|  |  |  |  |
| --- | --- | --- | --- |
| Student Information | | | |
| **Student ID:** | 100610335 | **Student name:** | Jonathan Richardson |
| **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**

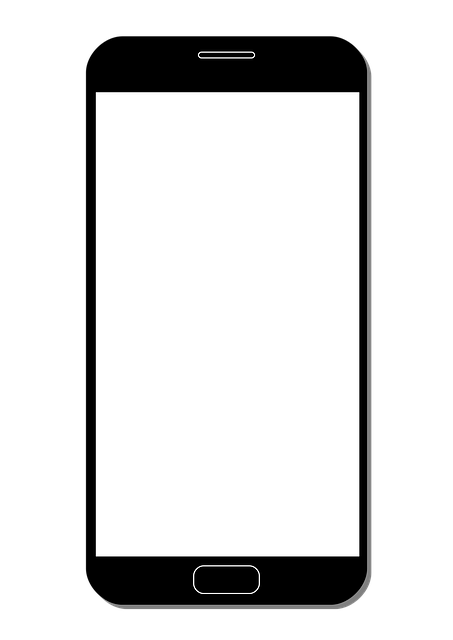
You are required to design an Android mobile app that meet the requirements given by graphic design and this specification. Graphic design resources are provided separately; requirements from the software perspective are explained in this document. Consult the client for further clarification as required.

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

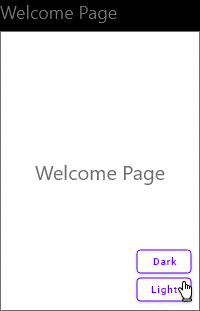
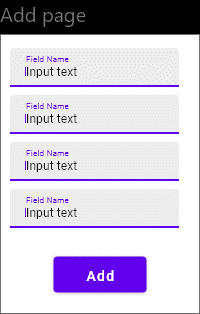
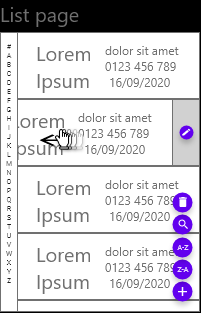
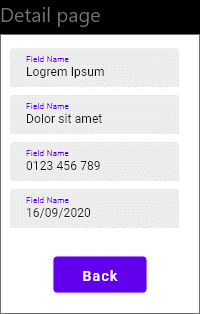
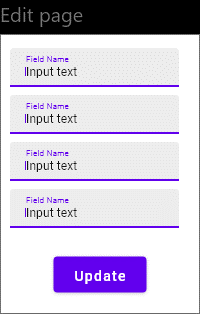
Each component and its functionalities are explained in the following sections.

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

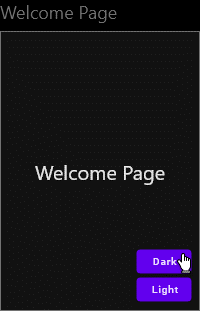
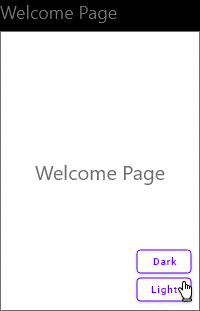


The mobile application consists of 4 pages. The navigation relationships of the four pages are as follow.

* “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 – Android Studio * 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 | JAVA | | (B) IDE for Android app development | Android Studio | | (C) Test environment for Android app | Mobile Device or Virtual Mobile Device | | (D) Another OOP language for server-side system (must be different from the one chosen in (A)) | C# | | (E) IDE for server-side system development | Visual Studio | | (F) Test environment for server-side system | Microsoft SQL Management Studio | | (G) Source code version control system | Github | | | | | | | |
| **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 \*/*  Test environment is testing the application on the Virtual Mobile Device or on an Android Phone | | 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 \*/*  Visual Studio – C#, ASP .NET Core Web API solution | | Test environment for server-side system | */\* run the test project and screenshot the result \*/*  *The API can be tested using the Swagger interface* | | Source code version control system | */\* create a repository on selected git server and screenshot the empty repository \*/*  <https://github.com/richo247/Assessment-2> | | | | | | | |
| **Q1.3** Review the requirement and write down how you plan to implement the required system; then present to your client for approval.  The application will be developed in JAVA, using Android Studio for the IDE. | | | | | | |
| */\* use chart, diagram, or table to facilitate the planning \*/*   |  |  | | --- | --- | | Features | Implementation | | Multi-screen application | Use ‘intents’ for multiple pages | | Phonebook List | Using recycler views for lists | | Local Database | SQLite & Room | | Server-side Database | Using C# | | API | Using Retrofit | | Additional Feature (Camera) | Using MediaStore |   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 \*/*  The activity/fragment are what makes up a page. When you create an activity, a class file and a layout file will be created. The layout and activity allow for us to create an android page. The layout is specifically UI while the activity is the code class.  */\* How to implement navigation between pages \*/*  You need to implement on click buttons and gestures for the user to navigate properly. To code this you need to create intent objects and then add a button which links to the method. | | | | | | |
| **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 of 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* *\*/*  Activity lifecycles are states that the app can be in when using them, while they are idle, when you destroy them and when you create/start them. Lifecycles are important to for quality of life things and can help improve battery/memory consumption.    */\* What happens to activity when auto-rotating and auto-resizing the device \*/*  When you auto rotate or resize on the device the app will essentially die for a brief moment and then recreate itself in the proper orientation. The app will need to reload all resources and configs that relate to the screen orientation. | | | | | | |
| **Q2.5** Explain what lifecycle events are triggered when state changed? Screenshot your code where life cycle events are responded. | | | | | | |
| */\* Explain lifecycle events* *\*/*  ON\_CREATE: on create is used for when the app is created.  ON\_DESTROY: on destroy is an event to destroy the app.  ON\_PAUSE: on pause is used pause the app but keep data.  ON\_RESUME: on resume is when you come back from pause and it resumes what it was doing.  ON\_ANY: on any allows you to code for any event.  ON\_START: when you start the app.  ON\_STOP: when you end the app.  */\* 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* *\*/*  Shared preferences: the shared preferences are saved in XML as key value pairs and stored in a private directory. The preferences are used to save preferences and app settings.  */\* Option No.2* *\*/*  SQLite is a relational database which is built in into Android, SQLite database is built and created only for individual applications and restricting access from any other android app. | | | | | | |
| **Q2.7** How do you save colour theme configuration in file system when user makes a selection in “Welcome Page”? Screenshot code sample. | | | | | | |
| */\* Explain how do you save configuration in file system* *\*/*  Colour schemes and preferences are saved using the shared preferences which is a built in android studio library which uses XML to store key value pairs.  */\* 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* *\*/*  Constraint layout: Constraint layouts act as a container and restrict movement of certain items so that they can’t go off screen. You need the constraint layouts to make sure that when the phone rotates, the items will end up where they need to be.  */\* Technology No.2* *\*/*  DP or SP instead of Px: density independent pixels and scale independent pixels are used if you want to auto adjust font sizes based on orientation or user settings. | | | | | | |
| **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* *\*/*  UI elements will respond and adjust their positions based on the orientation and the constraint layout.  */\* Screenshot your code* *\*/* | | | | | | |
| **Q2.10** Data saved in Activity class may be 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* *\*/*  View model technology from Jetpack Library  */\*Explain how it works \*/*  The view model class will store UI data for an application which will keep the data in memory. | | | | | | |
| **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* *\*/*  We made a view model class to store our hash table. Which interacts with the list page. Our contact list is stored in our hash.myhash view model.  */\* Screenshot your code* *\*/* | | | | | | |
| **Q2.12** Explain how to build a list view in Android app? | | | | | | |
| 1. Create a layout. 2. Create an item layout. 3. Create the item view holder which holds each item in the recycler view list. 4. Create a recycler view adapter. 5. Implement all methods needed. | | | | | | |
| **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. | | | | | | |
| We store a contact list in view model in the hash.myhash object. The adapter is an instance of main list recycler view adapter which holds the contact list where the view model is stored. The adapter represents the contact list and is stored in the view. | | | | | | |
| **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* *\*/*  To use swipe gesture you need to import the gesture detector compatibility library which we use for gesture listeners. The gesture listener takes in user input such as swipes and flings and performs actions.  */\* 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* *\*/*  To move ui elements in response to the users touch you need an on touch listener attached to an element. Each element will have a x/y location like coordinates and you can track where an item is being dragged to by tracking the coordinates.  */\* Screenshot your code* *\*/*    */\* Explain how you implement changing UI position with animation* *\*/*  When the user drags and throws/flings an element, we return true.  */\* 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* *\*/*  You have an on click listener attached to a specific item that can be deleted, with that on click listener you can track the coordinates of the element as the user drags it and once it lands on the delete button which has an on drag listener, you can delete the element.  */\* 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 \*/*  To allow navigation from a contact into the full contact details you need to have an on click listener attached to the item, each contact unique position and when clicked, they will be passed to the detail page which will then display the full contact details.  */\* 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 \*/*  For button clicks you need to implement an on click listener attached to whatever element you want. You then write code that will run in the event that the button is clicked.  */\* 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 \*/*  Yes we are required to use hash technology to implement the left side navigation. How we implemented this was to essentially check what key was input between 0 and 26 which are just letters of the alphabet with # being 0. The off set value will lead to the letter we want.  */\* 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 \*/*  We create a list for each letter of the alphabet. Contacts will be put into different lists inside the hash table based on the first letter of their names. A would be at position 1 in the hash table.  */\* 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 \*/*  Each index in the hash table has a letter assigned to it in alphabetical order. We sort people into lists based on the letter first letter of their name. To sort them we just need to sort alphabetically and then change their position as needed, as alphabetical and reverse alphabetical.  */\* 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 \*/*  To search for a person we used a for each loop to loop through all the contacts and check the searching value against the name value. To end the loop we need to have either sorted through the whole table and found nothing or to have found a match and then return the view.  */\* 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 \*/*  Binary search is an algorithm that splits the data structure into two parts, left and right. With the lesser value going in the left and the greater value on the right. The searching element will be what directs which path it goes down, so if you’re searching for 1 in a list of 25 elements then the 1 would be on the left. For each iteration of the binary search the tree will be split again and again going left or right depending on the searching element and then once it finds the searching element it will end. | | | | | | |
| **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 \*/*  Nested classes are classes within another class. We use this as a grouping method to keep classes together if they belong to each other.  */\* 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* *\*/*  Test\_Plan.docx | | | | | | |
| **Q2.26** Address any issue(s) 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* *\*/*  No problems found | | | | | | |
| **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 \*/*  Test\_Plan.docx | | | | | | |
| **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? | | | | | | |
| You need a database class, entity class and dao interface. To create a table or entity you just need to define in the model classes where each value belongs by using @entity for table names, @primarykey for PKs, @columninfo for column names. Then you can insert data into these tables by creating objects based on the fields for that model. | | | | | | |
| **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. | | | | | | |
| The room data access objects are used for crud functionality. An interface class is used to store all dao functions and inside it will include an annotation to specify interfaces as dao interfaces by using @dao. | | | | | | |
| **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. | | | | | | |
| Synchronisation of data between the android app and the SQLite database was implemented through the loadlist function. Whenever a change occurs to data inside the database, be it from adding, updating, or deleting a contact, the loadlist function is called in order to save the data and refresh everything so that the hash table will now include the new things or exclude the deleted things. | | | | | | |
| **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 \*/*   1. Add room dependencies to the app’s gradle file. 2. Define model classes as a table using dao interface and assigning fields via @. 3. Create database class which represents the database in SQLite.   */\* Screenshot your code* *\*/* | | | | | | |
| **Q3.9** Explain how do you use a SQLite management tool to manage the database for this app | | | | | | |
| SQLIte database browser is the management tool that we use to manage the database for the phonebook app.  To use SQLite database browser is used to verify database data and it can open .db files in the device file explorer. | | | | | | |
| **Q3.10** Test the UI according to test plan using emulator or android mobile device, report test result using template. | | | | | | |
| */\* Test report file name* *\*/*  DO THIS | | | | | | |
| **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** | Retrofit & GSON | | **OOP language** | C# | | **Database** | “PhonebookDB” | | **DBMS management tool** | SSMS (Microsoft SQL Server Management Studio) | | **HTTP(s) server** | http://192.168.15.6:5000 | | **IDE for development** | Android Studio & 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* *\*/*  We used swagger for verification of the API’s functionality.    */\* 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* *\*/*  Restful stands for representational state transfer. It is commonly used for communication for web services. | | | | | | |
| **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* *\*/*  PhonebookDB\_WEB\_API.sln  */\* 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* *\*/*  DO THIS | | | | | | |
| **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* *\*/*  DO THIS | | | | | | |
| **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  GSON  */\* URL to its official documentation* *\*/*  <https://square.github.io/retrofit/>  https://github.com/google/gson | | | | | | |
| **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 android app mostly works with the local database rather than the remote/server side database. With the local DB we can store and cache data rather than having the user have to load it every time which requires internet connectivity. This approach is okay but would eventually take up too much data if you had too many contacts. | | | | | | |
| **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* *\*/*  DO THIS | | | | | | |
| **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* *\*/*  DO THIS | | | | | | |
| **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 I implemented was the camera.  */\* Screenshot the source code* *\*/*    */\* Screenshot the test result* *\*/* | | | | | | |
| **Q6.7** Implement communicates with another device, using a communication protocol (e.g. socket)  Note this program does not need to be part of the mobile app. | | | | | | |
| */\* Which feature did you implemented?* *\*/*  Web sockets operate over TCP as an upgrade to HTTP connections. This allows more complex communication between clients and servers using a single socket and is exposed using javascript interfaces. The web socket generally just makes communication much easier, especially for real time event driven communication and was introduced with the HTML5. | | | | | | |
| **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 | | | | |  |  |
| 1. Inter device communication has been programmed | | | | |  |  |
| **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* *\*/*  DO THIS | | | | | | |
| **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* *\*/*  DO THIS | | | | | | |
| **Q7.3** Address any issues found in the test report. Explain how do you troubleshoot and resolve the issue. | | | | | | |
| */\* Problem Description* *\*/*  No issues found | | | | | | |
| **Q7.4** Establish at least 3 performance objectives and explain how did you apply them to the Android app? | | | | | | |
| */\* 3 objectives* *\*/*  **Refer to developer.android.com/topic/performance**  **Optimize for battery life – data are cached in local sqlite db to avoid frequent network data echange**  **Reduce app size.**  CPU usage, network, and battery.  */\* How are they applied* *\*/*  Network: Most data is stored locally and doesn’t require constant internet connectivity.  Battery: The program is small and doesn’t use up much battery power.  CPU usage: The program doesn’t use up much CPU as it is rather small and doesn’t have many intense functions. | | | | | | |
| **Q7.5** Conduct performance analysis and report the result using provided template. | | | | | | |
| */\* Analysis report file name* *\*/*  DO THIS  Enable performance window, run the project and collect performance graph then analyze it. | | | | | | |
| **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* *\*/*  To find useless applications you need to use the emulator and select the unused applications then long click and uninstall them. | | | | | | |
| **Q7.7** Review application against system specifications and address variances if there are any. Please make notes below. | | | | | | |
| */\* Review notes* *\*/*  All requirements met for the basic application. Out of the 3 optional items I chose to do the camera. | | | | | | |
| **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* *\*/* | | | | | | |
| **Q8.2** Explain and confirm that your codes follow provided conversion document | | | | | | |
| */\* Explain and screenshot code samples* *\*/*  Java conversion doc @2.7 java:https://google.github.io java thing | | | | | | |
| **Q8.3** Confirm that your codes are fully documented | | | | | | |
| */\* Confirm and screenshot code samples* *\*/*  Comment 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 \*/*  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/)? | | | | | | |
| Explain how | | | | | | |
| **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 and 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*