



UNIVERSITI MALAYSIA TERENGGANU

CSM3114

Framework Based Mobile Application Development

PROJECT 2 REPORT

SCHOOL MANAGEMENT APP

SEMESTER I 2023/2024

NAME : AKMAL HISYAM BIN AHMAD FARIS

MATRIC NO : 63689

PROGRAMME : BACHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING)

LECTURER : DR. MOHAMAD NOR BIN HASSAN

Table of Content

GitHub Link.....	3
Executive summary	3
Use Case.....	4
The common structure of tree widgets you used when designing and developing the application	5
Flutter widget and features adopted in the application.....	7
UI for the application with explanation	9
Conclusion.....	12
Reference	13

GitHub Link

https://github.com/Hisye/school_management_app.git

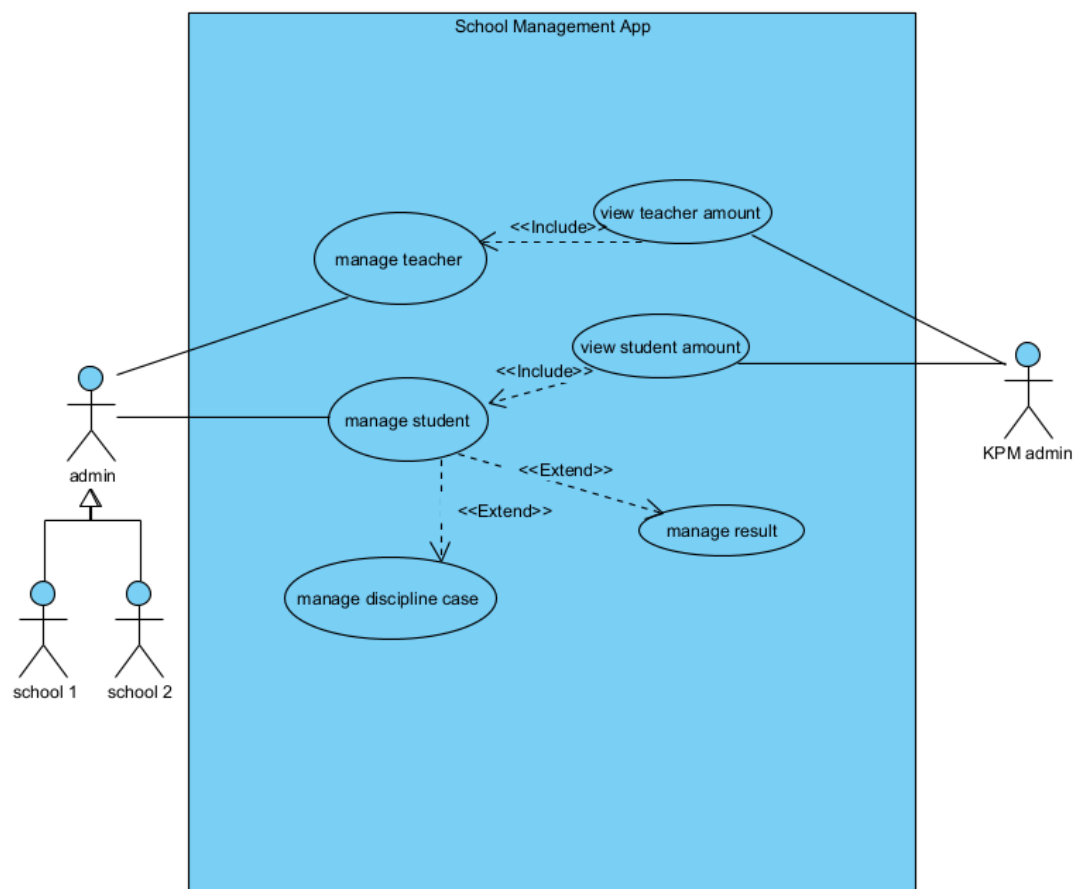
Executive summary

The School Management App, developed using Flutter, serves as a comprehensive solution to streamline administrative tasks in educational institutions. With a user-friendly interface, the application empowers administrators to efficiently manage both student and teacher data. Administrators can seamlessly add, modify, and remove student and teacher information, ensuring accurate and up-to-date records for both.

The app incorporates dynamic search functionality based on matriculation numbers, enabling administrators to quickly retrieve specific details for both students and teachers. Additionally, a robust disciplinary module allows administrators to track and manage disciplinary cases effectively for students. The integration of Firebase Realtime Database ensures secure storage and retrieval of both student and teacher data, contributing to the app's reliability.

Administrators can stay informed about disciplinary cases, access student and teacher profiles, and oversee the overall database. The app's Material Design, inspired by Flutter, offers a visually appealing and user-friendly experience for administrators. Features like dismissible tiles and snackbar notifications enhance the usability of the application, search icon to handle big data easier, and theme changer making it an essential tool for efficient administrative management in educational institutions.

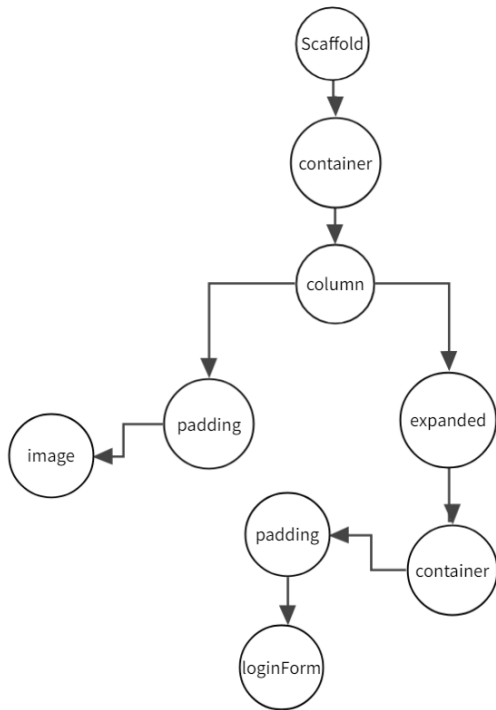
Use Case



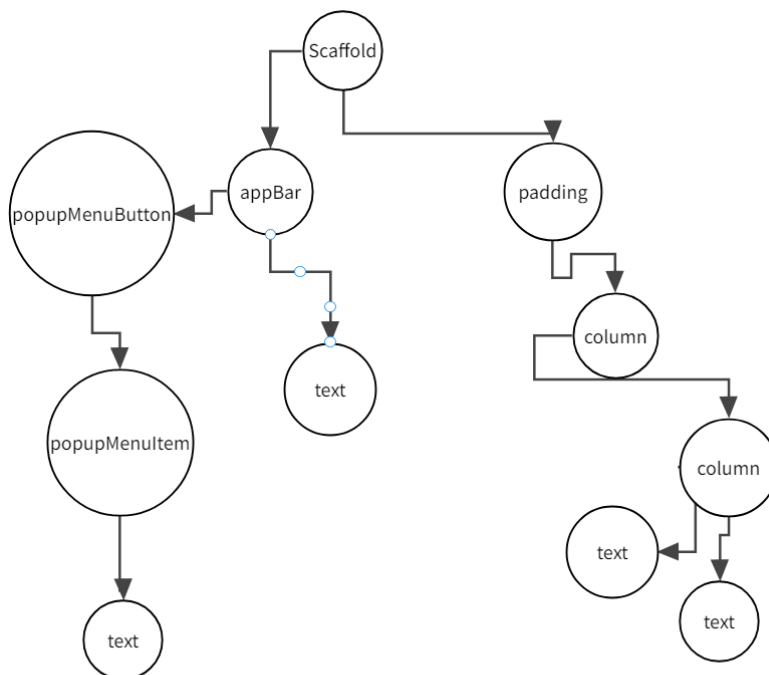
The use case diagram is for School Management App. The app would be used by two user type, which is admin for school and admin from KPM. Each school can only have one admin that manage the application in term of, teacher management and student management. School admin can add, view, update and delete the teacher data. Admin also can do that for the student data. Student may have a discipline case and exam result, so admin also can perform CRUD for both of the data. KPM admin can only view the amount of student and teacher or each school. From this, the admin can allocate the teacher based on the need of each school.

The common structure of tree widgets you used when designing and developing the application

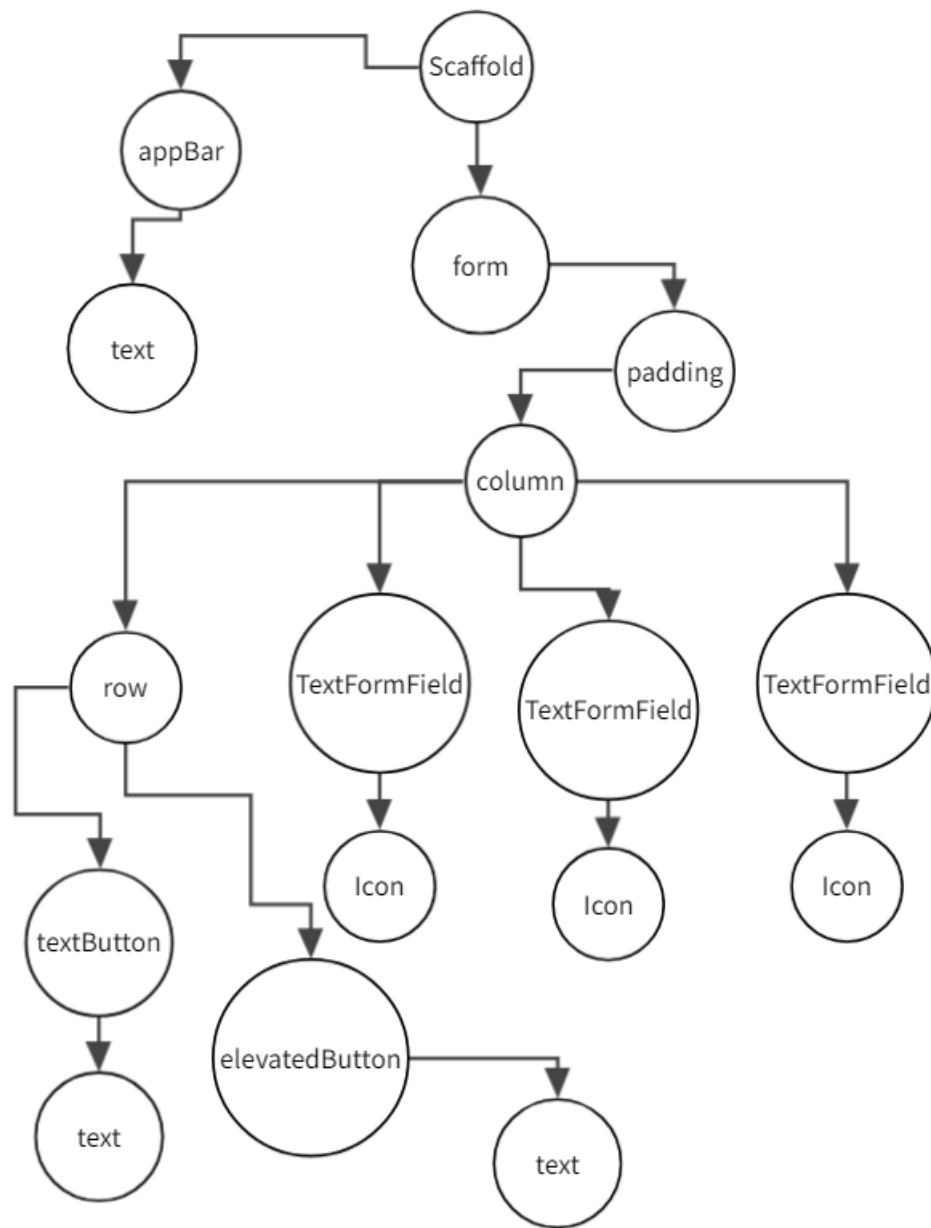
Login Widget Tree



Teacher info screen widget tree



Add student screen tree widget



Flutter widget and features adopted in the application

Flutter Widget Used in the Application

1. Inkwel
2. DataTable
3. DataCell
4. Spacer
5. ExpansionTile
6. PopupMenuButton
7. PopupMenuitem
8. ThemeConsumer
9. Column
10. Row
11. Container
12. Padding
13. Scaffold
14. Expanded
15. TextFormField
16. SizedBox
17. Drawer
18. Icon
19. IconButton
20. Form
21. AlertDialog
22. Dismissible
23. ElevatedButton
24. SnackBar
25. DropDownButtonFormField
26. ListView
27. ListTile
28. Card
29. Text
30. AppBar
31. TextButton

Features adopted in the application

1. Login

User login use firebase authentication for security measure

2. CRUD for teacher data

Can create, view, update and delete for teacher data

3. CRUD for student data

Can create, view, update and delete for student data

4. CRUD for student discipline case

Can create, view, update and delete for student discipline case data

5. CRUD for student result

Can create, view, update and delete for student result data

6. Search bar

Can search teacher based on staff id and student based on matric no

7. Theme changer

Can change theme, light or dark based on user preference

8. View amount of teacher

Can view the amount of registered teacher in the homepage screen


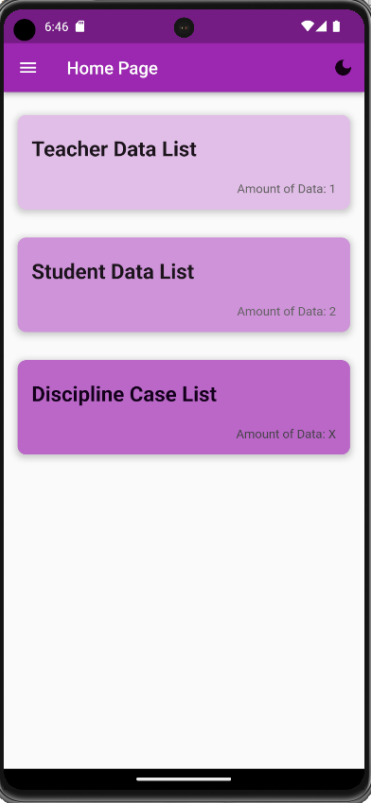
9. View amount of student

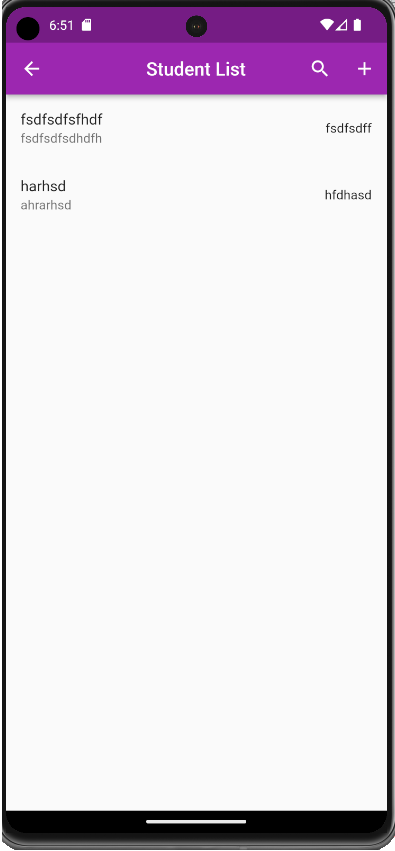
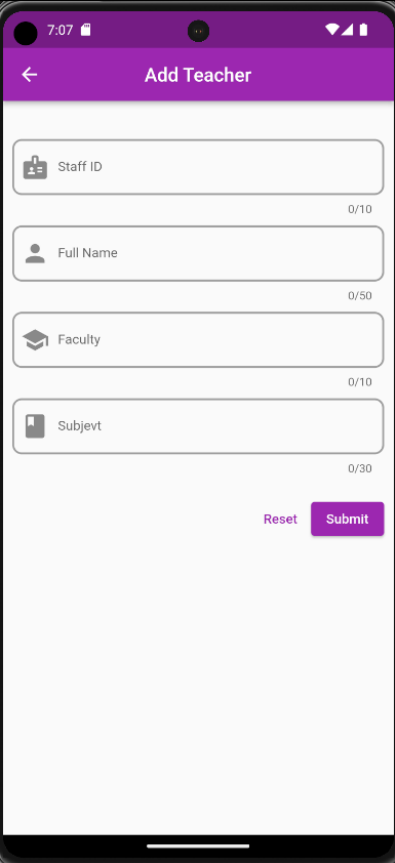
Can view the amount of registered teacher in the homepage screen

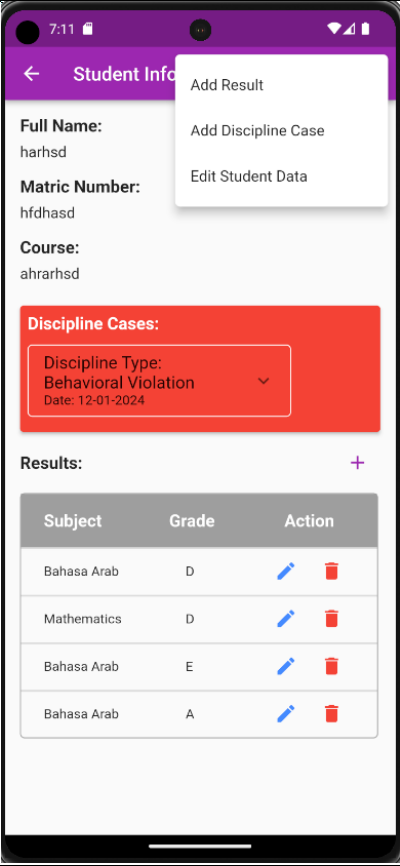
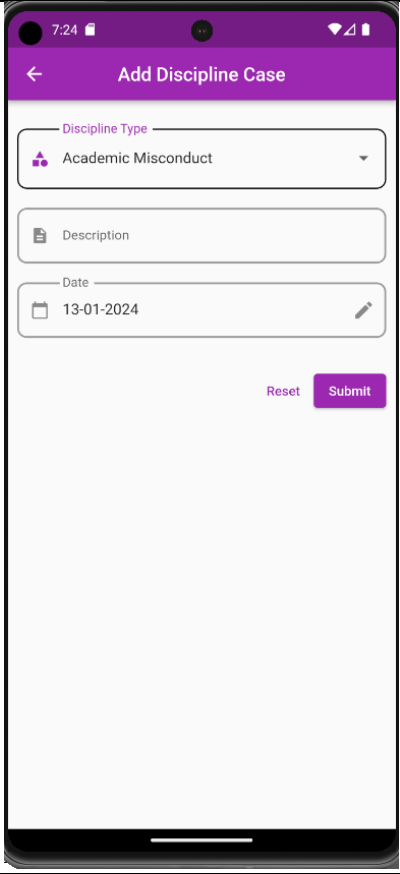
10. User authentication

Different user type will have different interface after login, and the user will have different database data based on login credential.

UI for the application with explanation

UI Interface	Explanation
 The image shows a mobile app interface for login. At the top, there's a header with the logo of 'KEMENTERIAN PENDIDIKAN MALAYSIA' (Ministry of Education Malaysia) and a background image of a school building. Below the header, there's a 'Login' section with the subtitle 'School Management App'. It contains two input fields: 'Email' and 'Password'. At the bottom of the login section is a purple button with a lock icon and the text 'Login'.	<p>This is the login page or the admin to log into the application, different admin type will direct to different page upon login, admin same admin type will direct to same page, but the data inside is based on the admin school data.</p>
 The image shows the home page of the app. At the top, there's a purple header bar with a hamburger menu icon on the left, the text 'Home Page' in the center, and a moon icon on the right. Below the header, there are three purple rectangular widgets. The first widget is titled 'Teacher Data List' and shows 'Amount of Data: 1'. The second widget is titled 'Student Data List' and shows 'Amount of Data: 2'. The third widget is titled 'Discipline Case List' and shows 'Amount of Data: X'.	<p>Upon login, the admin can see the amount of data for teacher and student., admin then can navigate to the screen that listing the student or teacher upon tapping on the respective widget corresponding to the data.</p>

UI Interface	Explanation
	<p>This is the list or the student data, there are plus sign button to add a new student and a search button to search respective student based on their matric no. The student list tile uses dismissible to delete the data of respective student. The list tile also can be tapped to further navigate the user to the student info screen. The teacher list screen also work the same way as the student list screen.</p>
	<p>This is the interface to add new teacher data, each input form have a validator to check if the input form whether is it null or not. Then, the data will be saved into the database when the submit button is tapped. When tap the reset text button, it will reset the form into blank again. The user interface for the adding student is also the same as teacher but have a different amount of parameter and value.</p>

UI Interface	Explanation
	<p>This is the student info screen. Inside here, we have student data and the added discipline case and result for the student. The popup menu button gives the user a better navigation whether to add a new discipline case or add result or edit the student data. The student result will be displayed here as a record of the student exam result. The discipline case can be edited and deleted when the user taps the down arrow button inside the discipline case card widget.</p>
	<p>This is the interface for adding the discipline case. The discipline type uses a dropdown button form field to select the discipline type, then the user/admin needs to add a bit of description about what happened and lastly specify the date of the case. This can be a good record to see the student performance in school. The data will then be kept in the database when submitted and displayed like the above user interface.</p>

Conclusion

In the course of developing the school management app, I gained valuable insights into mobile app development using Flutter. Navigating through the intricacies of theme customization, responsive UI design, and integration with Firebase for data storage and retrieval provided me with a solid foundation in front-end development. The implementation of features such as discipline cases, student results, and a dynamic theme switcher not only showcases the functionality of the app but also reflects my proficiency in handling state management and asynchronous operations. As the project evolved, I recognized the potential for it to transition from a prototype to a fully functional real-world application. Future development could involve enhancing user authentication, refining data validation, and incorporating additional features like push notifications for timely communication. The project not only reinforced my technical skills but also emphasized the importance of user experience and code maintainability. Overall, this project journey has equipped me with a robust skill set and the confidence to pursue further advancements in mobile app development.

Reference

1. Flutter - Dart API docs. (n.d.). Api.flutter.dev. <https://api.flutter.dev/>
2. Flutter - YouTube. (n.d.). Www.youtube.com. <https://www.youtube.com/@flutterdev>
3. Flutter UI | Beautiful Login Page UI Design - Day 38. (n.d.). Www.youtube.com. Retrieved January 6, 2024, from https://www.youtube.com/watch?v=4tImN_3YNTI&list=PLf-jOHs0PF3uxzMCFLfj9W3hRxLzHPGX&index=27
4. ChatGPT. (n.d.). Chat.openai.com. Retrieved January 10, 2024, from <https://chat.openai.com/share/2d99880d-1eed-4f11-85c3-95c9e4526343>
5. HeyFlutter.com. (2022, April 21). Flutter Firebase Authentication [2023] The cleanest Way [Video]. YouTube. <https://www.youtube.com/watch?v=4vKiJZNPhss>
6. ChatGPT. (n.d.). <https://chat.openai.com/share/79724bdb-c85b-4a74-81c0-156336497e39>