UNIVERSITY MALAYSIA TERENGGANU

FACULTY OF COMPUTER SCIENCE AND MATHEMATICS (FSKM)

CSM3114
FRAMEWORK-BASED MOBILE APPLICATION DEVELOPMENT

Project 2 Final Report

School Management App

Prepared by:
Nisha Farhani Binti Khirul Fozi

Prepared for:
Dr. Mohamad Nor Bin Hassan

Bachelor of Computer Science (Mobile Computing) with Honors
SEMESTER I 2023/2024

Contents

1. Executive Summary

The proposed school management mobile application aims to digitize and automate core administrative processes in schools. Built using Flutter, Google's robust cross-platform development framework (Google LLC, n.d.), the app will be deployable on both Android and iOS devices. This allows for widespread accessibility and usage by school staff and parents.

The app design centers around key workflows like student registration, teacher management, uploading marks, and tracking discipline records. Using a combination of forms, tables and analytics, the app will capture and display all critical information. The data architecture will be powered by Google Firebase, a flexible cloud-based NoSQL database (Google LLC, n.d.). Its real-time database and built-in authentication will enable secure access to the data.

Key benefits of the proposed app include anytime, anywhere access to records, reduction in paperwork, structured data capture, and automation of mundane tasks. It will also facilitate transparency between the school, teachers and parents through instant updates. From an administrative perspective, the app will lead to time and resource savings through streamlined processes. Overall, the app aims to boost productivity, accountability, and stakeholder engagement.

In the future, features like attendance tracking, fees management, notifications and statistics can be incorporated to make the app more robust. With rigorous testing and focus on user experience, the school management app can become an essential tool for 21st century education management. It has the potential to transform antiquated approaches by leveraging the versatility of mobile platforms.

2. Use Case Diagram

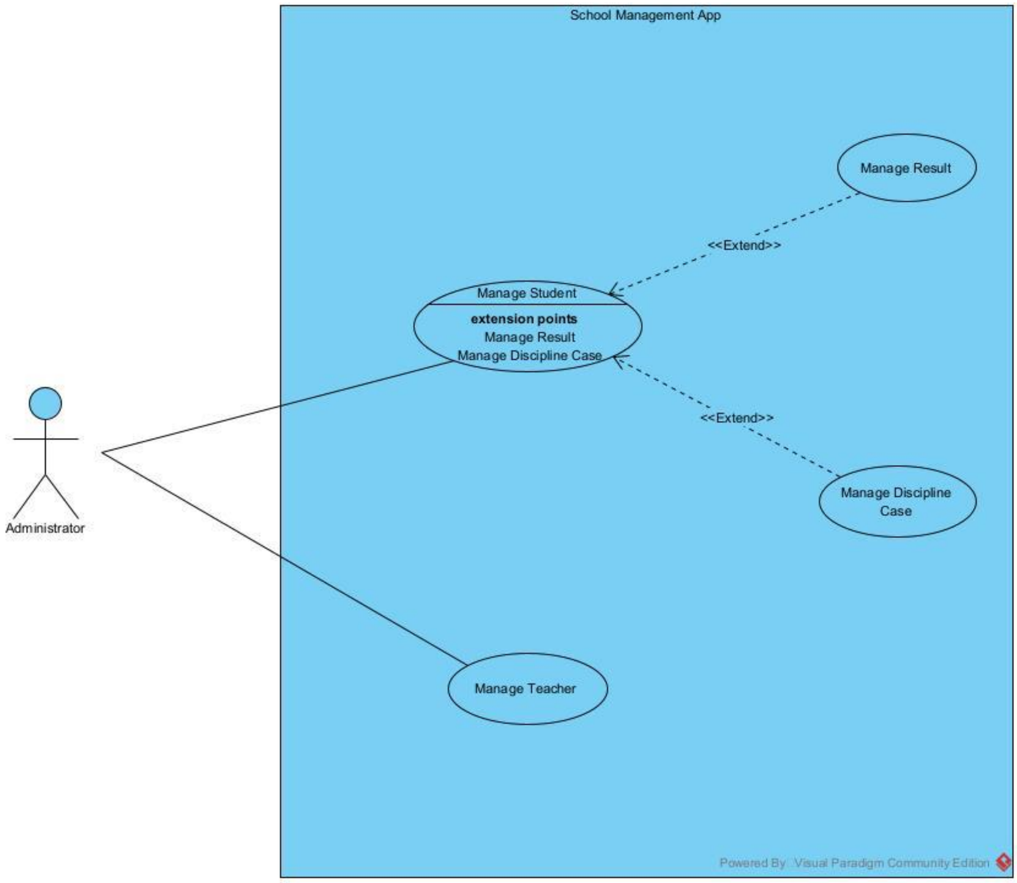


Figure 1 *Use Case Diagram*

Figure 1 shows the use case diagram of the app which illustrates on how the app will function. This app will be fully handled and used by the school administrator. He or she will log in with the correct credentials into the app. In the app, the admin can manage two things which is the student and teacher. The student functionalities can be extended to where admin can manage student result as well as manage student's discipline case.

3. Widgets Tree structure

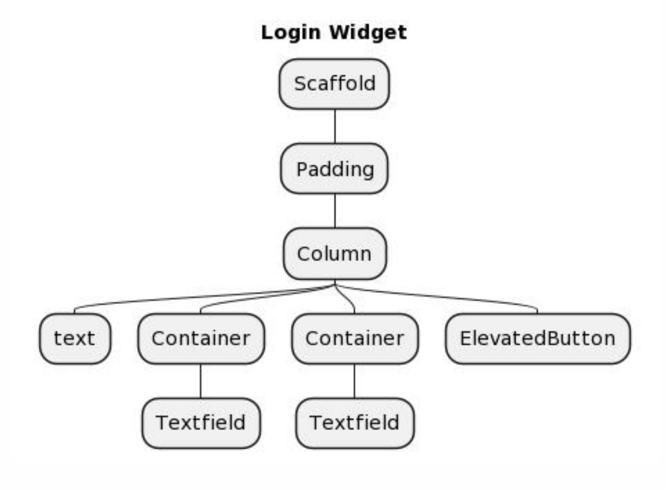


Figure 2 *Login Page Widget Tree*

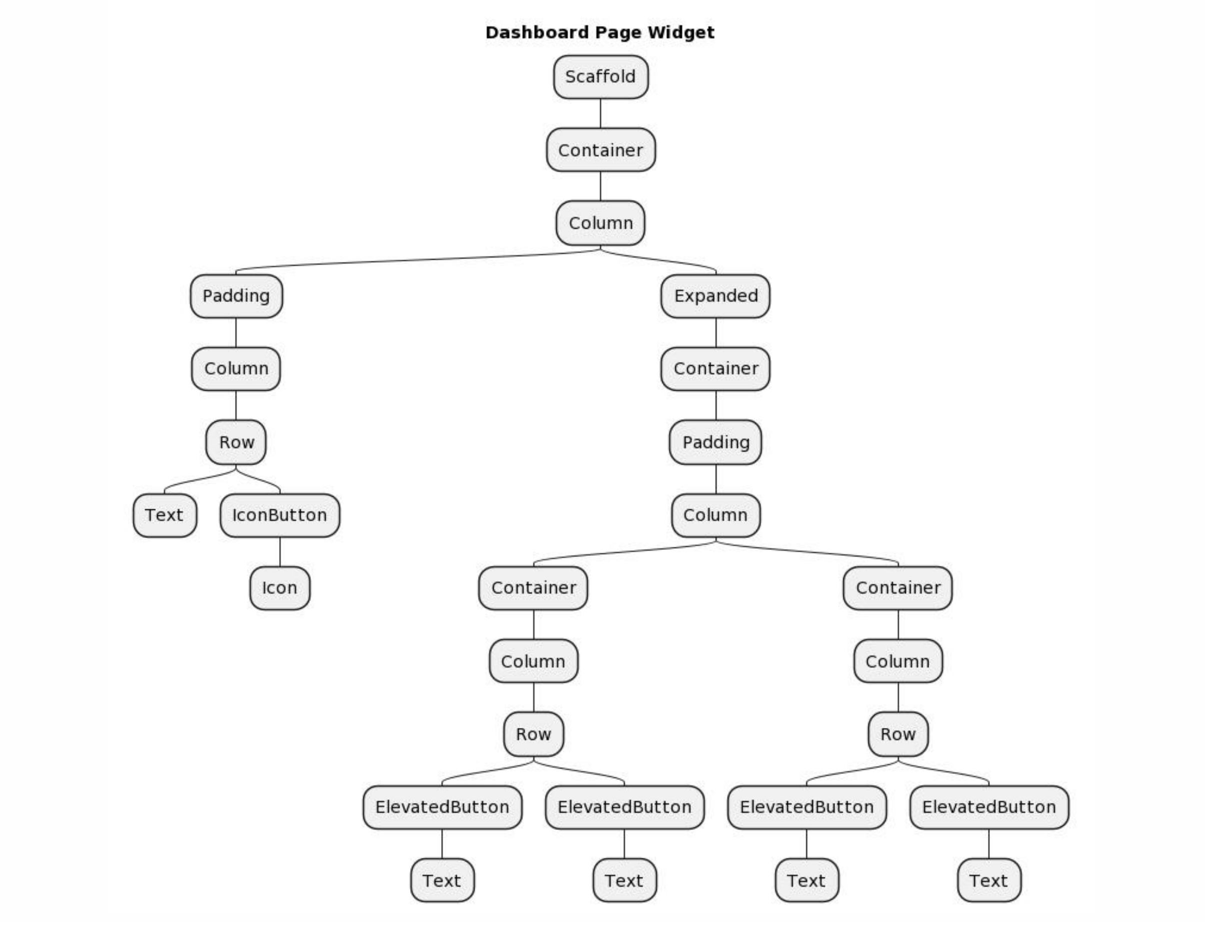


Figure 3 *Dashboard Page Widget Tree*

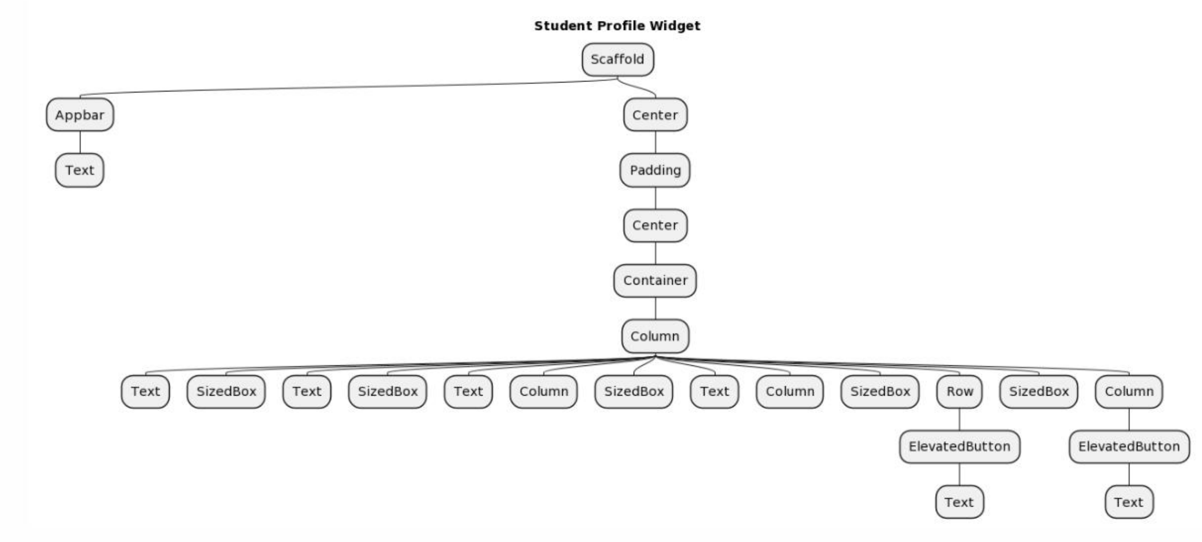


Figure 4 *Student Profile Widget Tree*

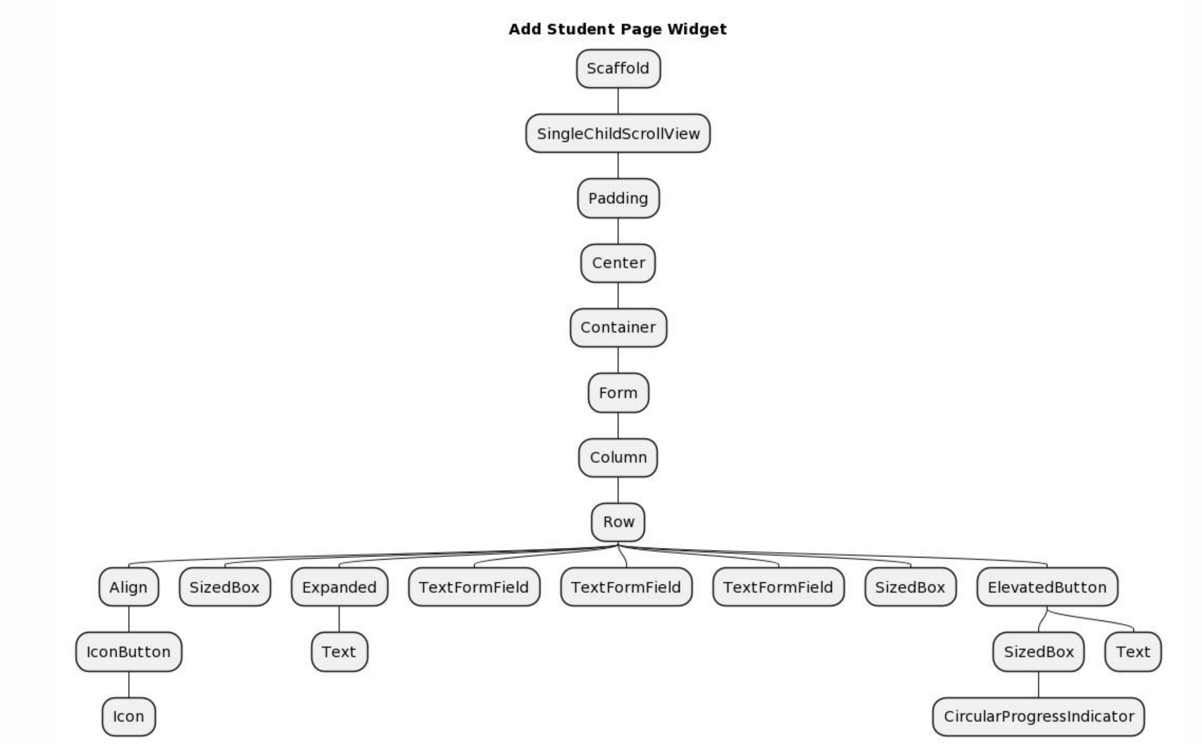


Figure 5 *Add Student Page Widget Tree*

4. Flutter Widgets & Features

4.1 Widgets

- **Scaffold**: Provides a basic structure for the visual interface of the app, including the app bar, body, and bottom navigation.
- **Container**: A box model that can contain other widgets and is used for styling and layout.
- **Column**: Arranges its children in a vertical column.
- **Row**: Arranges its children in a horizontal row.
- **Text**: Displays text with specified styles.
- **Padding**: Adds padding around its child widget.
- **ElevatedButton**: A material design raised button.
- **IconButton**: A button that contains an icon.
- **TextField**: Allows users to input text.
- **Stack**: Overlaps several children widgets.
- **Positioned**: Positions a child within a **Stack**.
- **Image**: Displays images.
- **ClipOval**: Clips its child in an oval shape

4.2 Features

- **Firebase Integration:**

  Use of Firebase for authentication (**FirebaseAuth**).

  Utilization of Firebase Realtime Database (**FirebaseDatabase**).
- **Navigation:**

  Navigation between different screens using **Navigator** and named routes.
- Use of **pushReplacementNamed** for navigating to a new screen and replacing the current one.
- **Form Validation:**

  Utilization of **Form** and **GlobalKey<FormState>** for form validation.

  Various **TextFormField** widgets for user input with different validation rules.
- **Asynchronous Operations:**

  Use of **async** and **await** for handling asynchronous operations (e.g., Firebase authentication, HTTP requests).

- **State Management:**

  State management using **StatefulWidget** and **setState** for updating the UI based on changes in the application state.

- **Loading Indicators:**

  Display of loading indicators (e.g., **CircularProgressIndicator**) during asynchronous operations.

- **Styling:**

  Styling of widgets using properties like **backgroundColor**, **color**, and **fontSize**.

  Customization of button appearance using **ElevatedButton.styleFrom**.

- **Dialogs:**

  Displaying dialogs using **showDialog** for user interaction during certain operations.

5. Sample Interface

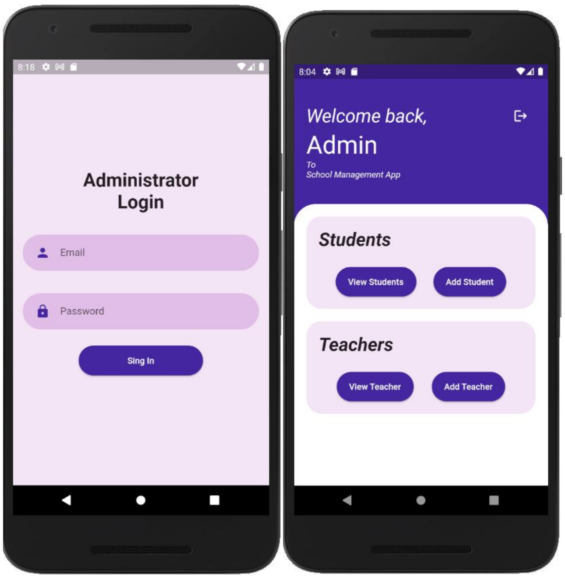


Figure 6 *Login & Dashboard Interfaces*

Figure 6 shows the login and dashboard interfaces where the school admin will first go through when opening this app. The login process used the Google Firebase Authentication to make it easy for the admin to enter the app with his credentials. The dashboard page will have a few buttons that can navigate to the main functions of the school management app.

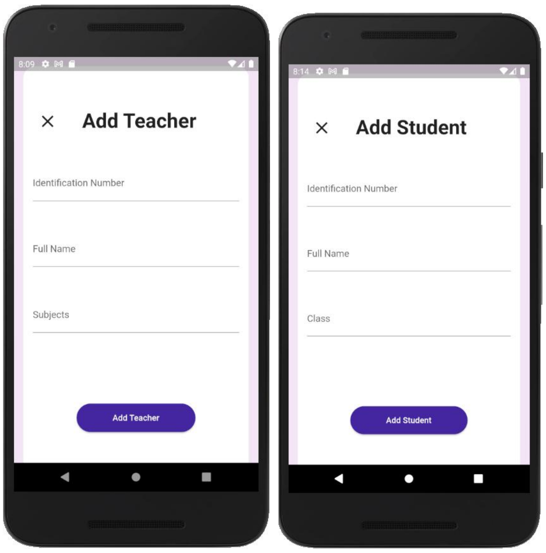
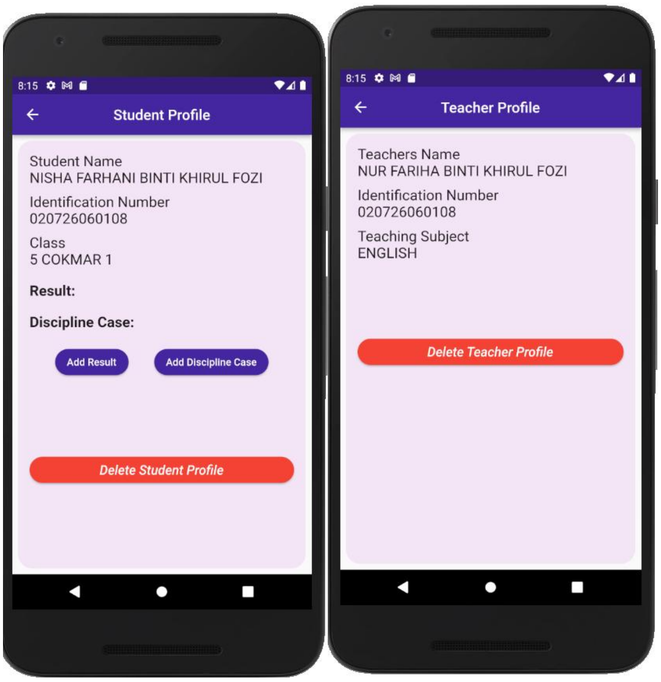


Figure 7 *Add Student & Teacher Interfaces*

Figure 7 shows the add teacher and add student forms where admin can enter the correct credentials of both individuals. The data entered will be stored in the Google Firebase real time database.



Student Name
NISHA FARHANI BINTI KHIRUL FOZI
Identification Number
020726060108
Class
5 COKMAR 1
Result:
Discipline Case:

Teachers Name
NUR FARIHA BINTI KHIRUL FOZI
Identification Number
020726060108
Teaching Subject
ENGLISH

Figure 8 *Student & Teacher Profiles Interfaces*

Figure 8 shows the student and teacher profiles interfaces where all the added details will be displayed here. For student, it has additional functionalities to add the student's result and add discipline case. Both of student and teacher details can be deleted from the system when the delete button is clicked.

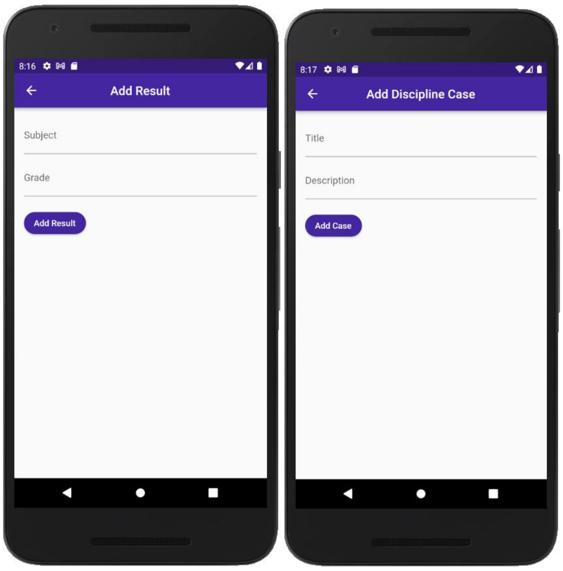
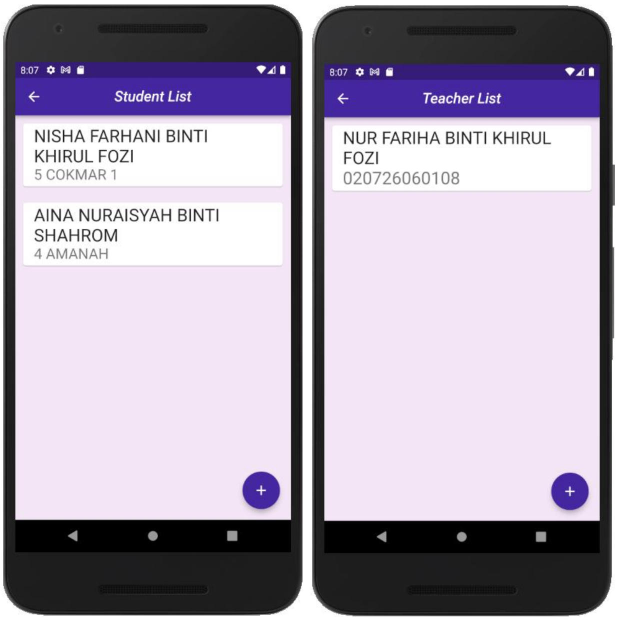


Figure 9 *Add Student Result & Discipline Cases Interfaces*

Figure 9 shows the simple forms that admin can used to add students' result and discipline cases if needed. When the details are added, they will appear add the student profile page.



Student List

NISHA FARHANI BINTI KHIRUL FOZI
5 COKMAR 1

AINA NURAISYAH BINTI SHAHROM
4 AMANAH

Teacher List

NUR FARIHA BINTI KHIRUL FOZI
020726060108

Figure 10 *Student & Teacher List Interfaces*

Figure 10 shows the list of students and teachers after admin has entered their details. The details will be shown in a card form. When the card form is clicked, the screen will navigate to the student or teacher profile screen.

6. Conclusion

In summary, the proposed school management app offers a modern solution to digitize and upgrade administrative processes in academic institutions. Built on Flutter and Firebase, it provides strong technical foundations for a scalable cross-platform app. Automating mundane workflows will lead to improved productivity for staff and faculty. It will also provide transparency to parents by giving instant access to student records. With a thoughtful design focused on core functions, the app can benefit multiple stakeholders.

Incremental enhancements through versions will further augment the app's capabilities. Extensive testing will be crucial to ensure full-proof functioning prior to rollout. Close engagement with users during development will allow for incorporating relevant feedback. Overall, by combining the ubiquity of mobile devices with the power of cloud-based tools, the school management app is poised to drive positive change in school administration. It is an idea fit for the digital age with immense potential for transforming antiquated systems.

Project GitHub Repository Link:

https://github.com/dothenish/S63652-project-2

References

Google LLC. (n.d.). Flutter. https://flutter.dev/

Google LLC. (n.d.). Firebase. https://firebase.google.com/

Biessek, A. (2019). *Flutter for beginners: An Introductory Guide to Building Cross-Platform Mobile Applications with Flutter and Dart 2*.

Flutter - Dart API docs. (n.d.). https://api.flutter.dev/

GeeksforGeeks. (2022a, April 26). *Flutter Smooth page indicator*. https://www.geeksforgeeks.org/flutter-smooth-page-indicator/

Taimoor. (n.d.). *Flutter Onboarding Screen – Flutter Liquid Swipe | Coding with T*. https://codingwitht.com/flutter-onboarding-screen-flutter-liquid-swipe/

Flutter : Unhandled Exception: Null check operator used on a null value. (n.d.). Stack Overflow. https://stackoverflow.com/questions/70748009/flutter-unhandled-exception-null-check-operator-used-on-a-null-value

No Firebase App "[DEFAULT]" has been created - call Firebase.initializeApp() in Flutter and Firebase. (n.d.). Stack Overflow. https://stackoverflow.com/questions/63492211/no-firebase-app-default-has-been-created-call-firebase-initializeapp-in

ChatGPT. (n.d.). https://chat.openai.com/share/29e6e640-97bb-4854-a786-0535ca9d96c8