | | | | | | |
| */\* E.g. screenshot Entity Relationship Diagram (ERD), or DB/table creation SQL command* *\*/* | | | | | | |
| **Q4.4** Create the server-side database and all entities (or tables) and add some sample data. Provide screenshot when it’s done. | | | | | | |
| */\* Screenshot that db and tables have been created and sample data have been injected* *\*/* | | | | | | |
| **Q4.5** Using the selected server-side OOP language to create a simple webpage that retrieve data from the database created and verify the result using web browser. Screenshot the test code and web browser result | | | | | | |
| */\* Screenshot test code* *\*/*    */\* Screenshot web browser result \*/* | | | | | | |
| **Checklist (To be completed by the learner’s facilitator)** | | | | | Yes | No |
| 1. Server-side stack has been identified and confirmed. | | | | |  |  |
| 1. Server-side database has been designed and created with sample data. | | | | |  |  |
| 1. Server-side database is accessible by the server-side OOP language, and has been tested with web browser. | | | | |  |  |
| **Assessor Name** |  | **Assessor Signature** |  | **Date** |  | |

**Phase 5 – Server-side Web API design**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Before starting Phase 5, you are required to have Phase 4 completed so that you have a database and some sample data to work on.  For Phase 5, you are required to develop a set of web APIs for client application to consume, the following are required   * HTTP(S) based * RESTful manner * Data exchange with JSON * CRUD functionalities   You are also required to verify the web APIs with API testing tool, like [PostMan](https://www.getpostman.com/).  In addition, you are also required to practice the following:   * Develop according to code conversion document * Follow the MVC design pattern to layer source code properly. * Document source code as you program * Manage source code with version control software | | | | | | |
| **Q5.1** Explain what does “RESTful” mean in web API design. | | | | | | |
| */\* Your answer here* *\*/*  There are multiple architectural styles for APIs, one of these is Representational State Transfer or REST API. REST API compared to other API styles uses less bandwidth, this being the case it makes it more appropriate for efficient usage of the internet. This type of API can also give the ability for users to connect, manage and be used with cloud services flexibly. REST, in order to enable functionality utilizes already existing HTTP functions such as GET, PUT, POST and DELETE.  <https://searchapparchitecture.techtarget.com/definition/RESTful-API> | | | | | | |
| **Q5.2** Design “RESTful” web API structure and create an API document (a Word document or HTML file). Screenshot API document below | | | | | | |
| */\* File name of API document* *\*/*  Assesment2API  */\* Screenshot API document* *\*/* | | | | | | |
| **Q5.3** Design test plan for Web APIs. Paste the test plan file name below.  Note that you are required to use API testing tool, like PostMan for API test in the phase. Not until full tested with API testing tool, should the web APIs be consumed by a client application. | | | | | | |
| */\* Test plan file name* *\*/*  API Test Plan | | | | | | |
| **Q5.4** Test the web APIs according to test plan using API testing tool, report test result using template.  Note all issues found during the test must be properly addressed. | | | | | | |
| */\* Test report file name* *\*/*  API Test Report | | | | | | |
| **Q5.5** For at least one web API, screenshot the step-by-step process that you use API testing tool to test the API. | | | | | | |
| */\* Screenshot how do you do it* *\*/* | | | | | | |
| **Checklist (To be completed by the learner’s facilitator)** | | | | | Yes | No |
| 1. HTTP(s) based RESTful web APIs with CRUD functionalities have been implemented | | | | |  |  |
| 1. Web APIs have been tested with API testing tool. | | | | |  |  |
| **Assessor Name** |  | **Assessor Signature** |  | **Date** |  | |

**Phase 6 – Connect front-end Android app to back-end web APIs**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Now the Android app and web API are already. It’s time to establish web connection for the Android app and consume web APIs.  In Phase 6, you are required to utilize a third-party web API library for Android to consume the web APIs developed in previous phases.  [Volley](https://developer.android.com/training/volley) is an HTTP library that can be used for Android app. You can also choose other libraries.  Ideally, the system requires a well-designed strategy / algorithm to synchronize the local database and remote database. However, for this assessment, it is important for you to demonstrate the skill to call the API CRUD operations from Android app and get proper results. **You will NOT fail this assessment if the two databases are not synchronized.**  In addition, you are also required to practice the following:   * Develop according to code conversion document * Follow the MVC design pattern to layer source code properly. * Document source code as you program * Manage source code with version control software | | | | | | |
| **Q6.1** Identify the third-party HTTP library used for the project; and have the official documentation ready. | | | | | | |
| */\* Which HTTP library to use?* *\*/*  Retrofit is a third-party HTTP library, for which is utilized, in the context of this application, to allow the connection of the server-side web API to the client android application. The use case of this was for the implementation of a POST request to add a contact to the server-side DB  */\* URL to its official documentation* *\*/*  <https://square.github.io/retrofit/> | | | | | | |
| **Q6.2** What is the planned strategy to work with both local and remote databases? Consider the availability of the mobile network. | | | | | | |
| */\* Describe the strategy here \*/*  The strategy for working with local and remotes will be to utilize the read-all function from the API which will load the given list of contacts into the localDB, whenever a contact is added to the database the Create Contact function is called which will ad the contact to the Server-Side DB as well. When deletion occurs the contact is deleted by the id that is provided to the contact delete function which is passed to the web api and removes it from server-side. The edit functionality follows in a similar manner expect the updated object is passed and is altered in Server-Side DB. | | | | | | |
| **Q6.3** Design test plan for calling web API using the selected third-party HTTP library in Android app. Paste the test plan file name below. | | | | | | |
| */\* Test plan file name* *\*/*  Web API Test Plan | | | | | | |
| **Q6.4** Implement web API call in mobile app. Screenshot the Android app code where web API is called. | | | | | | |
| */\* Screenshot the code* *\*/* | | | | | | |
| **Q6.5** Test the Android app calling web APIs according to test plan, report test result using template.  Note all issues found during the test must be properly addressed. | | | | | | |
| */\* Test report file name* *\*/*  Web API Test Report | | | | | | |
| **Q6.6** Implement any one of the following advanced features   * Click on a phone to make a call * Send SMS * Using camera   Provide screenshot of the source code and test result. | | | | | | |
| */\* Which feature did you implemented?* *\*/*  **The feature which was implemented is the camera function.**  */\* Screenshot the source code* *\*/*      */\* Screenshot the test result* *\*/* | | | | | | |
| **Checklist (To be completed by the learner’s facilitator)** | | | | | Yes | No |
| 1. Web API connectivity and data manipulation has been implemented | | | | |  |  |
| 1. Test plan has been created | | | | |  |  |
| 1. Web API connectivity and data manipulation has been tested and test results have been recorded. | | | | |  |  |
| 1. One advanced feature has been implemented | | | | |  |  |
| **Assessor Name** |  | **Assessor Signature** |  | **Date** |  | |

**Phase 7 – Software test and performance analysis**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| In the previous 6 phases, all the subsystems have been created. It is the time to create a “end-to-end” test plan to verify the all required functionalities according to the overall design specification.  Also in this Phase, you are required to analysis the performance of the Android app, and report the result. | | | | | | |
| **Q7.1** Design end-to-end test plan overall system according to system specification. Paste the test plan file name below. | | | | | | |
| */\* Test plan file name* *\*/*  End-To-End Test Plan | | | | | | |
| **Q7.2** Test the system according to test plan, report test result using template.  Note all issues found during the test must be properly addressed. | | | | | | |
| */\* Test report file name* *\*/*  End-To-End Test Report | | | | | | |
| **Q7.3** Address any issues found in the test report. Explain how do you troubleshoot and resolve the issue. | | | | | | |
| */\* Problem Description* *\*/*  When sorting an error arises where the position of the view holder is set to a null reference object, the issue was caused by an intergace which it’s method was to get the position from the view holder and pass to the parent view.  */\* Troubleshooting process* *\*/*  Looking at the stack trace pointed to the position of the code within the Recycler Adapter where the interface method was meant to receive the current position from the adapter.  */\* Cause of the issue* *\*/*  The position was being passed to the method as null.  */\* Solution Description* *\*/*  Removing the interface from the code entirely as there was no need for it, its purpose was rendered redundant. | | | | | | |
| **Q7.4** Establish at least 3 performance objectives and explain how did you apply them to the Android app? | | | | | | |
| */\* 3 objectives* *\*/*  Refer to <https://developer.android.com/topic/performance>  For reducing of battery depletion contact data are stored in a local database SQLite.  To reduce memory the utilization of RecyclerView to ensure that contact data aren’t all stored on a single page at the same time, when the app scrolls data is loaded when its visible. The recycler recuses views to display the data.  For the reducing of app size the implementation of drawable was incorporated.  */\* How are they applied* *\*/*  **For reducing battery size, the data was stored in a SQLite database saving the constant connection between the WebAPI, removing unnecessary battery usage.**  **To reduce memory the Recycler view was used to hold contacts which re-uses the views when a view is offscreen, saving the memory of storing any unused views.**  **To reduce app size, a drawable of a circle was used instead of redeclaring the values to create the circle repeatedly, the drawable is called from the resources folder under drawable to wherever the circle is needed.** | | | | | | |
| **Q7.5** Conduct performance analysis and report the result using provided template. | | | | | | |
| */\* Analysis report file name* *\*/*  Enable performance window, run the project, collect performance graph for analysis.  Performance Analysis Assessment 2 | | | | | | |
| **Q7.6** Usingproper tools to identify useless and garbage application on a target device, then delete them. Explain how did you do it? | | | | | | |
| */\* Explain how you did it* *\*/*  Within the emulator for the removal of unnecessary apps, you can navigate to settings and under apps and notifications you can select which application to uninstall. For example if you have a practice application such DarkLightModeDemo you can select the app under apps and notifications and select uninstall. | | | | | | |
| **Q7.7** Review application against system specifications and address variances if there are any. Please make notes below. | | | | | | |
| */\* Review notes* *\*/*  */\* The following variances have been address and explain how* *\*/* | | | | | | |
| **Q7.8** Present the system to client and get approval from the client | | | | | | |
| Client signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | | | | |
| **Checklist (To be completed by the learner’s facilitator)** | | | | | Yes | No |
| 1. End-to-end test plan has been created | | | | |  |  |
| 1. End-to-end test has been executed and results have been reported | | | | |  |  |
| 1. All issues found during the test have been addressed. | | | | |  |  |
| 1. Learner has presented the system to client and client has approved the result. | | | | |  |  |
| **Assessor Name** |  | **Assessor Signature** |  | **Date** |  | |

**Phase 8 – Documentation and quality control**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| In this phase, you are required to create a user level document.  Also you need to confirm the following requirements are met for quality control purpose   * Develop according to code conversion document * Document source code as you program * Manage source code with version control software | | | | | | |
| **Q8.1** Develop user help document for the system in HTML format. | | | | | | |
| */\* Screenshot the document* *\*/*  Easiest way, create a user document with MS Word, save as HTML file. | | | | | | |
| **Q8.2** Explain and confirm that your codes follow provided conversion document | | | | | | |
| */\* Explain and screenshot code samples* *\*/*  Java conversion doc <https://google.github.io/styleguide/javaguide.html>  **Classes reside in file of their own**    **If**, **else** and **for** statements are contained within braces. Indentation is also applied. | | | | | | |
| **Q8.3** Confirm that your codes are fully documented | | | | | | |
| */\* Confirm and screenshot code samples* *\*/*  Comment the code | | | | | | |
| **Q8.4** Confirm that your codes are managed by a version control system | | | | | | |
| */\* Confirm and screenshot the version control system,* ***URL must be included*** *in the screenshot \*/*    Upload the code to github | | | | | | |
| **Checklist (To be completed by the learner’s facilitator)** | | | | | Yes | No |
| 1. User level document has been created | | | | |  |  |
| 1. Development followed provided conversion document | | | | |  |  |
| 1. All codes are documented. | | | | |  |  |
| 1. Source codes are managed by version control system. | | | | |  |  |
| **Assessor Name** |  | **Assessor Signature** |  | **Date** |  | |

**Phase 9 - Publish mobile app and handover project**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| The mobile, user document (html page) need to be published for public download. You are required to   * Research and document how to publish Android app to [Google Play Store](https://play.google.com/) * Publish Android app to either 1) Google Play Store; or 2) a website with public access * Publish user document (html page) to a website with public access | | | | | | |
| **Q9.1** How to publish Android app to [Google Play Store](https://play.google.com/)? | | | | | | |
| Describe the steps | | | | | | |
| **Q9.2** Publish Android app to a site with public access, and screenshot the result. | | | | | | |
| */\* Screenshot result.* ***URL must be included*** *in the screenshot* *\*/*  Generate APK file, upload to github | | | | | | |
| **Q9.3** Publish user document to a site with public access, and screenshot the result. | | | | | | |
| */\* Screenshot result.* ***URL must be included*** *in the screenshot* *\*/*  Upload user guide to github | | | | | | |
| **Checklist (To be completed by the learner’s facilitator)** | | | | | Yes | No |
| 1. Learner understand how to publish Android app to Google Play Store | | | | |  |  |
| 1. Android app has been published with public access. | | | | |  |  |
| 1. User document has been published with public access. | | | | |  |  |
| **Assessor Name** |  | **Assessor Signature** |  | **Date** |  | |

*End of document*