Design and Development of Android-Based Mathematic Learning Application for Form 1 PT3 Students

Yong Wai Sheng[1], Mohd Norasri Ismail[1]*

[1]Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia (UTHM), Parit Raja, 86400, MALAYSIA

*Corresponding Author Designation

Abstract: Recently, mobile learning is a new trend to access learning content using mobile devices. However, there are insufficient educational apps on the market for the specific age group of students. At the same time, many secondary school students in Malaysia struggle with acquiring basic math skills and memorizing the formula. Therefore, the objective of this project is to create an organized platform for students to study mathematics. The Mathematics learning app called MATH#f1 will employ multimedia elements such as learning videos and infographics to attract students to learn Mathematics. Multimedia Mobile Content Development has been chosen as the development methodology for this project. MATH#f1 consists of seven modules: Login, Registration, Profile, Learning, Formula, Practice, and the Quiz module. Besides, the usability testing has resulted a score of 86.67 which is within the acceptable range based on the System Usability Scale (SUS). Therefore, it can be concluded that the MATH#f1 application can be used as a learning aid.

Keywords: Mobile Learning, Mathematics, Android, MATH#f1, Form 1

1.   Introduction

Mobile learning is a field that progresses fast and is already recognized as the future material of education [1]. Mobile devices users can earn new knowledge at any time and any place, even outside the school [2]. Technological evolution provides significant enhancement and spread the use of mobile education, as mobile devices become economical, more moderate, with better functionality and more extended battery life. More than 62,000 education-related apps obtainable in Google Play Store and IOS App Store, which are revolutionizing the current education [3]. Thus, a Mathematic learning application for Form 1 PT3 students called MATH#f1 was designed using Multimedia Mobile Content Development methodology. MATH#f1 has been developed based on the Malaysia Secondary School syllabus KSSM model with provided step by step teaching provided to improve students Mathematics knowledge.

Several issues arise from the research on the existing Form 1 Mathematics application. Many Mathematic learning apps are available in the app store, but the target user is not specific well or clear on a particular age group of students. Besides, only three Mathematics learning apps are found based on the Form 1 Mathematics syllabus in Google Play Store. In the current app store, the Mathematics learning apps related to Form 1 PT3 standard are very limited, which causes the student refuse to use the mobile learning app.

Therefore, the objectives of creating the MATH#f1 application are to design the content of the Mathematics learning app for Form 1 PT3 students based on the Malaysia Secondary School syllabus KSSM model, to develop the Mathematics learning app for the Android mobile platform, and to test the functionality and user acceptance of the Mathematics learning app to Form 1 PT3 students.

MATH#f1 is a Mathematics learning app used on the Android mobile platform for Malaysia Form 1 students. Form 1 students will be under the PT3 standard of learning based on the Malaysia Secondary School syllabus KSSM model. The chapter cover for the MATH#f1 application will be chapter 1 to chapter 6. MATH#f1 used a multifunctional module to improve student's understanding of the basic concept and solve the exam questions. There will be seven modules in the MATH#f1 application: Login Module, Registration Module, Profile Module, Learning Module, Formula Module, Practice Module, and Quiz Module.

In these proceedings, five sections are covering the project development. Section 1 is an introduction that describes the background of the project, which includes the problem statement that led to the project's development, the objectives achieved in the project, the importance of the project and the scope covered in developing the project. Next, section 2 is the literature review which covers a systematic analysis of scholarly articles and other sources related to the project topic, such as the technology used and a comparison between the existing applications that have been developed. While Section 3 describes the methodology chosen for the application development phase in detail using the Multimedia Mobile Content Development (MMCD). Section 4 describes the result and discussion of the project through the application testing, which consists of alpha and beta testing. Lastly, section 5 discusses the conclusion, including the advantages, disadvantages, and future improvement of the MATH#f1 application.

2.   Related Work

2.1   Domain Background

Mathematics is one of the core subjects that students should achieve in academic life. Based on research, student master in mathematics subject will tend to work professional job in future, for example, engineers and scientist [4]. Studying mathematics also generate more citizen who can learn and think creatively and critically, and it doesn't matter what their career field [5]. Therefore, Mathematics is incredibly crucial in our lives as we tend to use mathematical concepts as well as the skill that we learn from doing math problems to solve our daily problems.

In today's world, mobile learning is highly popular. Mobile learning, also known as mLearning, is a novel approach to access a wide range of online content through the use of a mobile device [6]. For students, mobile learning is the most convenient way to receive assistance.

2.2   Existing Application

The existing application for Form 1 Mathematics that available in Google Play Store such as Matematik Tingkatan 1 [7], Mathematics Form 1 Notes + Topical Revisions [8], and Form One Maths Notes [KLB] + Free Teachers Guide [9] have been review and compared with the MATH#f1 application. Table 1 displays the comparison of the Form 1 Mathematics application.
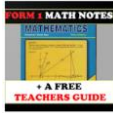
Table 1: Comparison Table

Factors	Matematik Tingkatan 1	Mathematics Form 1 Notes + Topical Revisions	Form One Maths Notes [KLB]	MATH#f1 application
Developer	Angelworks	Kaung Ka Lay	Samson mbugua	Yong Wai Sheng
Objective	Provide important formula and concepts in two languages.	Enable learner to understand well in a simple way	Create a very learner friendly mathematics learning app.	Create an organized platform for student to study mathematics
Related Content	Refer to Form 1 Mathematics syllabus	Refer to Form 1 Mathematics syllabus	Refer to KCSE standard	Refer to KSSM standard
Strength	- Free to use - Ease of use - Simple user interface design - Dual language - Provided step by step teaching - Suitable example for students in practice	- Free to use - Ease of use - Simple English language	- Free to use - Ease of use - Simple user interface design - Organize structure of learning module	- Free to use - Ease of use - Consistent button - Consist multimedia element - Consistent user interface design - Organize structure of learning module
Weakness	- Many advertisements appear randomly - Not follow the latest KSSM textbook syllabus	- Messy color combination in the entire app - Not follow the latest KSSM textbook syllabus. - Inconsistent layout - Difficult to understand the content	- Many advertisements appear randomly - Not follow the latest KSSM textbook syllabus	- Only cover chapter 1 to chapter 6 - Require login to access quiz module

Table 1 shows some distinguishing features between existing applications and developed applications in terms of the learning modules. It can be concluded that the MATH#f1 application is more suitable compare to others application as it refers to latest KSSM learning format, contain multifunctional module, organize structure of learning module and the content are verified by the subject matter expert.

3.   Methodology

Multimedia Mobile Content Development (MMCD) has been chosen as the methodology for this project. Many similar projects have proven that this methodology helps developers to achieve the objective in a short period, which can speed up the application development process and optimize the data usage and mobile processing usage [10].
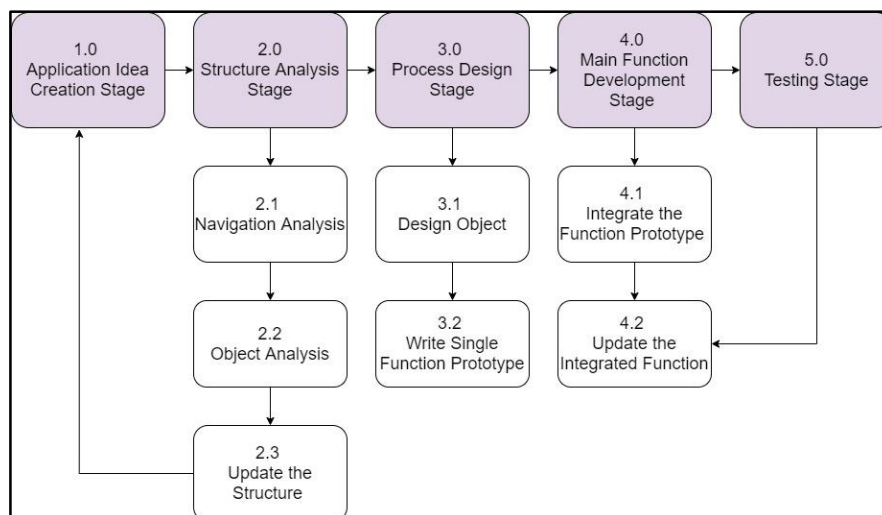

Figure 1: Multimedia Mobile Content Development model

Figure 1 shows the MMCD methodology consists of five main stages, and stage 2 to stage 4 consists of sub-stage according to the main stage. Thus, it is clear that this methodology provides a very comprehensive method in all stages for a multimedia mobile content developer to develop the mobile learning application, especially the MATH#f1 application, as MMCD methodology more focused on the objects used and content navigation.

3.1   Application Idea Creation Stage

For stage 1, the application idea creation stage is the first stage in MMCD methodology, which collects and prepares the necessary information before proceeding to the design stage and developing the application. Table 2 shows the checklist for this stage.

Table 2: Application Idea Creation Check List

Item	Note
Type of application	Learning application
Target device	Android platform
Target users	Form 1 PT3 students
GUI (Graphical user interface)	Login module, Registration module, Profile module, Learning module, Formula module, Practice module, Quiz module
Images	Icons, images of mathematic elements
Video	Learning video
Animation	GIF tutorial, App opening transition
Audio	Background music
Application synopsis	MATH#f1 is a Mathematics Learning application for Form 1 PT3 students that follow the latest KSSM format of learning which allow students to study mathematics in anywhere and anytime to improve their Mathematics knowledge.

3.2   Structure Analysis Stage

The structure analysis stage will help the project to create a well-structured and systematic analysis to minimize errors in the next implementation process. No doubt, having a complete analysis of the project will boost up the development process. Therefore, the task analyses will be explained before moving on to design the application structure.
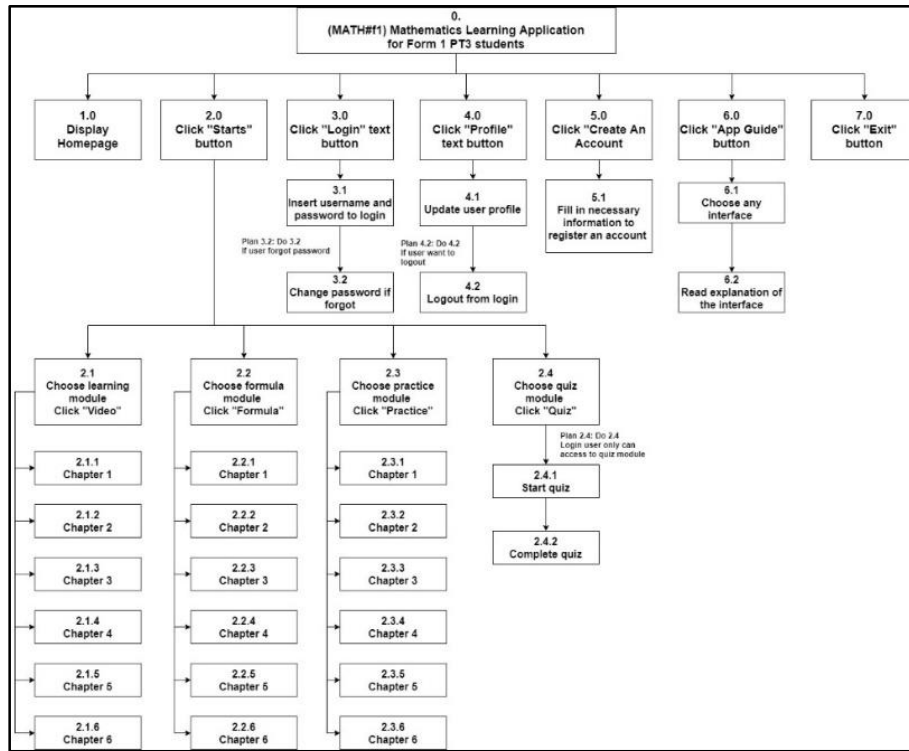
3.2.1   Task Analysis

Figure 2: Hierarchical Task Analysis

Figure 2 shows the Hierarchical Task Analysis chart that consists of seven main tasks and their subtasks. All tasks are arranged accordingly based on the main tasks. The caption of each plan has stated the condition to trigger the specific task. For instance, Task Plan 2.4 required the users to log in first before accessing the quiz module.

3.3   Process Design Stage

In third stage, it involves with the process of designing the structure of the application. There are four structures to be explained: Content Structure, Navigational Structure, Flowchart, and Database design.
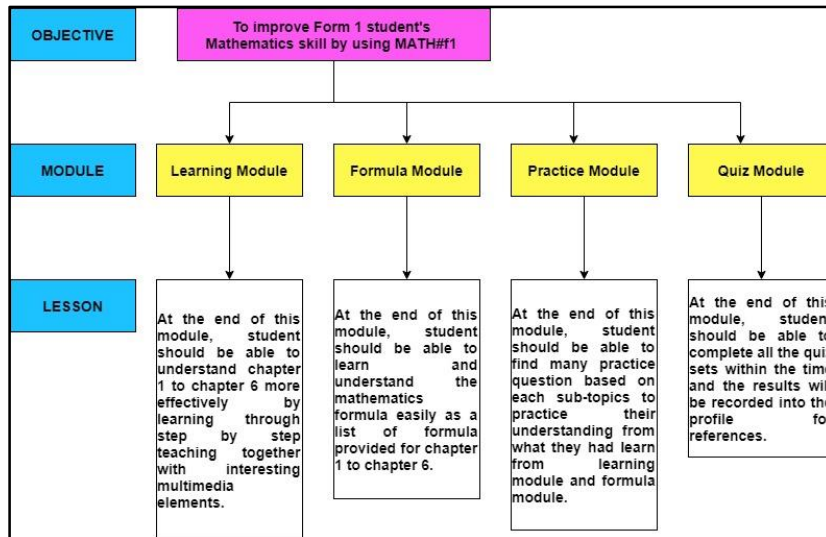
3.3.1   Content Structure

Figure 3: Content Structure

Figure 3 shows that the content structure is separated into three categories: objective, module, and lesson. The objective is to improve the Form 1 student's Mathematics skills by using MATH#f1. In contrast, several modules have been listed to clarify the module of the application. The user will learn from different modules to improve their Mathematics understanding from chapter 1 to chapter 6.

3.3.2   Navigational Structure

The navigational structure will clarify each interface's interaction with different buttons and provide the relationship between the button. A navigational structure has been made and shown in Figure 4.

Figure 4: Navigational Structure

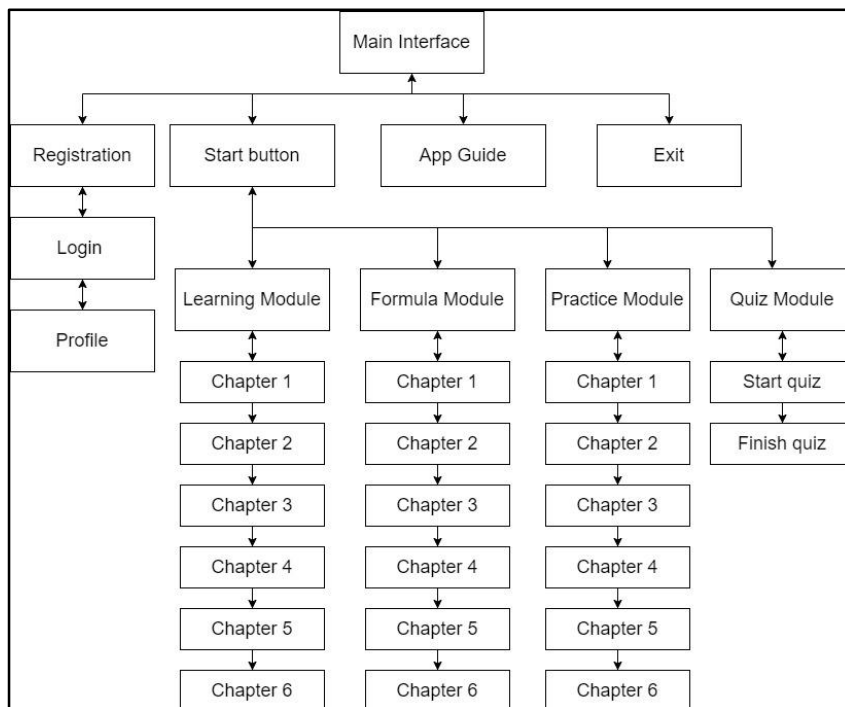Based on Figure 4, the navigational structure shows that the main interface consists of 4 main navigations: Registration, Start button, App Guide, and Exit. The Registration button will bring the user to the registration interface to sign up for an account. Users can log in with their registered email and password and view a personal profile. The Start button consists of 4 modules, which are learning,

formula, practice, and quiz. The app guide button will navigate the user to the app guide interface to guide the user on using the apps with a clear explanation on each button. The Exit button allows the user to quit the application.

3.3.3 Flowchart

The purpose of creating flowcharts is to communicate how an application should works. The flowchart will show the flow of each process in detail for the application. The flowchart has been designed and shown in Figure 5.
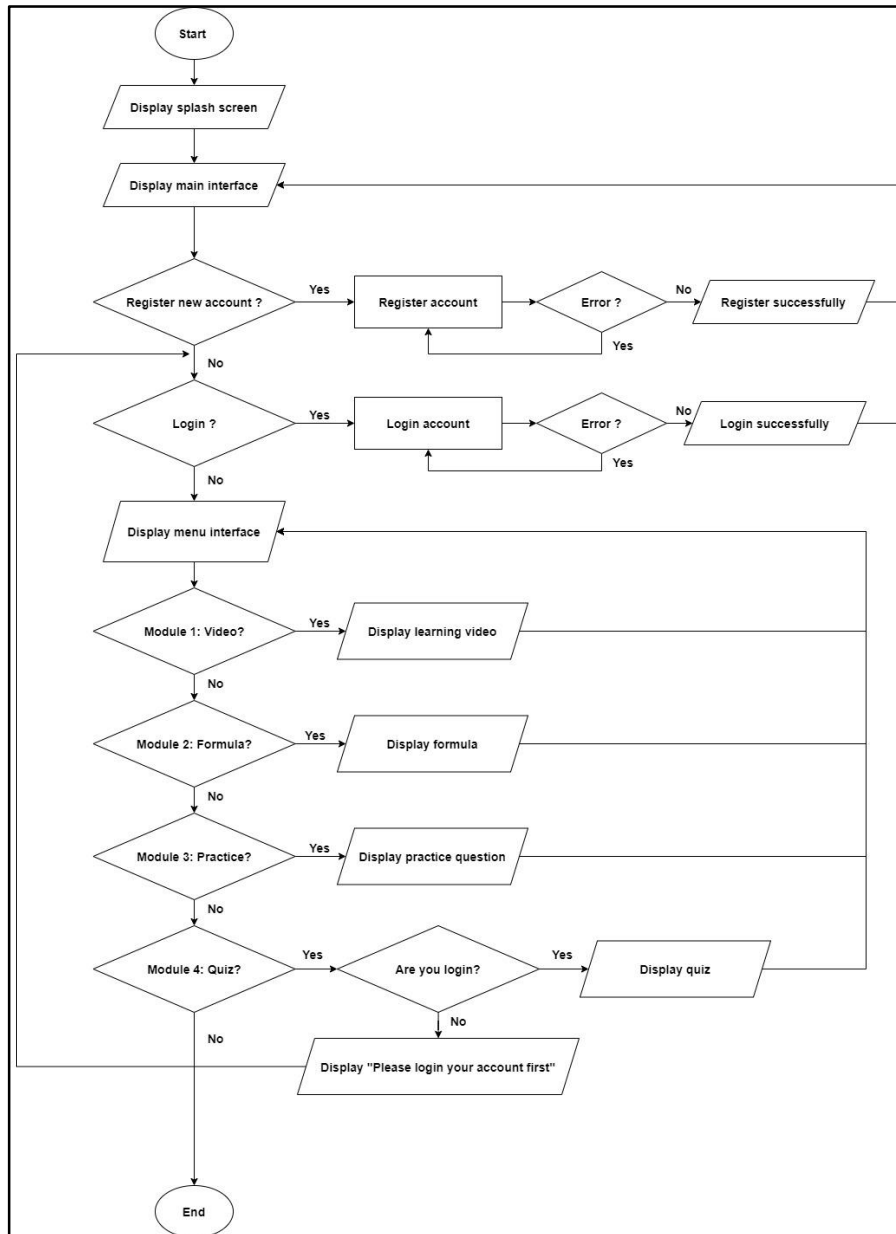
Figure 5: Flowchart

According to Figure 5, the MATH#f1 application will start with register an account and then proceed with the login. Once users have successfully logged in, users can proceed to the four main modules: Learning, Formula, Practice, and Quiz. From the four main modules, only the Quiz module are required users to log in before they can access it because the scores obtained from the quiz will be recorded to the user account.

3.3.4 Database Design

The design of the database can be sketched out by using the Entity Relationship Diagram (ERD). An Entity Relationship Diagram (ERD) is a sort of flowchart that shows how "entities" in a system, such as objects, people, or concepts, interact with one another. Figure 6 shows the ERD model for the MATH#f1 application.
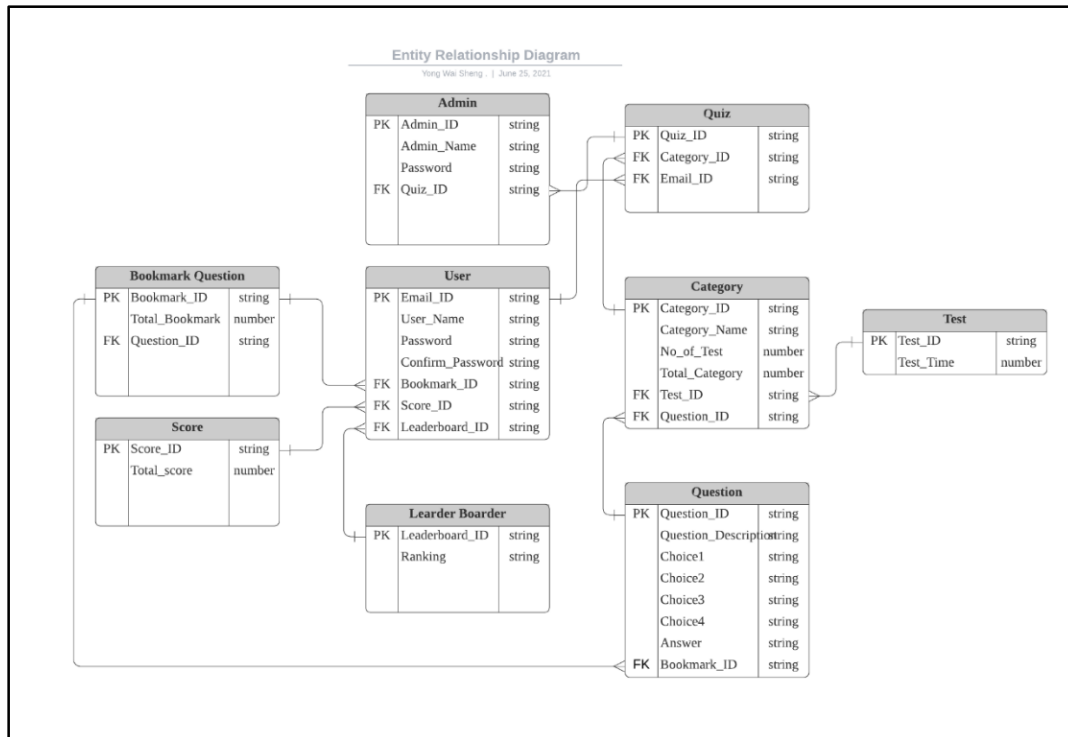

Figure 6: Entity Relationship Diagram

According to Figure 6, the Entity-Relationship Diagram consists of 9 entities and their respective primary key attributed to the diagram. For the user, the primary key is the email address which is unique for every user, whereas the foreign key consists of bookmark ID to store question, score ID to store the score, and the leader board ID to store the ranking of the users.

3.4 Main Function Development Stage

The main function development stage consists of three main parts which are Frontend Development (XML), Backend Development (Java), and Database Development (Google Firebase). After the application has been completed, the testing process will be carried out through black-box testing for the developer and user acceptance testing for the target user.

3.4.1 Fronted Development (XML)

Development of the user interface is the process of designing and building the interface, which focuses on the looks or style of the interface. In android app development, user interface layouts are designed by using XML. XML stands for Extensible Markup Language, which helps to design the interface of the app. The complete user interface can be referred in Appendix A.

3.4.2 Backend Development (Java)

Java programming is the technology of choice for building applications using managed code to execute on mobile devices. Java code is a backend language or server-side which responsible for storing and organizing data and ensuring everything on the client-side works as expected.

3.4.2.1 Main interface

In the main interface, the *setOnClickListener* is a method that used to allow the button to be press for further action. When the user presses the *textview* button, it will start the new activity, which is the Signup activity. Figure 7 will show part of the code segment for the backend development in the main interface.

```
TextView textView2 =(TextView)findViewById(R.id.txt_view2); //Register
textView2.setPaintFlags(textView.getPaintFlags()| Paint.UNDERLINE_TEXT_FLAG);
textView2.setOnClickListener((view) → {
        Intent i = new Intent( packageContext: Dashboard.this, SignUpActivity.class);
        startActivity(i);

        Toast.makeText( context: Dashboard.this,  text: "Create Account Here", Toast.LENGTH_SHORT).show();
});
```

Figure 7: Backend development of the main interface

3.4.2.2 Quiz Module

When the user presses the "Start" button to begin the quiz, the *loadQuestions()* function starts working on loading the question list from Firebase based on the unique ID of "CATEGORY" and "TEST". At the same time, both have to be equal to load the specific question. "CATEGORY" is representing for the particular chapter that selected by the user. "TEST" is representing for the number of tests being selected by the user. The *g_quesList* method will start loading the list of question that match these two criteria and display the questions to the user. Figure 8 shows the code segment for *loadQuestion()* function and Figure 9 will show the Question List in Firebase.

```
public static void loadQuestions(final OnQCompleteListener onCompleteListener)
{

    g_quesList.clear();

    CollectionReference collectionRef = g_firestore.collection( collectionPath: "Questions");
    collectionRef.whereEqualTo( field: "CATEGORY",g_catList.get(selected_cat_index).getDocID()) Query
            .whereEqualTo( field: "TEST", g_testList.get(selected_test_index).getTestID())
            .get() Task<QuerySnapshot>
            .addOnSuccessListener((OnSuccessListener) (queryDocumentSnapshots) → {

                    for(final QueryDocumentSnapshot doc : queryDocumentSnapshots)
                    {

                        boolean isBookMarked = false;

                        if(g_bmIdList.contains(doc.getId()))
                            isBookMarked = true;

                        g_quesList.add(new QuestionModel(
                                doc.getId(),
                                doc.getString( field: "QUESTION"),
                                doc.getString( field: "A"),
                                doc.getString( field: "B"),
                                doc.getString( field: "C"),
                                doc.getString( field: "D"),
                                doc.getLong( field: "ANSWER").intValue(),
                                NOT_VISITED,
                                 selectedAns: -1,
                                isBookMarked
                        ));
```
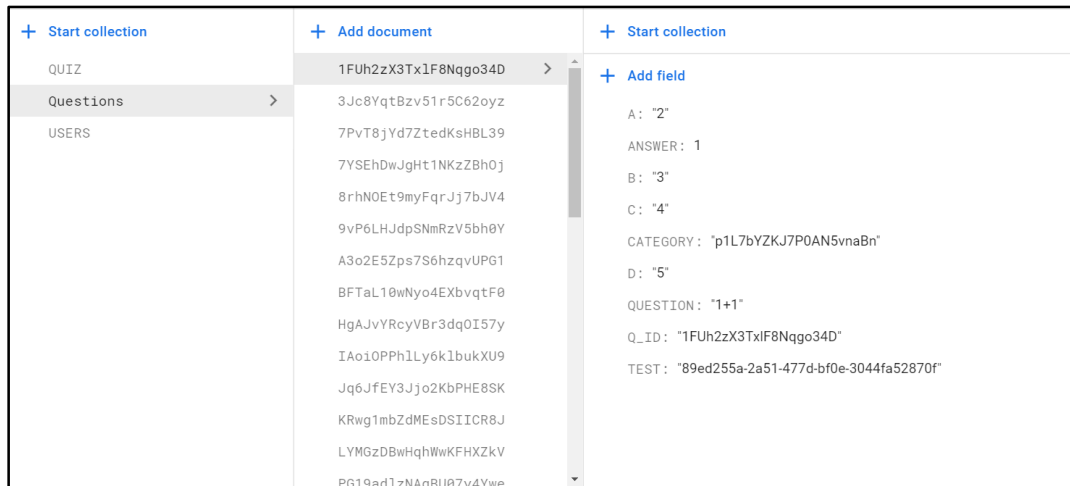
Figure 8: loadQuestion() function

Figure 9: Questions list in Firebase

The startTimer() function will start working when the user starts the quiz. The time limit for the quiz will be retrieved from the Firebase based on the specific test selected. The *CountDownTimer* will be based on the time limit of the quiz to start the countdown and the quiz will end when the countdown is equal to zero seconds. If the user manages to finish the quiz within the time limit, the time taken to finish the quiz will be recorded to the user profile. Figure 10 shows the code segment for startTimer() function.

```java
private void startTimer()
{
    long time = g_testList.get(selected_test_index).getTime()*60*1000;

    timer = new CountDownTimer( millisInFuture: time + 1000, countDownInterval: 1000) {
        @Override
        public void onTick(long millisUntilFinished) {

            timeLeft = millisUntilFinished;

            String remainingTime = String.format("%02d:%02d min",
                    TimeUnit.MILLISECONDS.toMinutes(millisUntilFinished),
                    TimeUnit.MILLISECONDS.toSeconds(millisUntilFinished) -
                            TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(millisUntilFinished)));

            timerTV.setText(remainingTime);

        }

        @Override
        public void onFinish() {
            Intent intent = new Intent( packageContext: QuestionsActivity.this, ScoreActivity.class);
            long total = g_testList.get(selected_test_index).getTime()*60*1000;
            intent.putExtra( name: "TIME_TAKEN", total);
            startActivity(intent);
            QuestionsActivity.this.finish();
        }
    };

    timer.start();

}
```

Figure 10: startTimer() function

When the user finishes the quiz and presses the submit button, the submitTest() function will start working on showing the *AlertDialog* to ask the user "Do you want to submit?". If the user presses the "Confirm" button, it will record the time taken to finish the quiz and also calculate the marks in percentage. The interface will be automatically navigated to score activity which showing the marks, time taken, and leader board. Figure 11 shows the code segment for submitTest() function.

```
private void submitTest()
{

    AlertDialog.Builder builder
            = new AlertDialog
            .Builder( context: QuestionsActivity.this);
    builder.setCancelable(true);

    View view = getLayoutInflater().inflate(R.layout.stop_test_dialog, root: null);

    Button cancel = view.findViewById(R.id.et_cancelB);
    Button confirm = view.findViewById(R.id.et_confirmB);
    TextView title = view.findViewById(R.id.et_title);
    TextView msg = view.findViewById(R.id.et_message);

    title.setText("Submit Test");
    msg.setText("Do You want to Submit ?");

    builder.setView(view);

    final AlertDialog alertDialog = builder.create();

    cancel.setOnClickListener((view) -> { alertDialog.dismiss(); });

    confirm.setOnClickListener((view) -> {

            timer.cancel();
            alertDialog.dismiss();
            Intent intent = new Intent( packageContext: QuestionsActivity.this, ScoreActivity.class);
            long total = g_testList.get(selected_test_index).getTime()*60*1000;
            intent.putExtra( name: "TIME_TAKEN",   value: total - timeLeft);
            startActivity(intent);
            QuestionsActivity.this.finish();
```
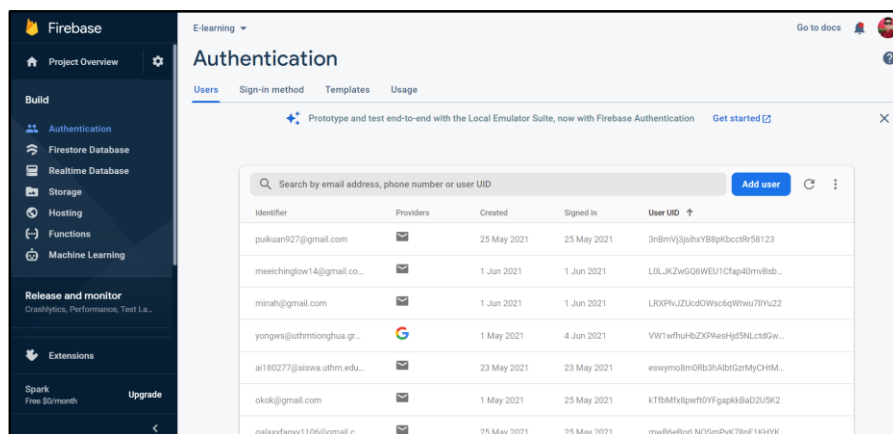
Figure 11: submitTest() function

3.4.3   Database Development (Google Firebase)

Google Firebase is a robust mobile backend-as-a-service that allows developers to develop mobile apps. A real-time database, user authentication, and hosting are the three main services provided by Firebase. In this project, user authentication services are mainly used to store user account data and authenticate the user when login. Figure 12 shows the authentication services provided by Google Firebase, whereas Figure 13 shows the database services from Google Firebase.

Figure 12: Authentication services from Google Firebase

According to Figure 12, authentication services is used to store user's email and password in Google Firebase. When the user successfully registers an account from MATH#f1, the Firebase Authentication services will create a new user account based on the registration data. Users able to login successfully when the log in detail is the same as the registration data in Firebase.
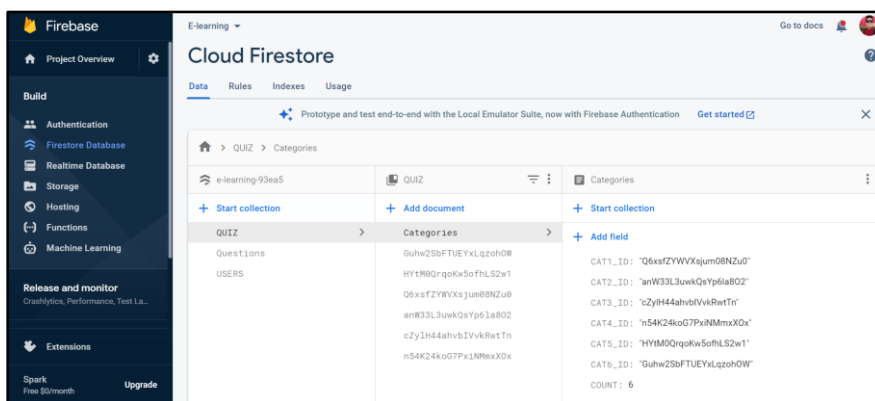
143

Figure 13: Database services from Google Firebase

Based on Figure 13, MATH#f1 is used Firestore Database to store quiz questions and user scores. Firestore Database can create a specific document for data collection with a unique ID. All the quiz questions will be stored in this section, and they can be retrieved by using the unique question ID.

3.5   Testing Stage

The subject of testing are the Form 1 students in SMK Seri Mutiara, Kuala Lumpur. The main purpose of the testing stage is to measure the functionality and usability of the application towards the target users and to make sure the objective of this application is achieved. Due to the Covid-19 pandemic, the beta testing was conducted through google meet with the students and teachers. The feedback from the target users would be taken into consideration for future improvement.

4.   Result and Discussion

This section discussed the fully functioning application used for testing process, and the analysis of the testing result.

4.1   Application testing

The testing process begins when the complete application is fully implemented and ready to run. The testing process is important to make sure the complete application can function well on the user side. In order to make sure that the application is working well, alpha and beta testing are conducted. The developer carried out alpha testing to test the entire application using black-box testing, whereas beta testing was done by involving the targeted users to perform user acceptance testing.

4.1.1   Alpha Testing (Black-Box Testing)

Alpha testing was carried out by the developer using black-box testing to identify all the possible bugs and issues before releasing the application to the end users. The black-box testing is required to test the functionality of each button in each interface. Once errors are found during the testing, the errors will be fixed immediately to ensure the functionality and usability of the application is in an optimum stage. Table 3 to Table 7 show the black-box testing and the result of each test case.

Table 3: Black-box testing for Main interface

Test case	Expected Result	Actual Result
Login button	Navigates to the Login Module	Works well as expected
Start button	Navigates to the Menu interface	Works well as expected
App Guide button	Navigates to the App Guide interface	Works well as expected
Exit button	Show exit warning when clicked	Works well as expected

Table 3: Black-box testing for Main interface (cont)

Create Account Button	Navigates to the Signup Module	Works well as expected

Table 4: Black-box testing for Learning Module

Test case	Expected Result	Actual Result
YouTube video button	Able to play a video with sound	Works well as expected
YouTube icon button	Navigates to the YouTube app and able to play full screen video	Works well as expected

Table 5: Black-box testing for Formula Module

Test case	Expected Result	Actual Result
Tab layout button	Able to switch chapter when clicked	Works well as expected

Table 6: Black-box testing for Practice Module

Test case	Expected Result	Actual Result
Check Answer button	Able to check correct answer when clicked	Works well as expected
Hints button	Able to check hints when clicked	Works well as expected
Image enlarge	Able to enlarge the image when double tap	Works well as expected

Table 7: Black-box testing for Quiz Module

Test case	Expected Result	Actual Result
Home button	Navigates to the Home interface in the quiz.	Works well as expected
Leaderboard button	Navigates to the Leaderboard interface in the quiz	Works well as expected
My Account button	Navigates to the User Account interface in the quiz.	Works well as expected
Saved Questions button	- Navigates to the Save Questions interface - Able to view questions that were saved during the quiz.	Works well as expected
My Profile button	Navigates to the User Profile interface.	Works well as expected
Logout button	Able to logout when clicked	Works well as expected
Chapter button	Navigates to the Test selection interface	Works well as expected
Test button	Navigates to the Test info interface	Works well as expected
Start button	Navigates to the Quiz interface	Works well as expected
Option button	Able to choose within four options	Works well as expected
Clear Selection button	Able to clear the current selection	Works well as expected
Mark for Review button	Able to mark the current question for further review	Works well as expected
Next button	Navigates to the next question	Works well as expected

Table 7: Black-box testing for Quiz Module (cont)

Submit button	Able to submit the quiz when clicked	Works well as expected
Reattempt button	Able to reattempt the quiz again	Works well as expected

4.2.2 Beta Testing (User Acceptance Testing)

Beta testing is the final stage of testing before releasing the apps to a wide audience. Beta testing aims to get feedback from the target user based on their experience in using the MATH#f1 and uncover as many bugs or usability issues as possible before publishing the app. The target users are Form 1 student in SMK Seri Mutiara at Kuala Lumpur. Due to the Covid-19 pandemic, the beta testing was conducted through Google Meet with the students and teacher. At the end of the meeting, a set of questionnaires prepared on Google Form are sent to students and teachers to answer. A total number of 25 respondents are involved in the beta testing divided into three categories: 12 Form 1 student, 1 teacher, and 12 random people.

The questionnaires consist of two sections which are focus on user acceptance level and functionality of the app. User acceptance level is depending on the satisfaction of the user after testing the app. The questions set on user acceptance level are based on ease of use, suitability, user interface design, and knowledge gain from using the app. On the other hand, the questions for the app's functionality mainly focus on each module.
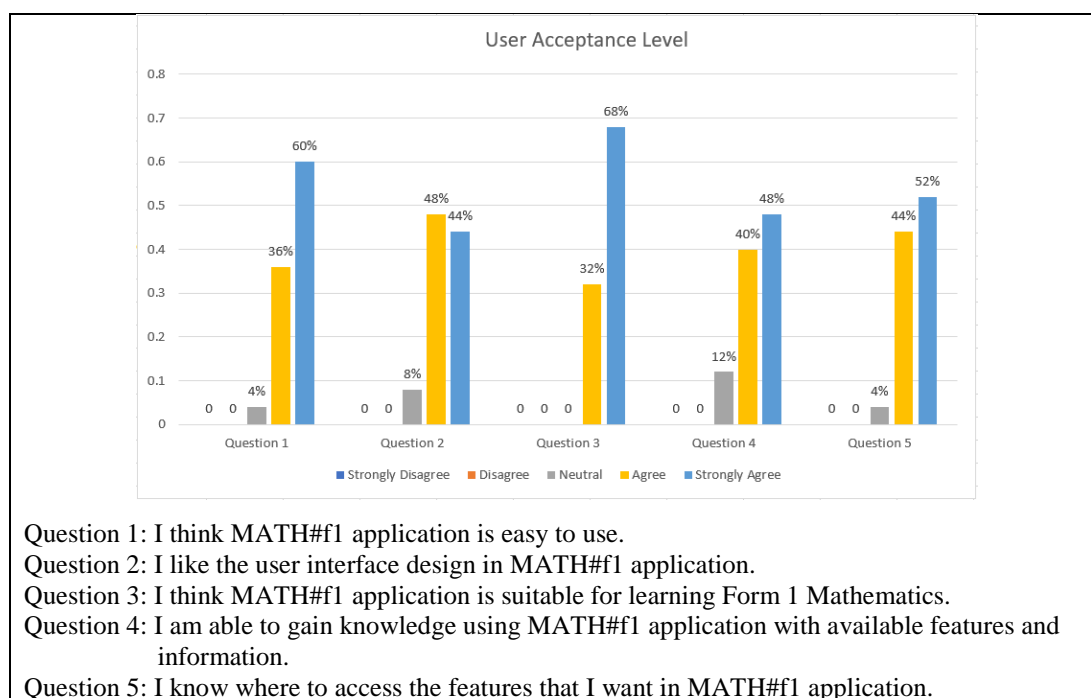

Question 1: I think MATH#f1 application is easy to use.
Question 2: I like the user interface design in MATH#f1 application.
Question 3: I think MATH#f1 application is suitable for learning Form 1 Mathematics.
Question 4: I am able to gain knowledge using MATH#f1 application with available features and information.
Question 5: I know where to access the features that I want in MATH#f1 application.

Figure 14: Analysis of user acceptance level

As shown in Figure 14, there are five questions to determine the user acceptance level. From the chart generated, more than 80% of respondents agree with the question asked about the app. For Question 1, 96% of respondents think that the MATH#f1 application is easy to use, while only 4% of respondents are neutral. For Question 2, 92% of respondents liked the user interface design in the MATH#f1 application. For Question 3, all respondent thinks that the MATH#f1 application is suitable for learning Form 1 Mathematics. For Question 4, 88% of respondents think they can gain knowledge using the MATH#f1 application with available features and information. Finally, for Question 5, 96%

of users know where to access the MATH#f1 application features. As a result, most of the respondents provided positive feedback in the user acceptance testing.
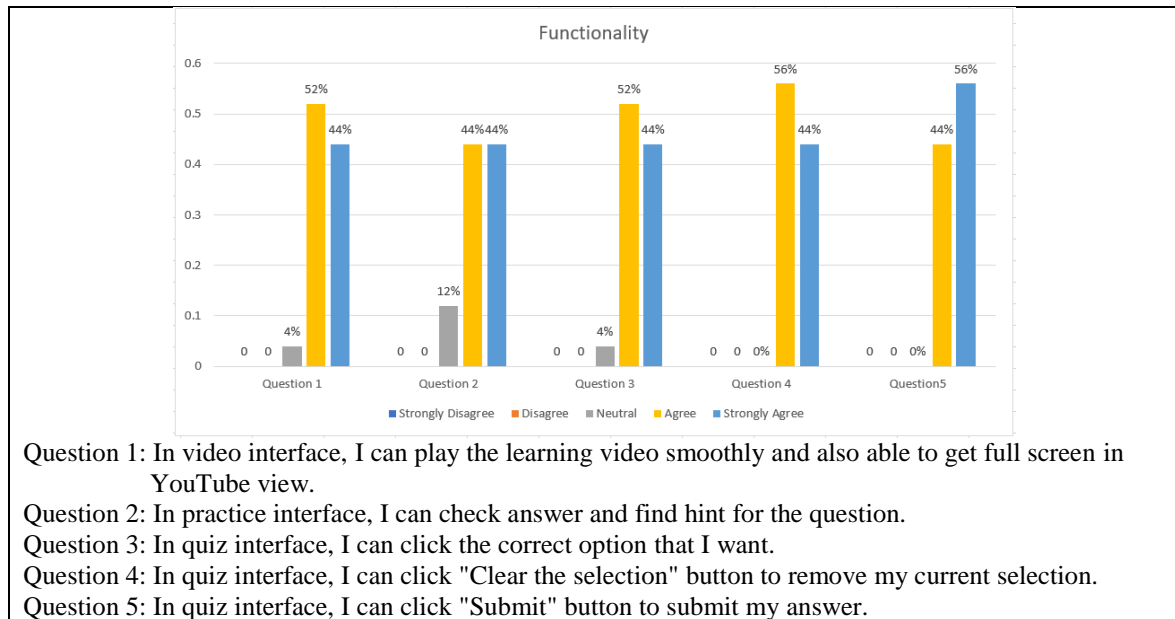

Question 1: In video interface, I can play the learning video smoothly and also able to get full screen in YouTube view.
Question 2: In practice interface, I can check answer and find hint for the question.
Question 3: In quiz interface, I can click the correct option that I want.
Question 4: In quiz interface, I can click "Clear the selection" button to remove my current selection.
Question 5: In quiz interface, I can click "Submit" button to submit my answer.

Figure 15: Analysis of functionality

Based on Figure 15, there are five questions to determine the functionality of the app from the users. For Question 1, total of 96% of respondents can play the learning video smoothly and get full screen in YouTube view. For Question 2, a total of 88% of respondents can check the answer and find the hint for the question. For Question 3, a total of 96% of respondents able to click the correct option in the quiz interface, while only 4% of respondents not sure about it. Finally, for Question 4 and Question 5, all respondents are able to click the "Clear the selection" button to remove the current selection and click the "Submit" button to submit the answer. Thus, the results show that the functionality of MATH#f1 is in the optimum stages.

Besides, System Usability Scale (SUS) has been used for the usability test to calculate the average score of user acceptance [11]. The test was performed by conducting feedback via Google Form to 12 students as selected respondents with five questions per section. The questionnaire contains five positive and five negative statements. Figure 16 shows the analysis of positive questions, whereas Figure 17 shows the analysis of negative questions.

Question 1: I think that I would like to use this application frequently.
Question 3: I thought the application was easy to use.
Question 5: I found the various functions in this application were well integrated.
Question 7: I would image that most people would learn to use this application very quickly.
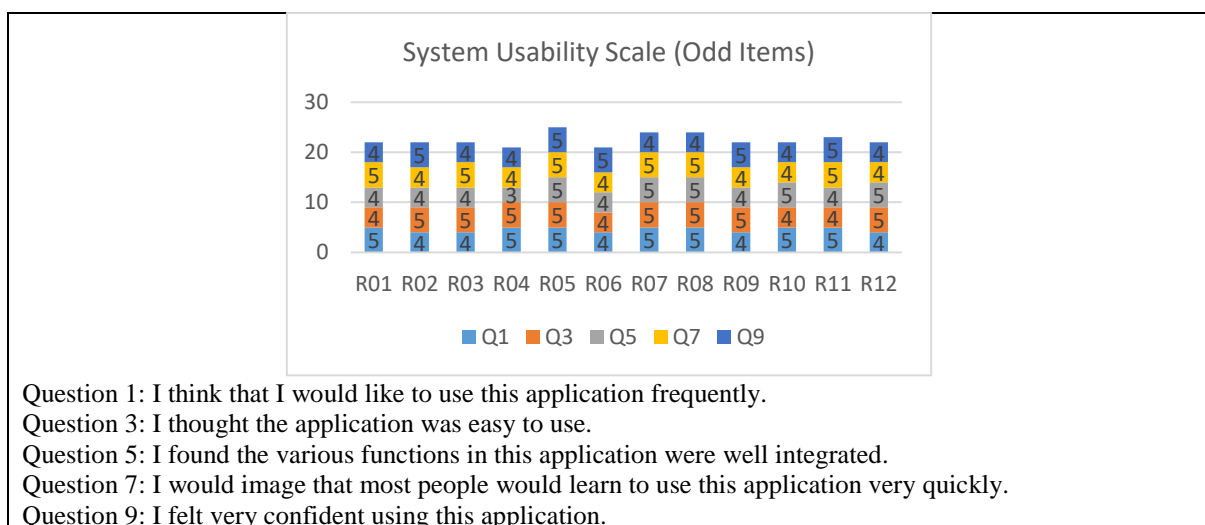Question 9: I felt very confident using this application.

Figure 16: Analysis of positive questions

Based on Figure 16, all respondents gave 4 marks and above, which agrees to the positive statement in questionnaires. Although some of the questions are obtained 3 marks which is neutral but none of them disagree with the positive outcome.
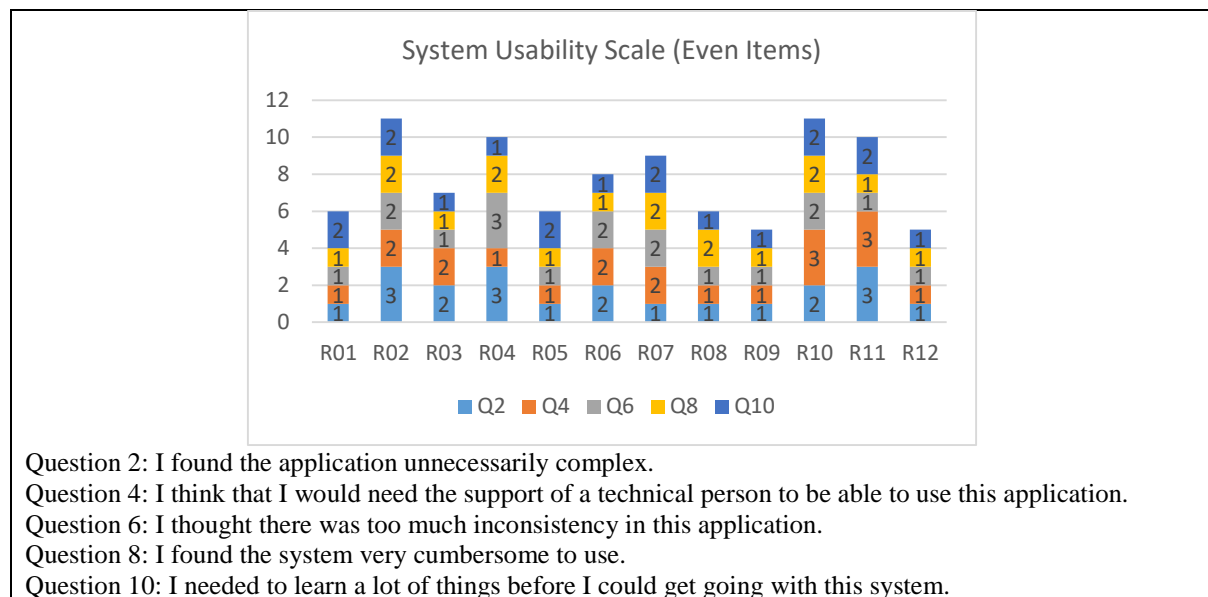

Question 2: I found the application unnecessarily complex.
Question 4: I think that I would need the support of a technical person to be able to use this application.
Question 6: I thought there was too much inconsistency in this application.
Question 8: I found the system very cumbersome to use.
Question 10: I needed to learn a lot of things before I could get going with this system.

Figure 17: Analysis of negative questions

Based on Figure 17, all the respondents gave marks in between 1 to 3, which disagrees with the negative statement in questionnaires. The result shows that most respondents do not have much problem using the MATH#f1 application.

The total scores for each question from the user tests were shown in Table 8 in detail, followed by calculating the average score by using the formula.

Table 8: Respondent's Score (User Acceptance Test)

Respondent	Item Score										TOTAL SCORE
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	
R01	5	1	4	1	4	1	5	1	4	2	90.0
R02	4	3	5	2	4	2	4	2	5	2	77.5
R03	4	2	5	2	4	1	5	1	4	1	87.5
R04	5	3	5	1	3	3	4	2	4	1	77.5
R05	5	1	5	1	5	1	5	1	5	2	97.5
R06	4	2	4	2	4	2	4	1	5	1	82.5
R07	5	1	5	2	5	2	5	2	4	2	87.5
R08	5	1	5	1	5	1	5	2	4	1	95.0
R09	4	1	5	1	4	1	4	1	5	1	92.5
R10	5	2	4	3	5	2	4	2	4	2	77.5
R11	5	3	4	3	4	1	5	1	5	2	82.5
R12	4	1	5	1	5	1	4	1	4	1	92.5
Average Score											86.67

Table 8 shows the total score of usability value of the MATH#f1 application that has been evaluated. The next part will show how the average score of usability value calculated according to the prescribed formula of the test performed.

The formula used to obtain usability results based on the SUS are:

Total score = (odd items + even items) x 2.5

$$\text{Average score} = \frac{\text{Total score}}{\text{Total respondents}}$$

Where:

Odd items (Q1, Q3, Q5, Q7, Q9) = 5 – contribution

Even items (Q2, Q4, Q6, Q8, Q10) = contribution - 1

Therefore:

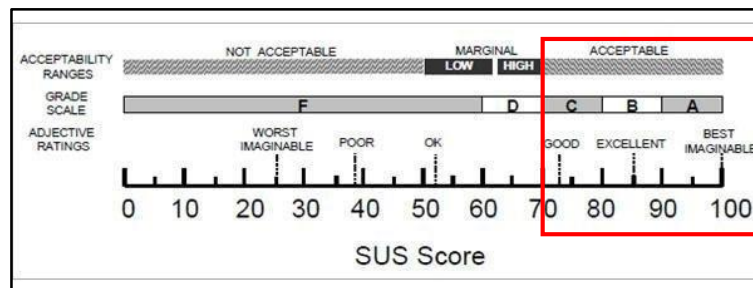$$\text{Average score} = \frac{90+77.5+87.5+77.5+97.5+82.5+87.5+95+92.5+77.5+82.5+92.5}{12}$$

$$= 86.67$$

Figure 18: System Usability Scale (SUS)

Based on the SUS score scale shown in Figure 18, the average score of the usability value is 86.67 which in the range of Acceptable in the score scale. Overall, the built-in applications can be classified as successfully meet the needs of the target users.

5. Conclusion

In conclusion, the MATH#f1 application was successfully developed with four main modules, namely Learning, Formula, Practice and Quiz. Besides, the usability testing obtained the SUS score of 86.67 which is within the acceptable range based on the System Usability Scale (SUS).

All three objectives set from the beginning of the project have been achieved. The first objective was achieved when all the design phases were completed based on the user requirement. Next, the second objective was achieved when the developed application was successfully installed into an Android device. Lastly, the third objective was achieved when the user acceptance test results were obtained from the target user.

MATH#f1 application has several advantages, such as the content has been verified by Form 1 Mathematics teacher as the subject matter expertise. Therefore, the application has the ability to improve student's conceptual understanding and basic Mathematic skills. Besides, the MATH#f1 application provided a list of complete formulas based on each chapter and each subtopic, and therefore, the student could get the correct formula with just a few clicks. MATH#f1 application also provided a quiz system, and it can be a short test for students to test their knowledge.

Apart from its advantages, the MATH#f1 application has a number of flaws that were discovered during user testing. There are 13 chapters for the Form 1 Mathematics syllabus, but the MATH#f1 application only covered 6 chapters. In the practice module, users can only check the hints and answer but unable to check the solution of the question. In the quiz module, limited questions for each chapter in the current version.
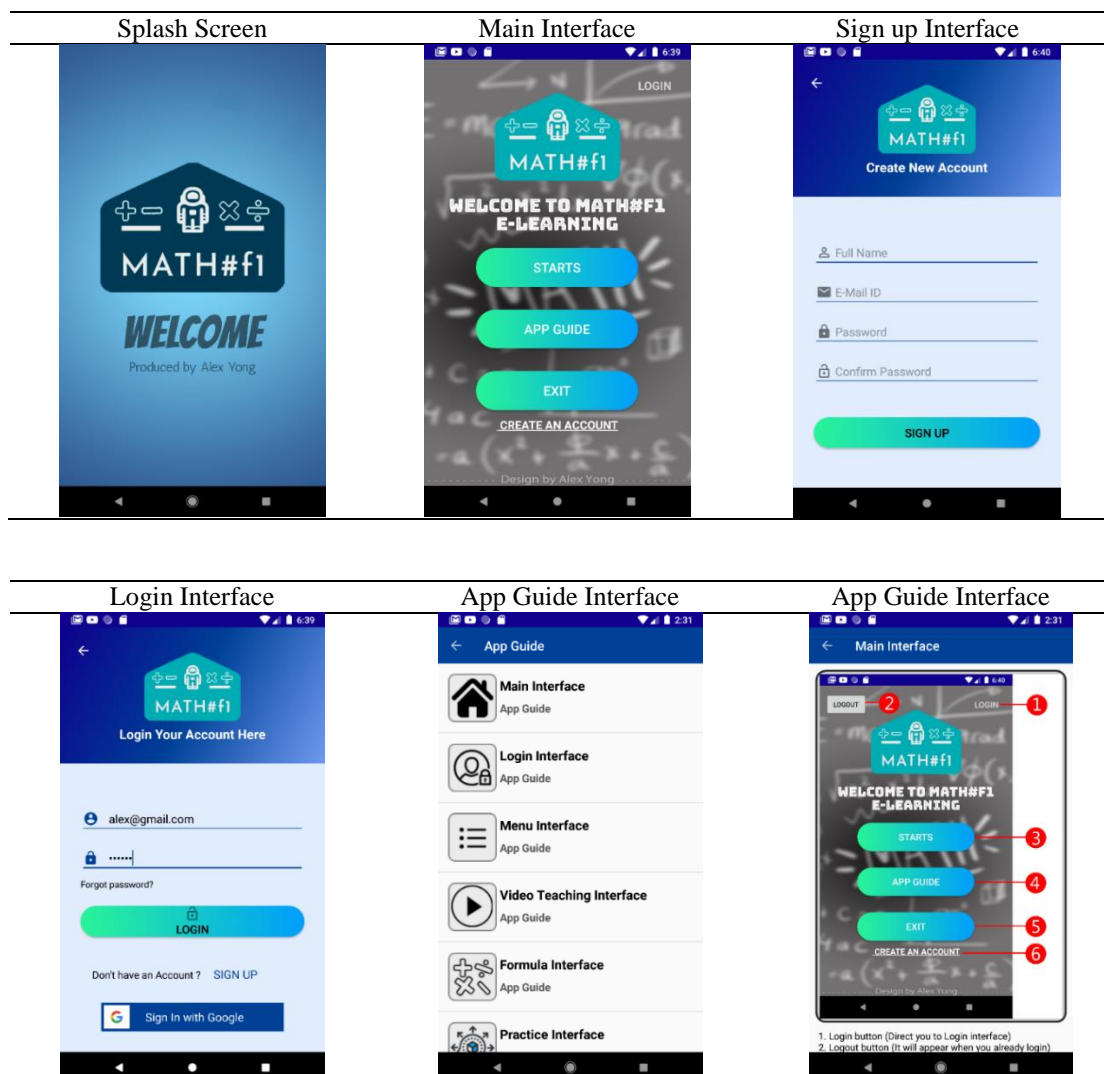
Improvements that can be made in future to resolve the current limitations are to cover all chapters for the Form 1 Mathematics syllabus in the application. In the practice module, provide a solution for each question to guide the user on solving it step by step. In the quiz module, provide more different questions for the quiz in each chapter.
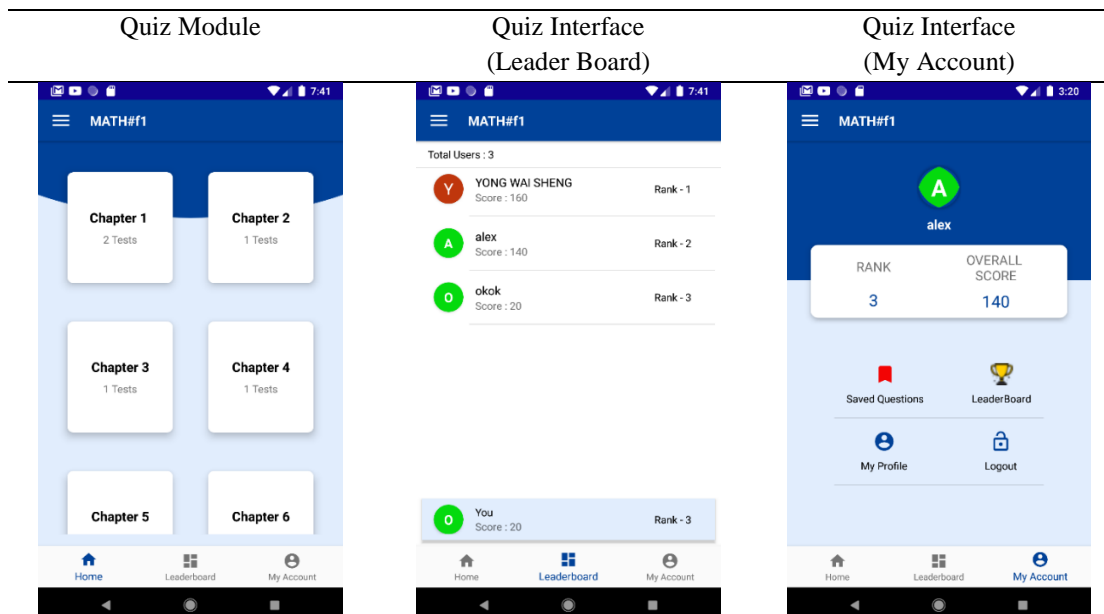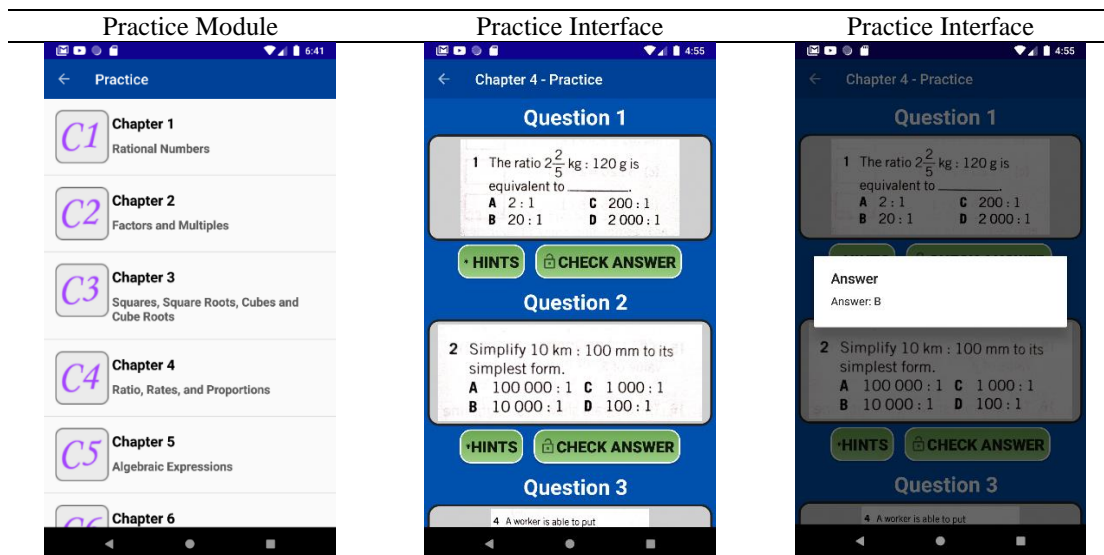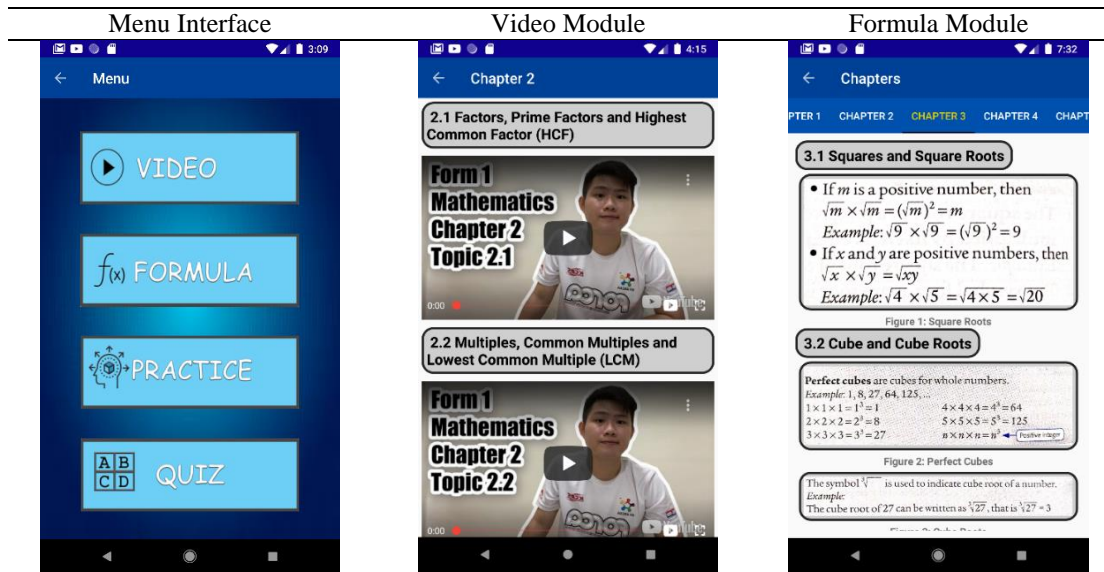
To sum up, the MATH#f1 application has successfully developed by following the well-planned and organized methodology and gained valuable feedback from user testing. All three objectives of this project have been achieved, and it is hope that this learning application can be use in a secondary school as a teaching and learning aid.

Acknowledgement

Appendix A

Splash Screen	Main Interface	Sign up Interface

Login Interface	App Guide Interface	App Guide Interface

Menu Interface	Video Module	Formula Module

Practice Module	Practice Interface	Practice Interface

Quiz Module	Quiz Interface (Leader Board)	Quiz Interface (My Account)

References

[1]   A. S. Drigas and M. A. Pappas, "A review of mobile learning applications for mathematics," *Int. J. Interact. Mob. Technol.*, vol. 9, no. 3, pp. 18–23, 2015, doi: 10.3991/ijim.v9i3.4420.

[2]   R. Alhassan, "Mobile Learning as a Method of Ubiquitous Learning: Students' Attitudes, Readiness, and Possible Barriers to Implementation in Higher Education," *J. Educ. Learn.*, vol. 5, no. 1, pp.1-10, Jan. 2016, doi: 10.5539/jel.v5n1p176.

[3]   "Benefits of Using Mobile Apps in Education." https://www.mobileappdaily.com/benefits-of-using-mobile-apps-in-education (accessed Oct. 19, 2020).

[4]   K. A. Blotnicky, T. Franz-Odendaal, F. French, and P. Joy, "A study of the correlation between STEM career knowledge, mathematics self-efficacy, career interests, and career activities on the likelihood of pursuing a STEM career among middle school students," *Int. J. STEM Educ.*, vol. 5, no. 1, pp.1-15, 2018, doi: 10.1186/s40594-018-0118-3.

[5]   "Part One: Connecting Mathematics with Work and Life | High School Mathematics at Work: Essays and Examples for the Education of All Students | The National Academies Press." https://www.nap.edu/read/5777/chapter/3 (accessed Jul. 01, 2021).

[6]   "Mobile Learning: Advantages And Disadvantages - eLearning Industry." https://elearningindustry.com/mobile-learning-advantages-disadvantages (accessed Jul. 01, 2021).

[7]   "Matematik Tingkatan 1 - Apps on Google Play." https://play.google.com/store/apps/details?id=com.angelworks.mathtingkatan1&hl=en_US&gl=US (accessed Nov. 23, 2020).

[8]   "Mathematics Form 1 Notes+Topical Revisions - Apps on Google Play." https://play.google.com/store/apps/details?id=com.dev.mox.math1&hl=en_US&gl=US (accessed Nov. 23, 2020).

[9]   "FORM ONE MATHS NOTES [KLB] + FREE TEACHERS GUIDE - Apps on Google Play." https://play.google.com/store/apps/details?id=com.the.form_1_4_kcse_mathematics_notes&hl=en_US&gl=US (accessed Nov. 23, 2020).

[10]   W. Sazli, N. Saifudin, S. Salam, M. Haziq, and L. Abdullah, "Multimedia Mobile Content Development Framework and Methodology for Developing M-Learning Applications," *J. Tech. Educ. Train.*, vol. 4, no. 1, pp. 15–22, 2012.

[11]   J. Brook, "SUS: a retrospective," *J. Usability Studies*, vol. 8, no. 2, pp. 29–40, 2013.