UNIVERSITY MALAYSIA TERENGGANU
FACULTY OF OCEAN ENGINEERING TECHNOLOGY & INFORMATICS

[CSM3114]
FRAMEWORK-BASED MOBILE APPLICATION DEVELOPMENT
(GROUP 1)

ASSIGNMENT 2
"SMART E- COURSE APPLICATION"

PREPARED BY:
NOORNAJWA AMIRAH BINTI ZAKARIA
(S61410)

PREPARED FOR:
DR. MOHAMAD NOR BIN HASSAN

[*BACHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING) WITH HONOURS*] SEMESTER III 2022/2023

Table Of Content

1.0 EXECUTIVE SUMMARY

The Smart E-course App is a dynamic and interactive ecosystem created to meet the changing needs of contemporary learners, not merely a platform for accessing educational content. A personalized learning journey is established by an easy-to-use login process that allows users to customize their experience based on their academic preferences and objectives.

The app's main feature, the "Course" screen, offers users a carefully chosen list of all the courses that are available. The app gives users the ability to make well-informed decisions about their academic pursuits by providing important details like lecturer names and credit hours. Users can enroll in the courses of their choice quickly and effectively thanks to the smooth enrollment process. Its dedication to exclusivity and user engagement is what makes this app unique. Users are encouraged to add new courses on the "Add Course" screen, which not only lets them sign up for already-existing courses but also helps the app grow.

When it comes to time management, the "Scheduled Course" feature is revolutionary. With access to comprehensive details regarding the day, time, and venue of their enrolled courses, users can plan and organize their academic commitments with ease. This leads to a more organized and stress-free learning environment in addition to increasing productivity.The Smart E-course App goes one step further in personalization by letting users enter and monitor their exam results. This feature encourages users to take responsibility for their academic performance in addition to acting as a digital grade book.

In summary, surpasses conventional e-course platforms. With a focus on community involvement, user empowerment, and customized learning experiences, it is a comprehensive educational companion. The app is a shining example of innovation in the field of digital education because it integrates course management, scheduling, and performance tracking with ease.

2.0 USE CASE DIAGRAM

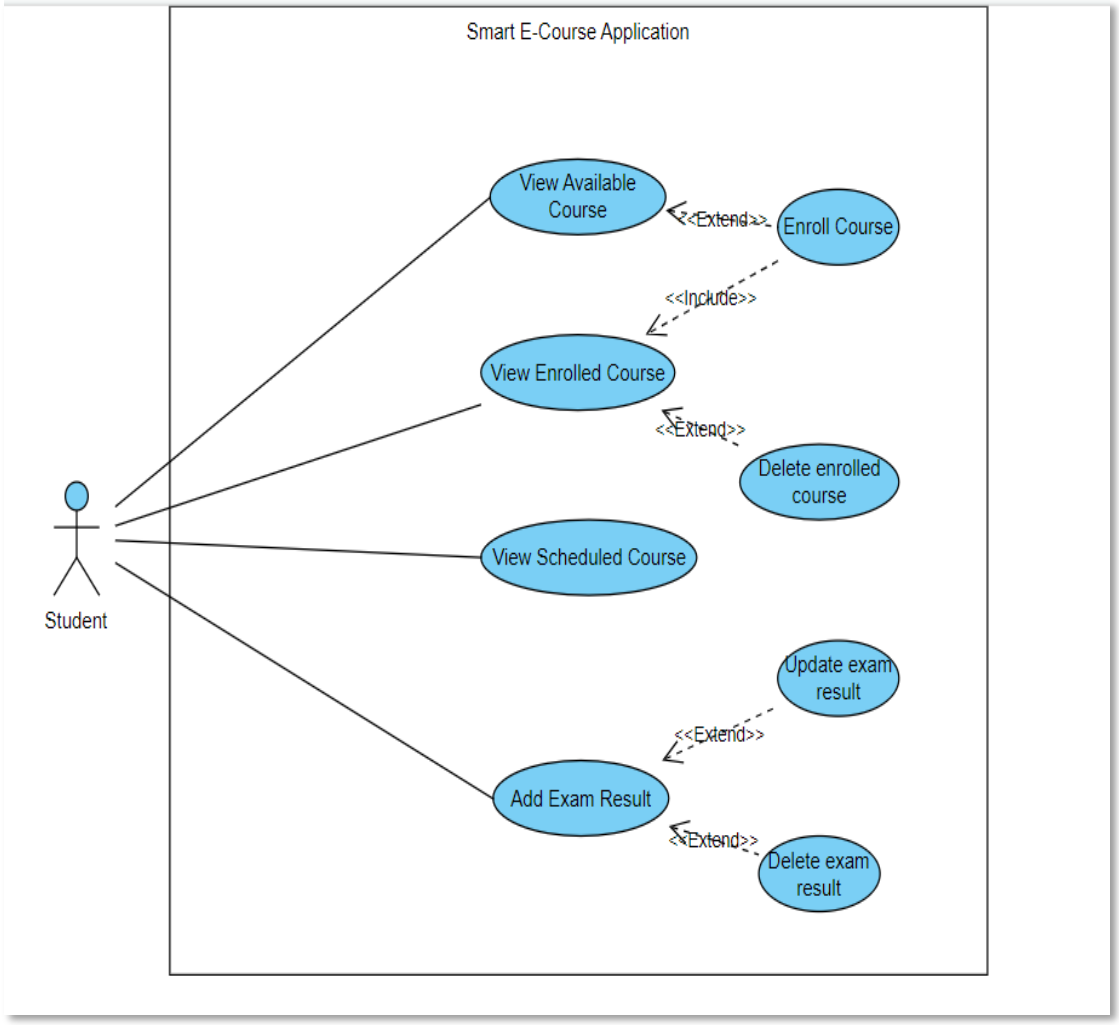
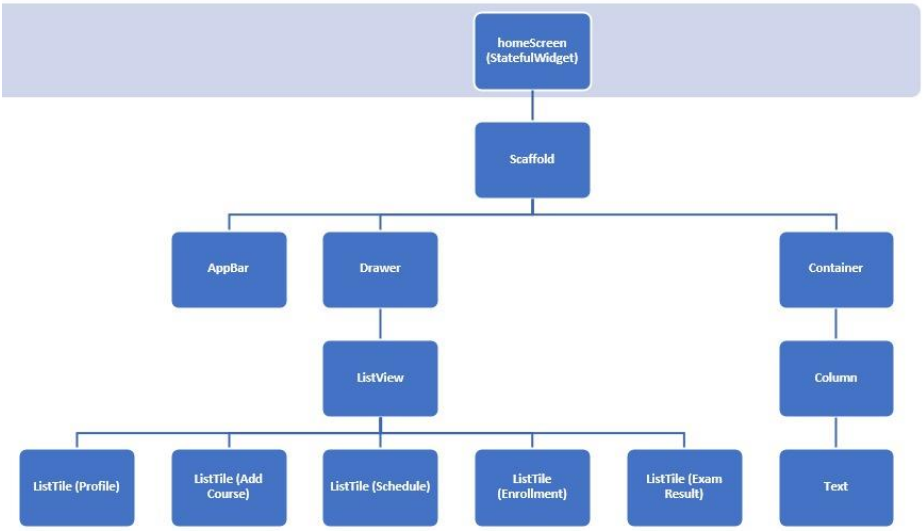


Figure 1. Use Case Diagram Smart E-Course Application

Firstly, Student logs into the Smart E-Course App. Student navigates to the "View Available Course" section and the system displays a list of available courses. At this section, If the student is interested in a course, they have the option to extend this use case by enrolling in that course. This is an optional extension as not all viewed courses will be enrolled.For View Enrolled Course, this use case involves a student viewing the list of courses they are currently enrolled. It also includes the functionality for viewing enrolled courses and user also can delete enrolled course. This is an optional extension as not all course will delete. Next, user can view Schedule Course which is allows a student to view the schedule for a specific information for course they are enrolled in.Lastly, at update exam result section, it allows a student to update and delete an existing exam result.

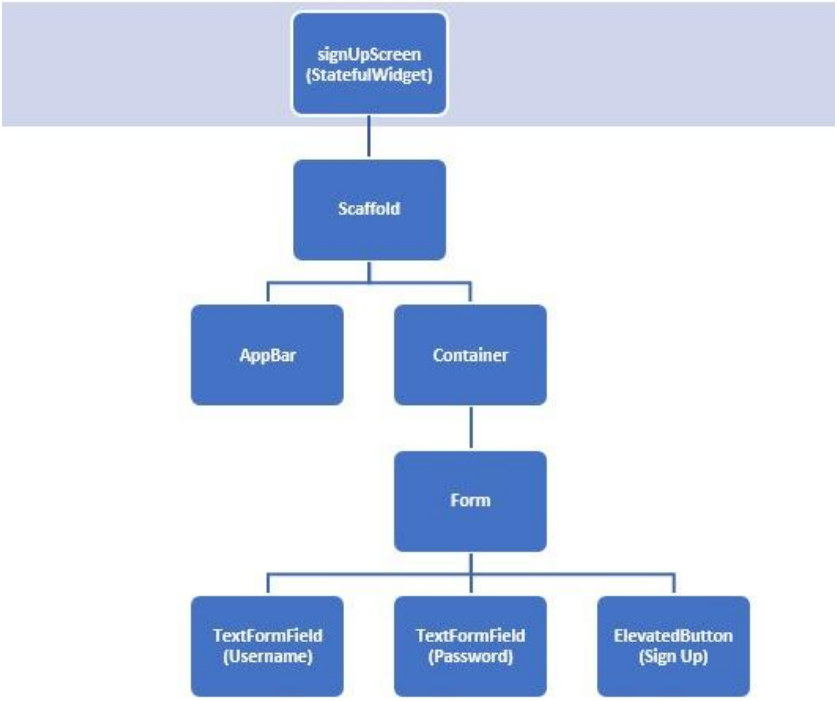
3.0 STRUCTURE OF TREE WIDGET

1. Home Screen



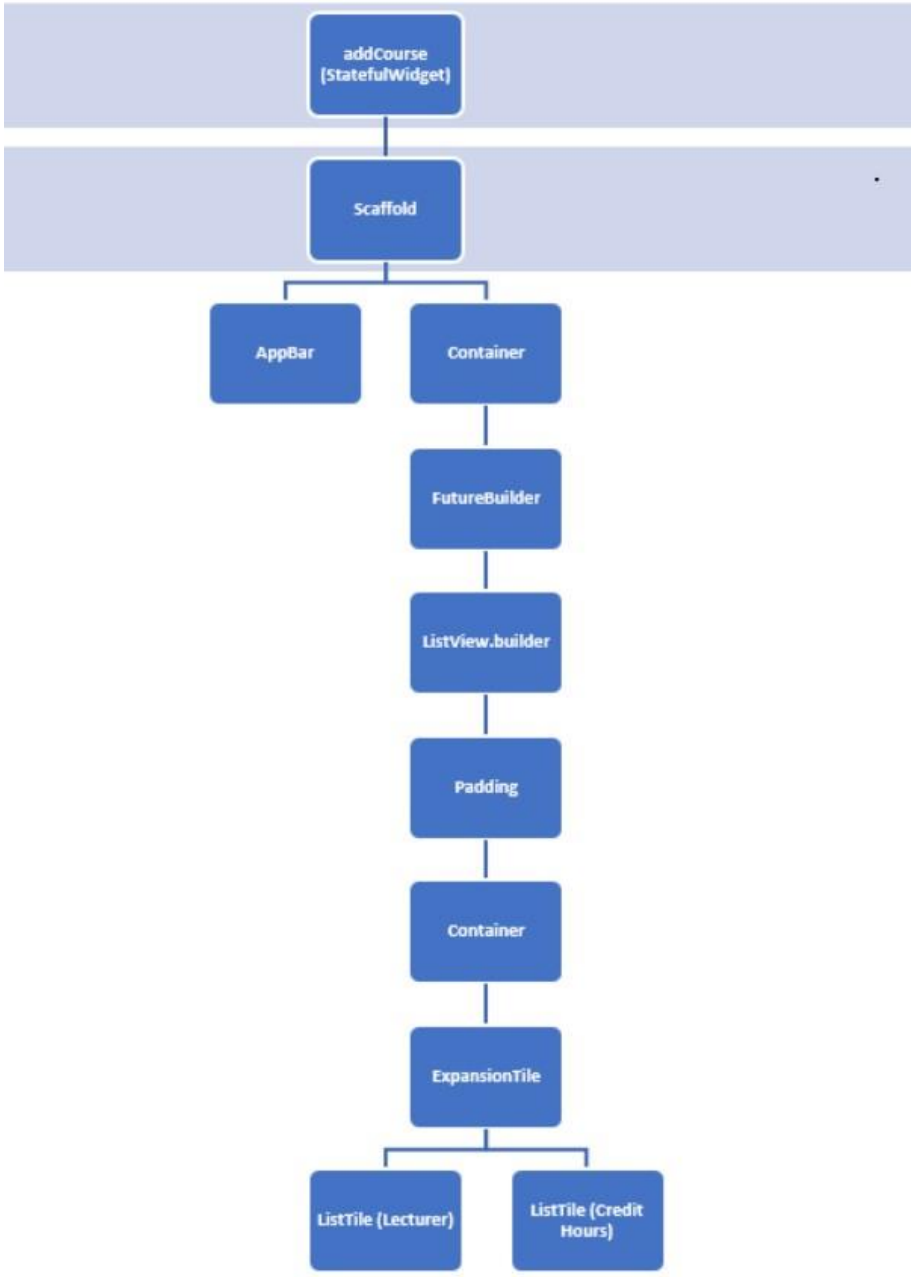
---

2. signUp_screen.dart



---

3. addCourse_screen.dart



4.0 FLUTTER WIDGET AND FEATURES

1. MaterialApp:

- The MaterialApp widget is the root of your Flutter application. It provides access to various Material Design widgets and features.

2. StatefulWidget:

The ExamResult class is a StatefulWidget. This allows for dynamic behavior and state changes within the widget.

3. **TextEditingController:**

TextEditingController for TextFormFields. These controllers are used to control the text input and retrieve the entered values.

4. Scaffold:

- The main structure of your screen is defined using the Scaffold widget. It provides a basic structure for the visual elements of the screen, such as the app bar, body, and floating action button.

5. AppBar:

- An AppBar is used to create a top app bar with a title. that customized it by setting a background color and a bold title.

6. Container:

- The Container widget is used as the main container for the entire body. It has a background image set using DecorationImage.

7. Card:

- A Card widget is used to contain the form elements. It provides elevation and a visually distinct card-like structure.

8. Column:

- A Column widget is used to arrange children in a vertical column. This is used to structure the form elements vertically.

9. ElevatedButton:

- An ElevatedButton is used to create a button for saving the entered details. It has an onPressed callback to handle the save action.

10. FloatingActionButton:

- A FloatingActionButton is used to create an "Edit" button. It triggers an action that sets previous data into the text controllers and updates the UI.

11. Image:

- An image is set as the background of the Container using DecorationImage.

12. Printing Data:

- The entered data is printed to the console when the "Save" button is pressed. This is a placeholder for any logic you want to implement when saving data.

13. Customization:

- The EmoticonFace widget is designed to be easily customizable by providing the emoticon face string through its constructor. This allows you to reuse the widget for different emoticons.

14. FutureBuilder:

- The FutureBuilder widget is used to asynchronously load data from the network using the fetchCourses function. It helps in managing the UI based on the state of the future.

15. http:

The http package is used to make HTTP requests. In this case fetching a list of courses from a remote database.

16. **ExpansionTile** :

- The ExpansionTile widget is used to create an expandable tile for each course, allowing users to see additional details like the lecturer's name and credit hours.

5.0 SAMPLE OF INTERFACE

1. Welcome Screen

Figure 2 Welcome Screen

User must log in at this screen in order to use the application. However, if a user is using it for the first time, they must first register by signing up before they can log in.

2. Sign up Screen

Figure 3 Sign Up Screen

The user can create an account by providing their email address, full name, username, and secure password. Click the back button and then the sign up button to proceed if the user wants to cancel.

3. Login Screen

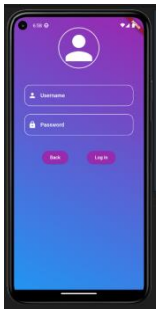


Figure 4 Login Screen

The user must enter their username and password on the login screen, exactly as they did when they registered. Clicking the back button will take the user back to the sign-up page; clicking the sign-up button will bring them to the home screen.

4. **Home Screen**

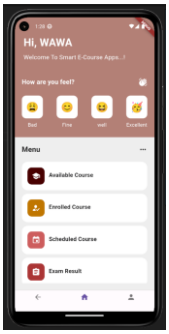


Figure 5 Home Screen

This screen shows a straightforward message to the user along with menus for scheduled courses, enrolled courses, available courses, and exam results. Additionally, the navigation bar icon is present. The icon home for the home page, the arrow icon to return to the previous screen, and the person icon to log out

5. **Course List Screen**

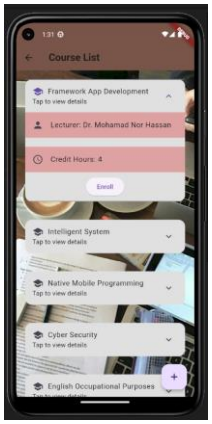


Figure 6 Course List Screen

The system presents a list of available courses, and the user can view it at this screen. Additionally, the user can tap on the course to view details such as the name of the lecturer and the course's credit hours. By clicking the add button, users can also add new courses and enroll in them. If a student is interested in a course at this point, they can enroll in it. The enrolled screen will appear after you click the enroll button.

6. **Enrolled Screen**

Figure 7 Enrolled Screen

The user can view the enrolled course from the course list screen, the previous screen, at the enrolled screen. To view the scheduled course, the user can click the "view schedule" button.

7. **Scheduled Screen**

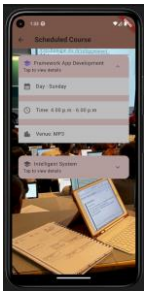


Figure 8 Scheduled Screen

The user can view the course details, including the day, time, and venue, by selecting the "view schedule" button on the "enrolled" screen.

8. **Exam Result Screen**

Figure 9 Exam Result Screen

Finally, the exam result screen gives the user the ability to add, edit, and remove previous exam results.

6.0 CONCLUSION

In summary, the Smart E-course App is a shining example of innovation and efficiency in the field of digital education. Unlike traditional e-course platforms, it recognizes the changing needs of modern learners and provides them with a personalized and engaging learning experience in addition to educational content. Users can customize their experience according to their academic goals and preferences thanks to the easy login process. Users can choose from a carefully selected list of available courses via the central "Course" screen, which provides information on lecturers and credit hours to help users make educated decisions. In addition to being quick, the enrollment process allows users to contribute to the platform by allowing them to add new courses. Time management is revolutionized by the "Scheduled Course" feature, which gives users comprehensive details about the day, time, and location of their enrolled courses. This degree of organization reduces stress in the classroom and increases productivity. The "Exam Result" screen, which functions as a digital grade book and encourages user responsibility and engagement in their learning process, is another example of the app's dedication to personalization.

7.0 REFERENCE

1. *Let's clone a dribbble design using FLUTTER*. (n.d.). Www.youtube.com. Retrieved January 6, 2024, from https://www.youtube.com/watch?v=eegl7of4g-o

2. *Online Course App - Flutter UI - Speed Code*. (n.d.). Www.youtube.com. Retrieved January 13, 2024, from https://www.youtube.com/watch?v=8abMF1Y2Xnk&t=12s

3. *Error fetching JSON Data from Firebase Real Time Database in Flutter*. (n.d.). Stack Overflow. Retrieved January 13, 2024, from https://stackoverflow.com/questions/68823132/error-fetching-json-data-from-firebase-real-time-database-in-flutter

4. *How to delete data using http.delete Flutter*. (n.d.). Stack Overflow. Retrieved January 13, 2024, from https://stackoverflow.com/questions/57394495/how-to-delete-data-using-http-delete-flutter

5. *How To Use Gradients in Flutter with BoxDecoration and GradientAppBar | DigitalOcean*. (n.d.). Www.digitalocean.com. Retrieved January 13, 2024, from https://www.digitalocean.com/community/tutorials/flutter-flutter-gradient

6. *json.decode to List> issue in flutter*. (n.d.). Stack Overflow. Retrieved January 13, 2024, from https://stackoverflow.com/questions/70489300/json-decode-to-listmapstring-dynamic-issue-in-flutter

7. *Future Builder stuck at ConnectionState.waiting*. (n.d.). Stack Overflow. Retrieved January 13, 2024, from https://stackoverflow.com/questions/68033632/future-builder-stuck-at-connectionstate-waiting

8. *How to use Expanded in SingleChildScrollView?* (n.d.). Stack Overflow. Retrieved January 13, 2024, from https://stackoverflow.com/questions/56326005/how-to-use-expanded-in-singlechildscrollview

9. *Travel Mobile App UI Design In Flutter - Flutter UI Design Tutorial*. (n.d.). Www.youtube.com. Retrieved January 13, 2024, from https://www.youtube.com/watch?v=3ipHn2tcbqE&t=217s

10. *ChatGPT*. (n.d.). https://chat.openai.com/c/27e9273c-d929-44a1-9e44-c68558e180a2

Link canva:
https://www.canva.com/design/DAF5weAPA4c/elUIFOmWnmAoljOP_LnuEQ/edit?utm_content=DAF5weAPA4c&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton