



**UTM**  
UNIVERSITI TEKNOLOGI MALAYSIA

**SCHOOL OF COMPUTING**  
Faculty of Engineering

**UNIVERSITI TEKNOLOGI MALAYSIA**  
**FINAL EXAMINATION SEMESTER II, 2021/2022**

**SUBJECT CODE : SECJ3623**

**SUBJECT NAME : MOBILE APPLICATION PROGRAMMING**

**TIME / DURATION : 10:00 AM / 4 HOURS**

**DATE/DAY : 17 JULY 2022 / SUNDAY**

---

**GENERAL INSTRUCTIONS**

- This is an **OPEN-BOOK** examination. You can refer to any resources online or offline.
- However, any kind of discussions whether verbal, chats, online, etc. is strictly prohibited.
- This is a practical examination. You will attempt the examination by doing coding on a computer.
- You will use eLearning only for downloading the question and submitting the answers.
- You will do coding offline using an IDE such as VS Code.
- All comments you write in the program will not be graded.
- Programs that **CANNOT RUN** will get **50% PENALTY**.
- No question about the examination question will be entertained. Read the question carefully.
- No help or any technical support is given regarding issues related to the Flutter SDK, the IDE, emulator, git, Firebase, etc. Using these tools is part of the practical examination.

**PLAGIARISM WARNING**

- Be warned that your submission will undergo a screening process for **PLAGIARISM**.
- Programs that are suspected cheating will be **rejected**, thus will get a **ZERO** mark.  
This includes both parties, i.e., the one **who copies** and the one **who gives copy**.

## VIDEO RECORDING INSTRUCTIONS

- Some parts of the examination (i.e., Part A and Part C) require deliverables in videos.
- You can use the Webex meeting software to do the video recording.

### *Notes about Webex Meeting:*

- A free account Webex only opens for 50 minutes per session. Thus, should you need more time, you will need to open another session once the current one ends.
  - Free account Webex also allows only local recording, i.e. the video will be stored on your computer. Thus, later you will need to upload the videos to the cloud (e.g., to Google Drive) manually.
- Alternatively, you can do the recording using software like Open Broadcaster Software (OBS) or any other tools.
  - However, you **cannot use a livestream** (such as YouTube Live) to do the recording.
  - Later during the submission, upload all the recorded video files to your Google Drive account (or any other online storage). Put them in a single folder, and submit only the folder's link. Set the video file (or folder) permissions to "**Anyone can view**".
  - You do not need to edit the videos at all. Submit them as is.
  - Alternatively, you may upload the videos to YouTube, set the permission to "**Unlisted**", and submit all the video links.

## SUBMISSION INSTRUCTIONS

- This examination requires two types of submissions:
  - a) Source code:
    - This submission applies only to **Part B**.
    - to be submitted twice:
      - Interim submission at **12:00 PM**, will be open for 10 minutes.
      - Final submission at **2:10 PM**, will be open for 10 minutes.
    - submit in a **git bundle** file.
    - Notes: **You are NOT expected to submit or push anything to Github.**
  - b) Recorded video:
    - This submission applies to **Part A** and **Part C**.
    - You will upload the videos outside of the examination session. You submit the video link to eLearning by latest **18 July 2022, 4:00PM**.
    - Upload the videos to your own Google Drive account, put them in a folder and submit only the folder link to UTM eLearning.
    - Alternatively, upload the videos to YouTube and submit the video links.

## Question

You will develop a Note application for mobile devices that provides three basic operations: adding new notes, viewing notes and editing existing notes. The application will be developed using Flutter framework for the front-end and using Firebase as the backend. Regarding Firebase, the application will be utilizing Live Firebase, using two Firebase services: Firebase Authentication and Firestore.

This examination comes with a starter project which includes the following source codes:

- three screens, `login_screen.dart`, `home_screen.dart` and `edit_screen.dart`
- a model class, `note.dart`.
- the main program, `main.dart`.

**You MUST USE the starter project.** For example, when dealing with data, **you must use the model class instead of directly using map data.** You may modify, reorganize or refactor the starter codes. You may also add other codes whenever necessary.

Further, some packages have already been added to the `pubspec.yaml` including `firebase_core`, `cloud_firestore` and `firebase_auth`. **You may add other packages whenever necessary.**

The tasks for this examination are divided into three parts, Part A, B and C. **Part A and B have to be done and submitted during the examination session while Part C is done after the examination session.** The question and details for each part is described in the following section.

## Part A: Setting Up Firebase and Flutter

[15 Marks]

**Instruction:** Record this part in a video. You are expected to explain what you are currently showing in the video. It is not necessary to turn on the camera. You must use Live Firebase services instead of Firebase Emulator. You do not need to use Git yet in this part.

Create a new Firebase project from the firebase console

(<https://console.firebase.google.com>) and accomplish the following tasks:

Task A1. Enable Firebase Authentication service with “**Sign in with email and password**” authentication provider, and create some user accounts. Use the sample accounts given in the file `sample_data.txt`.

(5 marks)

Task A2. Enable Firestore service, and create some data for each user. See the same file above for sample data.

(5 marks)

Task A3. Configure your Flutter project so that it can connect to the Firebase project.

(5 marks)

## Part B: Coding

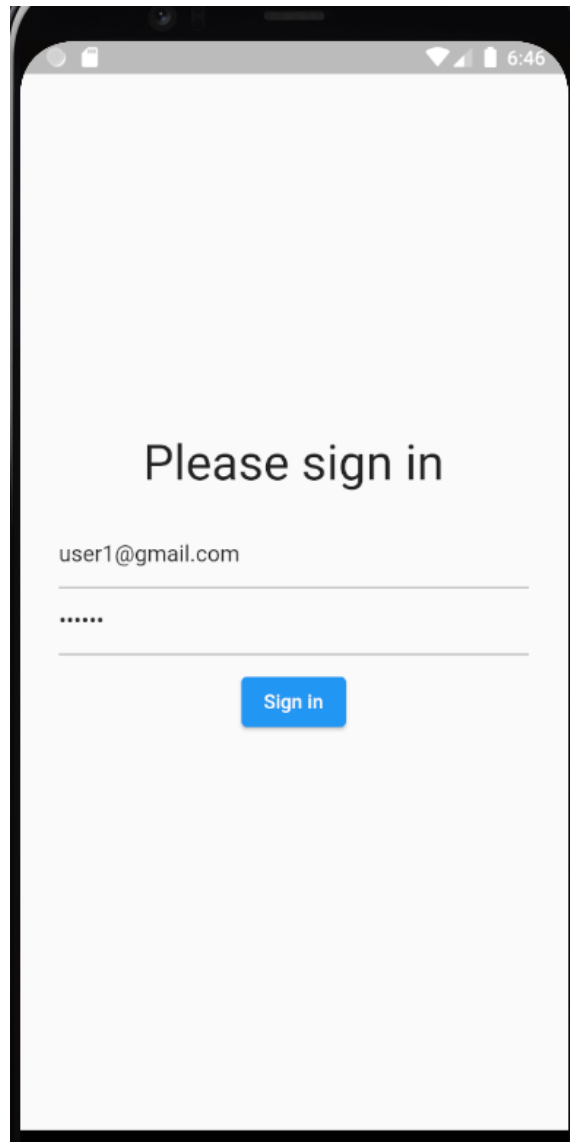
[75 Marks]

**Instruction:** You do not have to record this part. You are expected to use Git in this part.

Each task in this part must have its own git commit. Include the task number for the commit message, for example, “Task B1 completed”. You will perform git commit based on the main tasks. For example, in Task B1 (see below), which consists of 2 sub-tasks (i and ii), you will perform only ONE commit, and it should be done after the sub-task (ii) completed. That means, you will perform NINE (9) git commits in total for this part. **Each git commit has its own mark and it is inclusive in the task.**

Details of each task and the assessment mark are given in the following sub-sections. They are TEN (10) tasks in this part, numbered from Task B1, Task B2, and so on.

## The login screen



**Figure 1:** The login screen

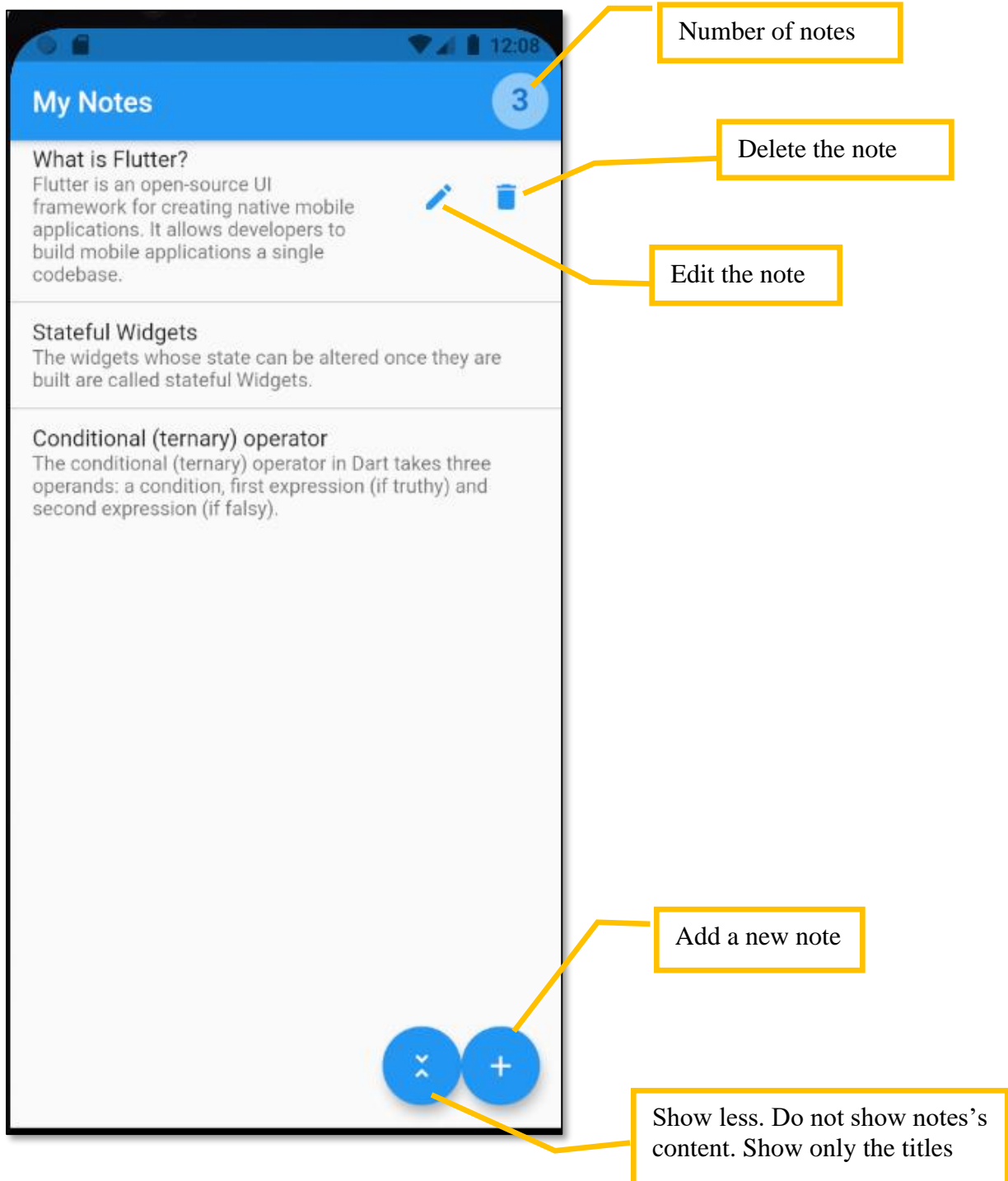
Task B1. User can sign-in on the login screen using his/her account.

- i. The user can only proceed to the next screen (i.e., the home screen) if the sign-in is successful.
- ii. If the sign-in is unsuccessful, the user will remain at the login screen. You do not need to perform any other validations.

*See the **video b1** for the expected result of this task.*

(8 marks)

## The main or home screen



**Figure 2:** The main screen


Task B2. Showing the user's notes on the home screen.


- i. The home screen will show only the notes that belong to signed in user.  
The information shown for each note includes the note's title and content.
- ii. Besides, the main screen will also show the number of notes that belongs to the user at the screen bar.

*See the **video b2** for the expected result of this task.*

(8 marks)

Task B3. Showing and hiding the notes' content:

- i. When the “**Show Less**” button  is tapped on, it will hide the contents of all notes from being displayed, and leaving only the notes' titles. Also, the button icon will change to the “Show More” icon.

- ii. When the “**Show More**” button  is tapped on, it will show the notes' contents and the button icon will turn back to the “Show Less” icon.

*See the **video b3** for the expected result of this task.*


(7 marks)

Task B4. Turning on and off the editing tools:

- i. When a note title is long-pressed , it will show the editing tools (which consists of Edit and Delete buttons) for the note.
- ii. Long-pressing on the same note title again will hide the editing tools.
- iii. Long-pressing on another title will show editing tools on the title, and hide it from the previous title. That means, the editing tool only visible on one title at a time.



*See the **video b4** for the expected result of this task.*

(10 marks)

Task B5. When the **Delete** button  on a note is tapped on, it will delete the note from the database and reflect the changes on the user interface accordingly.  
*See the **video b5** for the expected result of this task.*

(4 marks)

Task B6. Opening the second screen:

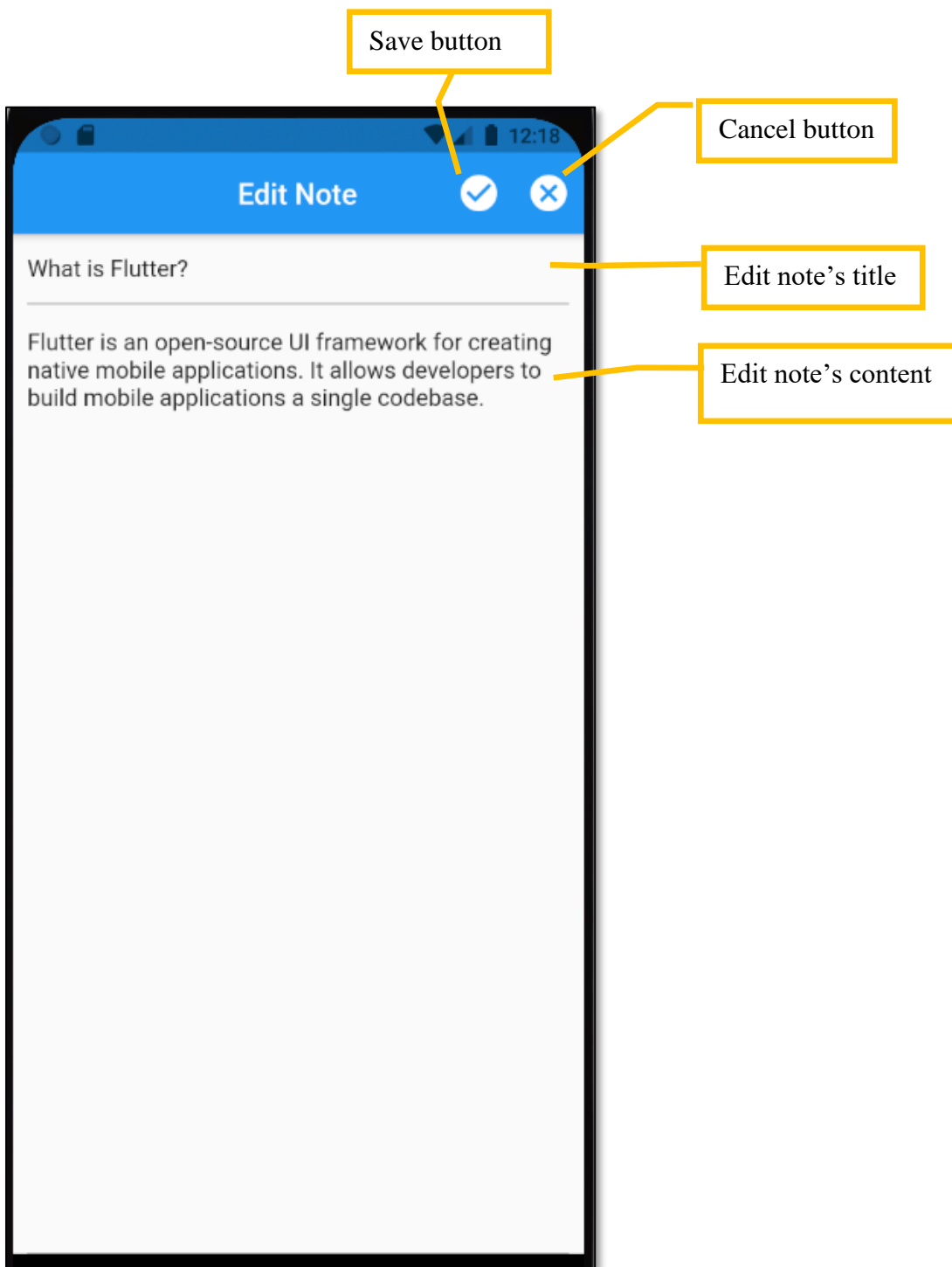
- i. When a note title is tapped on, it will open the second screen for the note in “View” mode.
- ii. When the **Edit** button  on a note is tapped on, it will open the second screen in “Edit” mode.
- iii. When the **Add** button  at the bottom of the screen is tapped on, it will open the second screen in “Add” mode. This will add a new note to the list and reflected in the database.
- iv. All the above operations (i)-(iii) are achieved by making use of the same Edit screen.

*See the **video b6** for the expected result of this task.*

(14 marks)



## The edit screen



**Figure 3:** The edit screen

Task B7. Showing the screen bar according to the operation mode:

- i. If this screen is opened for view (from B6(i) above), the screen bar will have the title “**View Note**” and only the **Cancel** button.
- ii. If this screen is opened for editing a note (from B6(ii) above), the screen bar will have the title “**Edit Note**” and both the **Save** and **Cancel** buttons.
- iii. If this screen is opened for adding a new note (from B6(iii) above), the screen bar will have the title “**Add new Note**” and both the **Save** and **Cancel** buttons.

*See the **video b7** for the expected result of this task.*

(10 marks)



Task B8. Show only or allow editing the note’s data based on the operation mode:

- i. If this screen is opened for view (from B6(i) above), the user can **only view** the note’s data (i.e., title and content). The data cannot be edited at all.
- ii. If this screen is opened for edit or add (from B6(ii) and (iii) above), The user is allowed to edit the note’s data.

*See the **video b8** for the expected result of this task.*

(7 marks)

Task B9. Confirming or cancelling changes:

- i. When the **Save** button  is tapped on, it will bring the user back to the main screen. This operation also saves the note to the database and reflect the changes on the UI of the main screen accordingly.
- ii. When the **Cancel** button  is tapped on, it will cancel any changes and bring the user back to the main screen.

*See the **video b9** for the expected result of this task.*

(7 marks)

## Part C: Showcasing

[10 Marks]

**Instruction:** You do this part outside of the examination session. Record this part in a separate video. You are expected to explain what you are showing in the video. It is not necessary to turn on the camera. You **MUST NOT** change at all the code you have done in previous parts.

Prepare a **15-minute** video that showcases the app you developed in previous parts. **You are not allowed to do any modification to your code.** Should you showcase a project that is different from the code you did in previous parts, it will be considered cheating and thus will be rejected automatically.

You will showcase the app from the end user perspective. Also, you should show that any changes on the app will be **reflected in the backend** (i.e., in Firestore). In this regard, arrange your emulator and Firebase console side-by-side. **Showcase the result of Part B, i.e., from Task B1 to Task B9.**

(10 marks)