



**SCHOOL OF COMPUTER SCIENCES UNIVERSITI SAINS
MALAYSIA**

**CAT201 INTEGRATED SOFTWARE DEVELOPMENT WORKSHOP
SEMESTER 1 2020/2021**

Java App Development Project

Group 22

App Developed: PocketWit

Lecturers: Dr. Tan Tien Ping & En. Azlan Osman

Submission Date: 17th January 2021

Group Members:

Names	Matric No.	E-mail
SAIFUL NAQUIB BIN MOHD YUSOFF	149941	saifulnaquib@student.usm.my
PAMELA RABECCA SARASWATHY	146958	pamelarabecca12@gmail.com
RAFIA HOSSAIN	149684	otherwiserh@gmail.com
KIRTINIROOPA SHANKAR	146127	kirtini31@student.usm.my

Table of Contents

INTRODUCTION.....	1
SOFTWARE AND HARDWARE REQUIREMENTS	2
CLASS DIAGRAM.....	6
FUNCTION DESCRIPTION	9
OVERALL ISSUES AND SUGGESTED IMPROVEMENTS	10
VERSION CONTROL LOG	11
GANTT CHART	122
OUTPUT AND USER GUIDE	13

INTRODUCTION

The majority of students attending at least one online course has almost multiplied over the last five years. Online students face the difficulty of balancing their educational commitments with their family or full-time career, which is definitely not a simple feat. Technology has helped students have a better control academic learning by offering everything from study aids and testing resources to time-management applications.

Our application, Pocket Wit is an Android based application that can be used during lessons and applied to the student's life outside of the classroom. This application does not require the use of a network connectivity which save a lot mobile data for the students. The motivation behind our application is to provide options for how students can use technology and do their work creates pathways to learning for all students. Besides that, Pocket Wit eliminates the need for paper, writing utensils and other school supplies. The use of educational apps undoubtedly serves to preserve trees.

Some of the functions included are, a note-taking function, a quiz function which includes the actual quiz itself with flashcards to go over them as a study material and a medical history function to review any health information. The concept of Pocket Wit is to provide all these functionalities in a single app.

In order to build our app, Pocket Wit, we used Android Studio. With Android Studio, the structured code modules allowed us to divide our project into units of functionality that we could independently build, test and debug.

SOFTWARE AND HARDWARE REQUIREMENTS

This app is built for Huawei Mate 20 Pro and similar phone screen size and resolution.

Phone Size : 6.39 inches, 100.2 cm² (~87.9% screen-to-body ratio)

Screen Resolution : 1440 x 3120 pixels, 19.5:9 ratio (~538 ppi density)

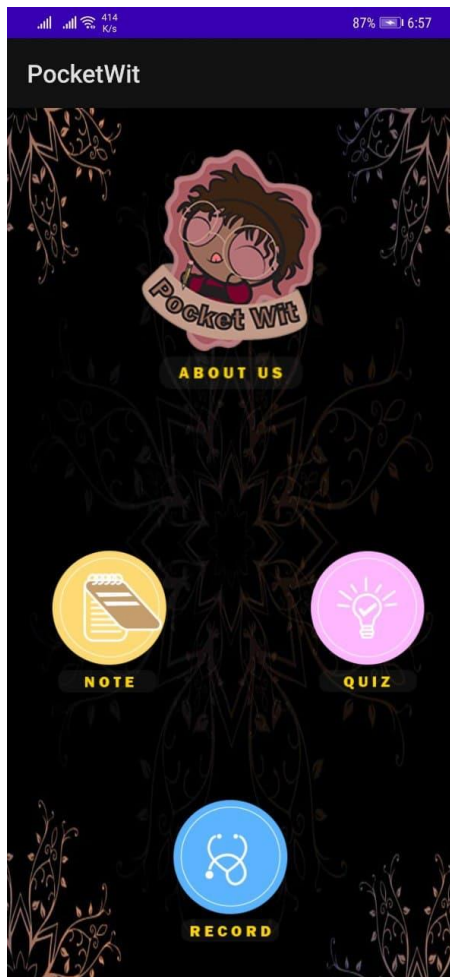
Android Version : Android 10 (This app support Android 4.1 and above)

Installing the PocketWit app in other than the designated phone size, screen resolution, and android version may cause instability, incompatible and non-functioning app. Thus, it is highly recommended that a phone with similar hardware requirements is used for the app installation.

After clicking the PocketWit icon and opening the app, the application shows three icons which are for 'NOTE', 'QUIZ', 'RECORD and 'ABOUT US'. The 'QUIZ' icon leads the user to the quiz function. In quiz user should be able to take a quiz and answer 15 questions. At the end of the quiz user should be able to retake the quiz or exit the quiz application. The 'NOTE' icon leads the user to the 'NOTE' page which should allow them to make new notes or edit existing notes. The 'RECORD' page lets users save their medical records and view them. Finally, the 'ABOUT US' icon leads users to the details of the app and its developers.



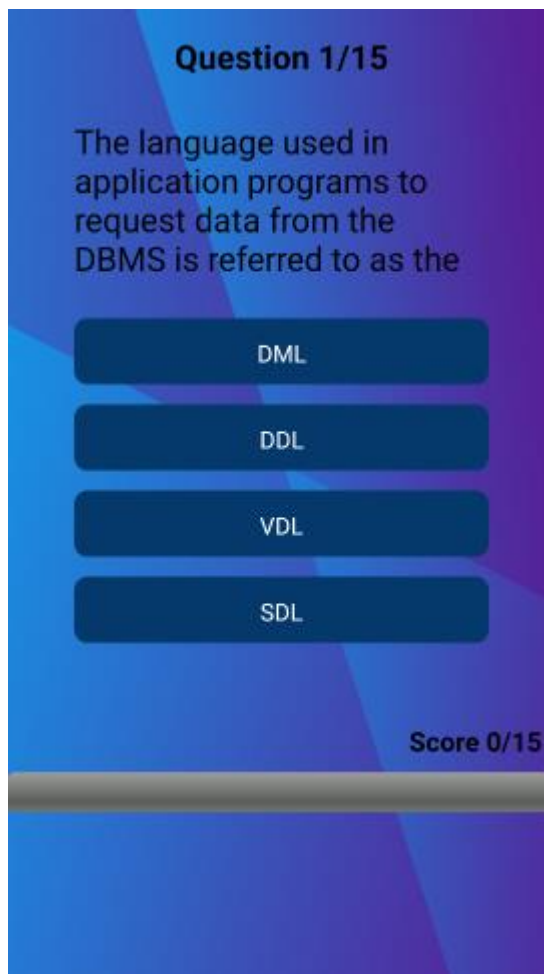
PocketWit App icon



Main page when app is opened



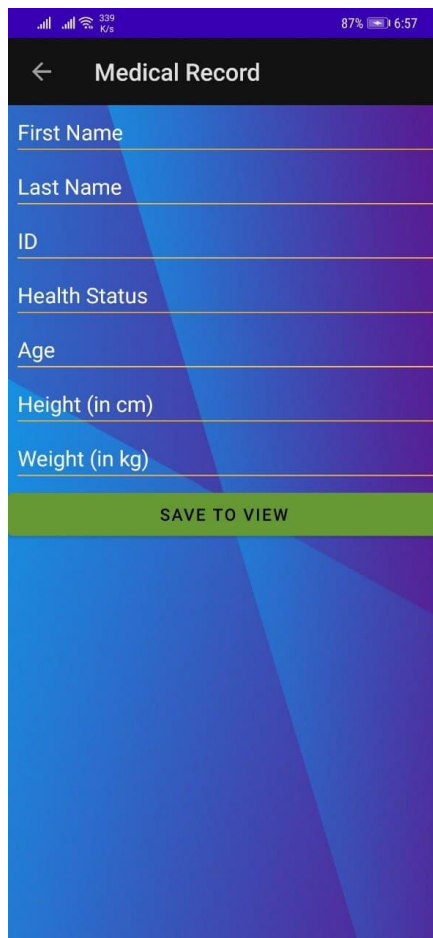
About Us page



Quiz page



Note page



The image shows a mobile application interface for a 'Medical Record'. At the top, there is a status bar with signal strength, 339 K/s, 87% battery, and the time 6:57. Below this is a dark blue header with a back arrow and the title 'Medical Record'. The form consists of several text input fields with light blue borders, stacked vertically: 'First Name', 'Last Name', 'ID', 'Health Status', 'Age', 'Height (in cm)', and 'Weight (in kg)'. A green button labeled 'SAVE TO VIEW' is positioned below the last field. The background of the app is a dark blue gradient with abstract light blue geometric shapes.

Medical Record

First Name

Last Name

ID

Health Status

Age

Height (in cm)

Weight (in kg)

SAVE TO VIEW

Medical Record page

CLASS DIAGRAM

Main Activity
- About Us : imageview - Note : imageview - Quiz : imageview - MedicalRecord : imageview - Navigation : intent redirect new layout activity
- About Us - # public boolean onTouch(View v, MotionEvent event) Notepad # public boolean onTouch(View v, MotionEvent event) - Quiz # public boolean onTouch(View v, MotionEvent event) - MedicalRecord # public boolean onTouch(View v, MotionEvent event)
Splash Screen
onCreate (Bundle): void
About Us
onCreate (Bundle): void

Notepad

- Note Option : listview
- Note : textview
- Menu : inflater
- Note Content : edittext
- Delete Note :

- Display Note : void

protected void onCreate(Bundle savedInstanceState)

- Delete Note : edittext

public boolean onItemLongClick(AdapterView<?> parent, View view, final int position, long id)



Answer

- OptionA : int
- OptionB : int
- OptionC : int
- OptionD : int
- Answer : int
- Question : int

+ Answer(question : int, a : int, b : int, c : int, d : int, answ : int) : int,int,int,int,int,int

+getOptionA(int) : int

+getOptionB(int) : int

+getOptionC(int) : int

+getOptionD(int) : int

+getQuestion(int) : int

+getAnswer(int) : int

Quiz / FlashCard
-optionA : TextView -optionB : TextView -optionC : TextView -optionD : TextView -questionNum : TextView -question : TextView -score : TextView -selectAns : TextView -correctAns : TextView +Tscore : int +qn : int +progressBar : ProgressBar +CurrentQuestion : int +CurrentOptionA : int +CurrentOptionB : int +CurrentOptionC : int +CurrentOptionD : int +PROGRESS_BAR : final int
#onCreate(Bundle savedInstanceState) : void -CheckAnswer(int) : void -UpdateQuestion() : void

MedicalRecord
+ first: TextView + last: TextView + id: TextView + health: TextView + age: TextView + height: TextView + weight: TextView + firstName: EditText + lastName: EditText + studentID: EditText + healthStat: EditText + studentAge: EditText + studentHeight: EditText + studentWeight: EditText
onCreate (Bundle): void + onSaveInstanceState (Bundle): void + onRestoreInstanceState (Bundle): void + saveView (Bundle): void

FUNCTION DESCRIPTION

The SplashScreen displays a picture when the app icon is clicked. When the about us icon is clicked it displays information about the app and its developers.

For the note function, it currently uses a minimalist layout and function. User only need to click the add button and type the content that they want to save. Then the note will be saved when they have done writing. For deletion, the user needs to highlight the note that they want to delete, then a prompt / dialog box menu button will appear for deletion. It will ask for user confirmation prior to deletion.

The quiz function currently holds 15 questions with its corresponding options and answers. The 15 questions are based on the course Database Organisation and design.

When a student first clicks the quiz icon in the PocketWit app, the first question should be displayed. On top of the screen the question number verses the total number of questions should be displayed whereas, on the bottom of the screen a progress bar and a score tracker are displayed. When the student answers the first question, the function will compare the chosen answer by the student and the actual answer. If the answer is correct, a “CORRECT” notification will appear at the bottom of the screen and the score will be incremented by 1. On the other hand, if the answer is wrong, a “WRONG” notification will appear at the bottom of the screen and the score will remain the same. Once the first question is completed, the progress bar will be slightly incremented showing that the quiz is progressing and proceeding forward.

After the first question, the second question is displayed and all the actions that occurred during the first question will repeat itself. When the fifteenth and final question has been answered, a message box should appear. In the message box, the student will be notified that the quiz has ended and the student’s quiz score will be displayed. The student will also be prompted to choose either to exit the quiz interface back to the main page or to retake the quiz.

The Medical Record function is to store student’s details. It saves the student’s details like First Name, Last Name, ID, Health Status, Age, Height and Weight.

When you open the PocketWit app and click on the icon, Medical Record, it will need to be filled in with the student’s details. Then, you click ‘Save to View’, to save the details.

The purpose of this function is to simply, store the student's information in the application for any future uses.

OVERALL ISSUES AND SUGGESTED IMPROVEMENTS

For the 'NOTE' function a search function could be implemented to allow students to search for a particular note. Then, we need to integrate the note function and app to a firebase. With firebase, we will be able to add more functionality to the note function. Such function is changing font type, font colour, the colour of each note background, adding a picture inside the note etc. However, by using firebase, we need to pay subscription service and learn extensive knowledge to apply the functionality, money and time is a luxury which we can't afford during this project timeline.

The quiz function that was built for this application is quite static. The questions and their corresponding options are not randomised causing a static quiz. Furthermore, if more questions are to be added or existing questions need to be edited, the codes in java and xml files need to be modified and the whole application needs to be updated for the student to get the modified quiz.

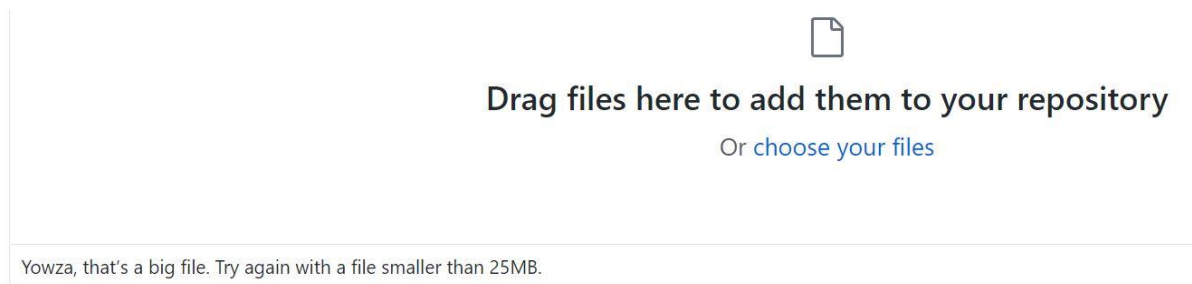
One of the improvements that can be made is to add more questions in the quiz function and randomize 10 to 15 questions every time the students take the quiz. In this way, the quiz function can be equipped with 100s of preinstalled questions and the students can get a variety of question each time they attempt the quiz. The students could also be given the option to choose the number of questions they would like to attempt at a time. The questions in the quiz can also be categorized to different courses to allow students to have quizzes on multiple subjects. The students can also be given the option of adding their own questions into the quiz function.

The Medical Record function is for medical purposes in this application that can only edit and save the information. The function is limited and not flexible. The Student Information function definitely could do more. Some suggestion for future improvements are, to be able to add other categories. Not just that, another suggested function would be the be able to share the Medical Record out of the application.

VERSION CONTROL LOG


Version	Date	Description of changes
1.0	15/12/2020	Quiz component (Kirtiniroopa Shankar 146127)
1.1	17/12/2020	Notes component (Saiful Naquib Bin Mohd Yusoff 149941)
1.2	19/12/2020	Medical records component (Pamela Rabecca Saraswathy 146958)
1.3	19/12/2020	Splash screen, fragments, about us and UI (Rafia Hossain 149684)
2.0	21/12/2020	Splash screen integrated with Quiz functionality
2.1	23/12/2020	Integration of Medical records
2.2	24/12/2020	Integration of Notes
2.3	30/12/2020	UI updates
Final	31/12/2020	Complete mobile app

For GitHub version control, our file is too big to be uploaded... Max file size is 25MB.
Screenshot of our proof.



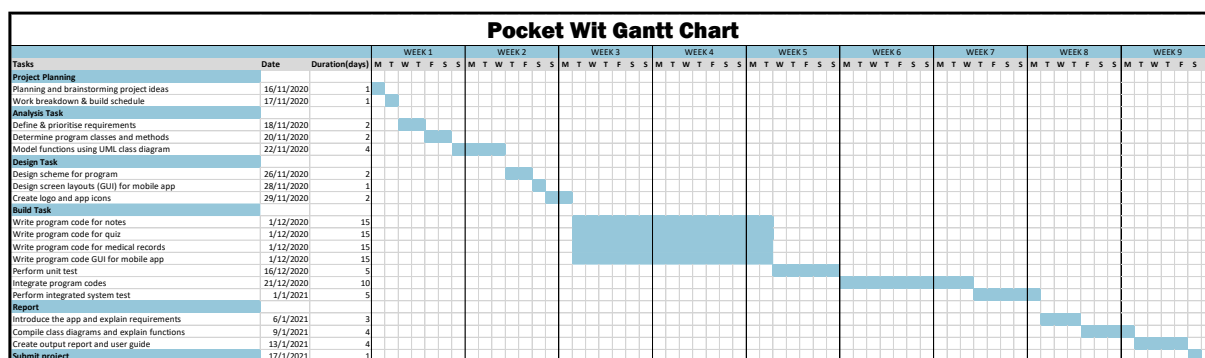
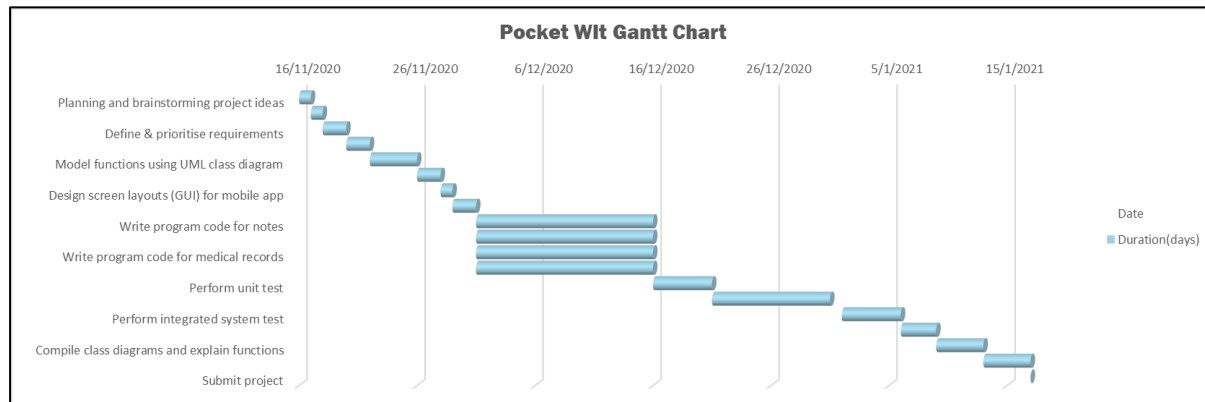
PocketWit : Android Studio File only (30MB)

PocketWit_CAT201 : Android Studio File + Report Video presentation + Screenshot (47MB)

 PocketWit	17/1/2021 7:08 PM	Compressed (zipp...	30,505 KB
 PocketWit_CAT201	17/1/2021 9:41 PM	Compressed (zipp...	47,420 KB

GANTT CHART

Pocket Wit Gantt Chart Work Breakdown		
Tasks	Date	Duration(days)
Project Planning		
Planning and brainstorming project ideas	16/11/2020	1
Work breakdown & build schedule	17/11/2020	1
Analysis Task		
Define & prioritise requirements	18/11/2020	2
Determine program classes and methods	20/11/2020	2
Model functions using UML class diagram	22/11/2020	4
Design Task		
Design scheme for program	26/11/2020	2
Design screen layouts (GUI) for mobile app	28/11/2020	1
Create logo and app icons	29/11/2020	2
Build Task		
Write program code for notes	1/12/2020	15
Write program code for quiz	1/12/2020	15
Write program code for medical records	1/12/2020	15
Write program code GUI for mobile app	1/12/2020	15
Perform unit test	16/12/2020	5
Integrate program codes	21/12/2020	10
Perform integrated system test	1/1/2021	5
Report		
Introduce the app and explain requirements	6/1/2021	3
Compile class diagrams and explain functions	9/1/2021	4
Create output report and user guide	13/1/2021	4
Submit project	17/1/2021	1



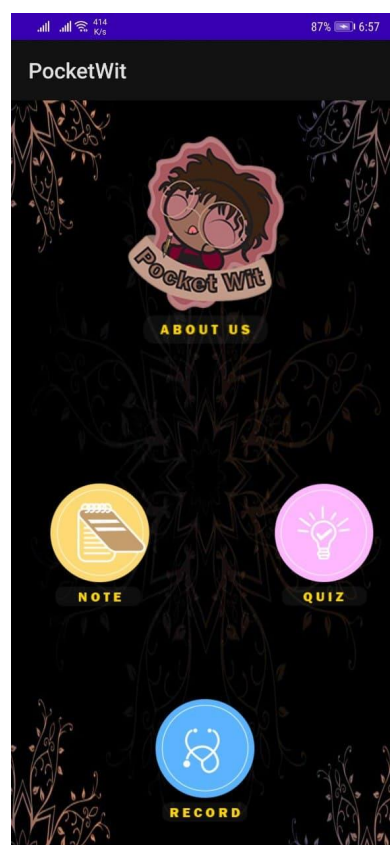
OUTPUT AND USER GUIDE



First Click The App Icon



Then Splashscreen will appear



Main Layout of the App after Splashscreen

Has 4 Submenu which are ABOUT US, NOTE, QUIZ & RECORD



Clicking ABOUT US submenu will redirect to About Us layout. It contains the information regarding the app and developer information.



Clicking NOTE submenu will redirect to the Note Pad Layout. It has pre-set instruction.



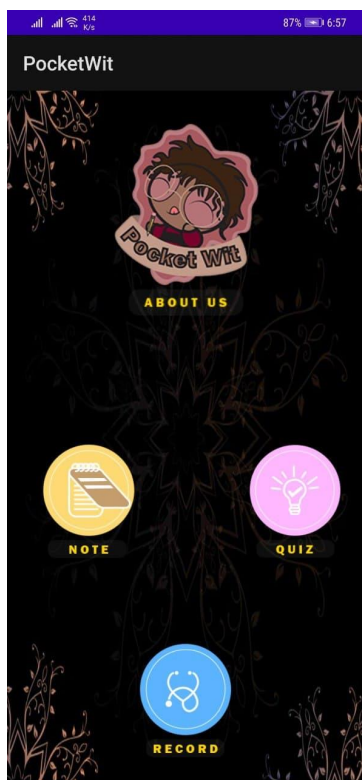
Clicking the Right Top Corner Button will prompt a new submenu, it will have the Add Note option.



When the Add Note option is click, user will be redirect to a new Note Editor Layout. User may type whatever note they want to store here. For example, Test is typed here



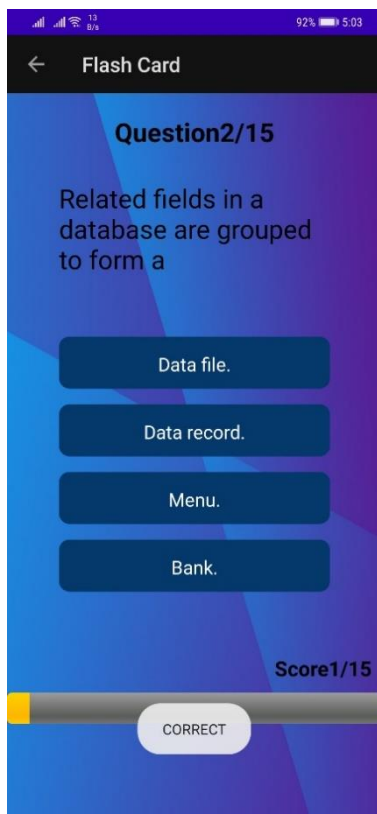
Click back or returning to the activity will save the note. As you can see, the note content Test has appeared.



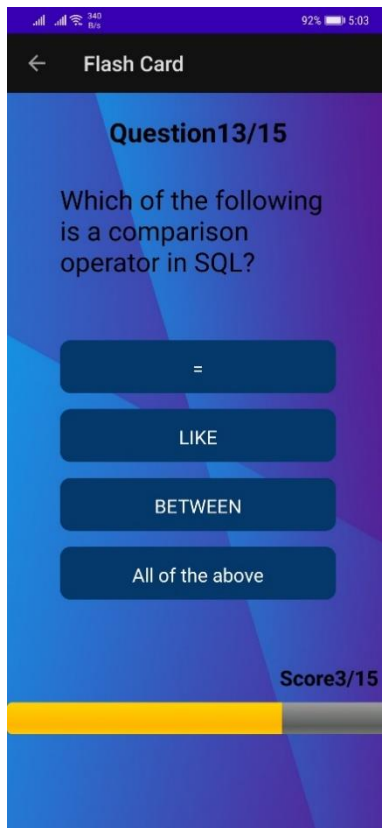
Back to the main menu. This time we are clicking QUIZ submenu.



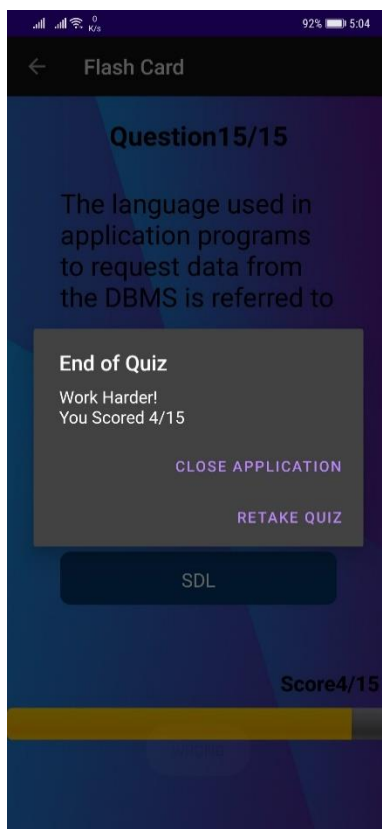
It will redirect to the Quiz / Flashcard layout.
It has 15 questions (as of now) and user may answer the MCQ questions.



When answered, the layout will then change to a new layout (next question). If you answered right, it would appear a notification box with the message CORRECT. Else, it will appear with the

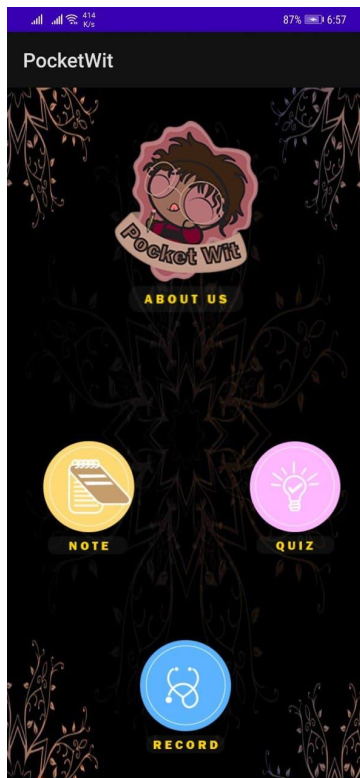


As you go through the next questions. The question number progress will be updated on top. The progress bar down below (yellow bar) will go to inform your milestone. The score of correct answer will also be updated.



When you completed the quiz, a dialog box will appear with message End of Quiz and your score result.

You may close the Quiz Activity and it will redirect you to the main menu. You may also retake the quiz.



When 'CLOSE APPLICATION' is clicked, we go back to the main menu. This time we are clicking RECORD submenu.

A screenshot of the 'Medical Record' form in the PocketWit application. The form has a dark header with a back arrow and the title 'Medical Record'. Below the header, there are several input fields for user information: 'First Name', 'Last Name', 'ID', 'Health Status', 'Age', 'Height (in cm)', and 'Weight (in kg)'. Each field is represented by a horizontal line with a small blue highlight on the left. At the bottom of the form, there is a green button labeled 'SAVE TO VIEW'. The background of the form is a dark blue and purple gradient.

Type in the details in the empty columns.

Medical Record

Andrew

Jackson

149941

Ok

21

170

70

SAVE TO VIEW

After the details are typed in.

Medical Record

Andrew

Jackson

149941

Ok

21

170

70

SAVE TO VIEW

Andrew

Jackson

149941

Ok

21

170

70

Click 'SAVE TO VIEW' to save the data.