UNIVERSITI MALAYSIA TERENGGANU (CENTRE)

FAKULTI SAINS KOMPUTER DAN MATEMATIK

FRAMEWORK – BASED MOBILE APPLICATION DEVELOPMENT

CSM3114

PROJECT 2

Haulier Tracking App

Project Report

Prepared by:

Ilham Hanina Madiha binti Othman (S63762)

Prepared for:

Dr. Mohamad Nor Hassan

[Bachelor of Computer Science (Mobile Computing) with Hons.]

SEMESTER I 2023/2024

Table of Contents

1.0 Executive Summary

Haulier Tracking App becomes a comprehensive answer for companies in need of delivery and truck management as the requirements of the logistics industry expand. This ground-breaking application is a businesses that require efficient cargo tracking and a great tool for truck managers seeking simpler truck operations management.

In the transportation and logistics sector, inefficiencies, imprecise communication, and delivery delays are frequent issues. Our Haulier Tracking App directly solves these problems by providing a single platform that improves real-time tracking, optimizes inbound and outbound, and facilitates seamless alignment between companies, truck management, and drivers.

Truck managers and organizations may access this mobile application, which was designed with user experience in mind, and has an intuitive interface. By delivering sophisticated capabilities like adding trucks and updating the schedule for truck, which goes beyond simple tracking, this platform gives users insights into managing logistics operations.

Our business model combines subscription plans intended for logistics companies with a per-usage fee for fleet managers to provide a scalable and sustainable revenue stream. This application stands out in a market where improvements in technology are constantly occurring because it establishes new standards for efficiency and innovation.

Haulier Tracking App has the ability to completely change how truck managers and businesses approach the challenges of modern transportation. It is a brilliant example of innovation in a world where efficient logistics are crucial to business success.

2.0 Use Case

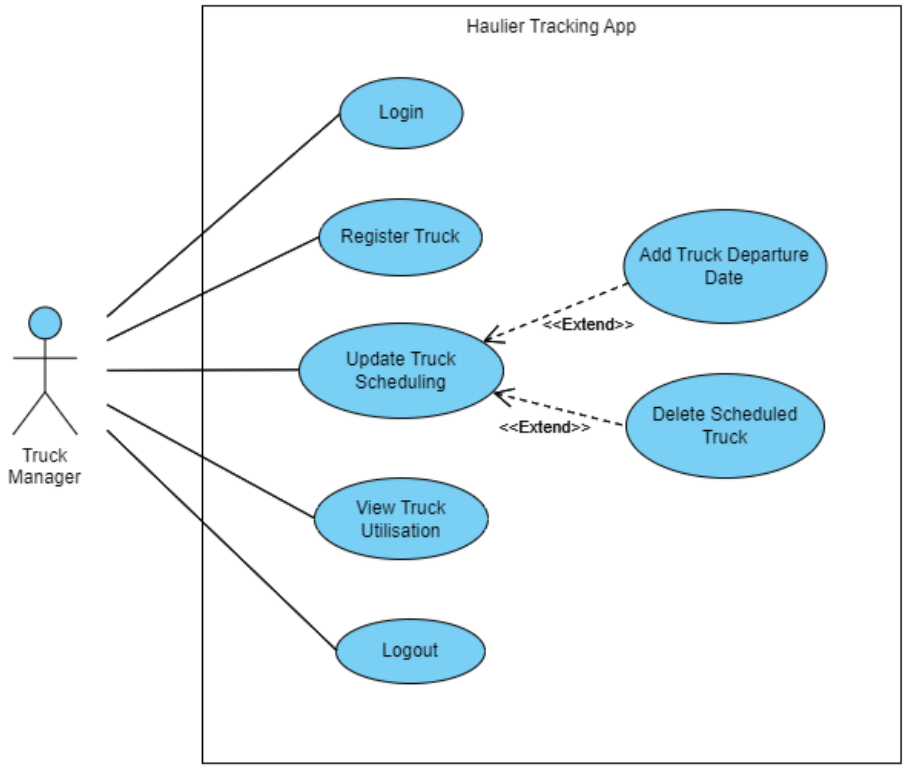


Figure 2.1: Use Case Diagram for Haulier Tracking App

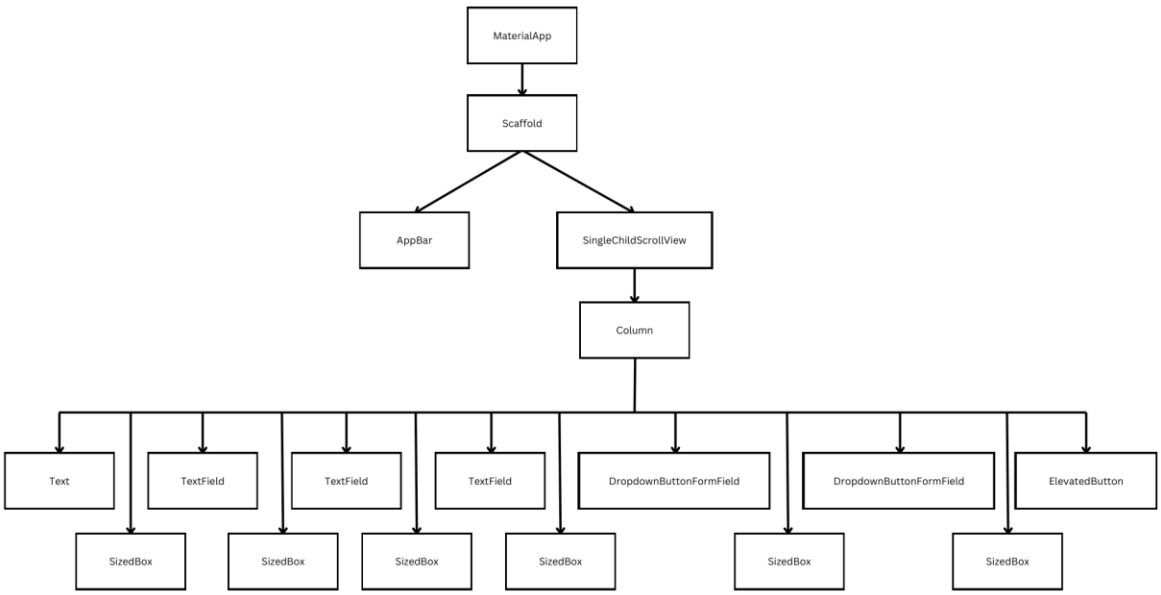
Use Case Diagram for Haulier Tracking Application according to Figure 2.1 clearly shows the system boundary specified for the Truck Manager exclusively. The statement above explains that the Haulier Tracking Application is only intended for use by Truck Managers.

There are five different use cases in this system, and each one corresponds to a specific task that might be performed by the truck manager. Logging in, registering a new truck, updating the truck schedule, viewing the truck utilisation and logging out are some of these actions.

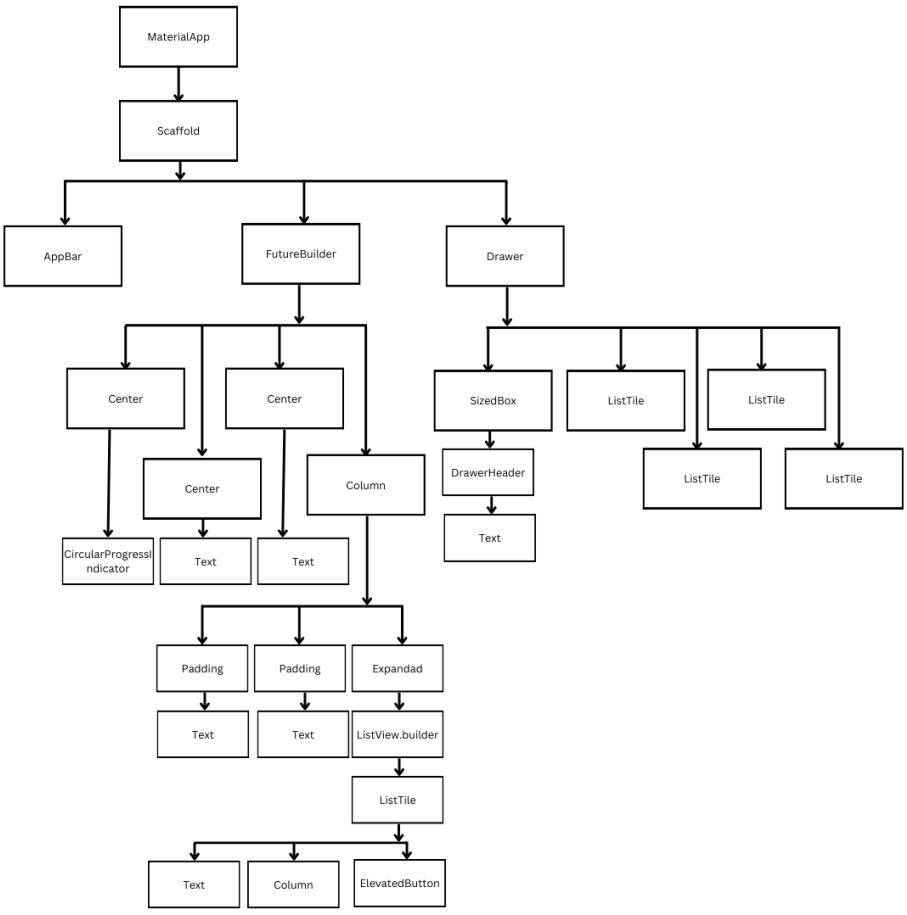
There are two extensions included in the use case update truck scheduling. In the first extension, adding the truck departure date and in the second extension, delete the scheduled truck. These connections enhance the functionality of the app and give the truck manager more options to manage the truck scheduling accurately.

3.0 Structure of Tree Widgets

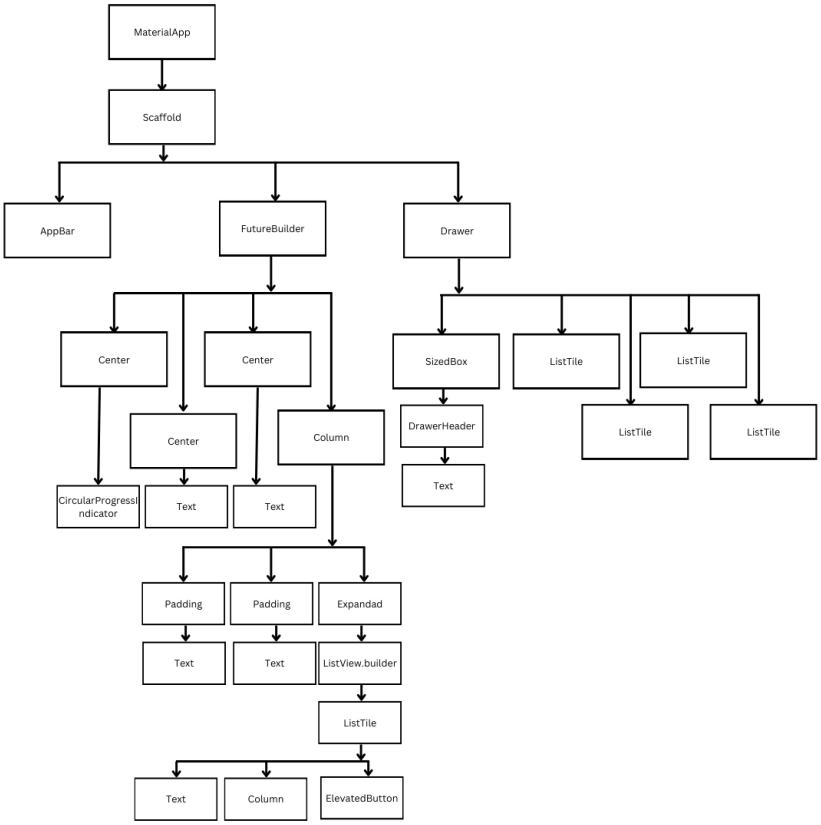
Add truck screen:



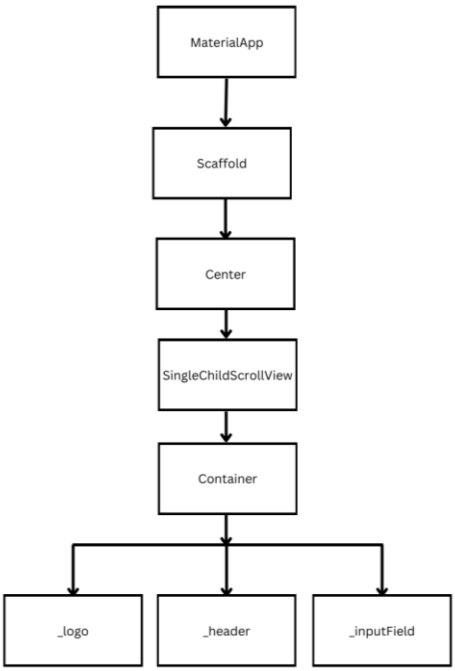
Home screen:



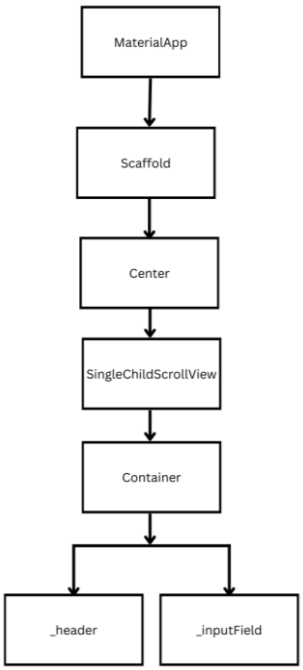
Schedule Truck Screen:



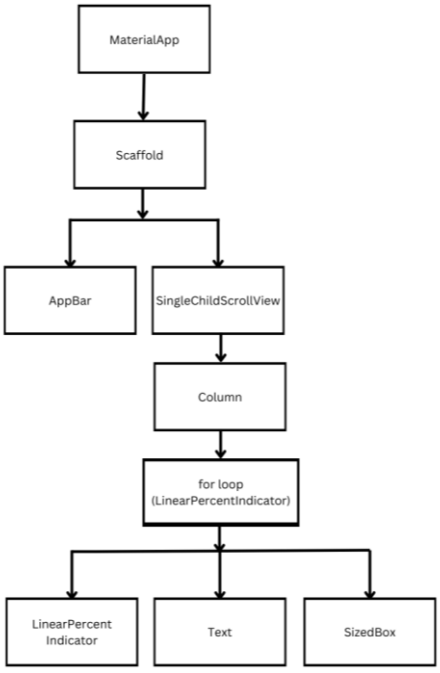
SignIn Screen:



SignUp Screen:



Truck Utilisation Screen:



4.0 Flutter Widgets and Features Adopted

Generally, Dart language and Flutter framework is used to make this application. There are some additional dependencies added in the pubspec.yaml which are intl package,http package and percent_indicator package.

SignUp Screen:

1. Scaffold: Provides the basic structure of the visual interface.
2. AppBar: Displays the app bar at the top.
3. Column: Organizes child widgets in a vertical column.
4. TextField: Allows the user to input text.
5. ElevatedButton: A Material Design raised button.
6. GestureDetector: Detects gestures, like taps, to trigger actions.
7. SnackBar: A small message at the bottom of the screen.

SignIn Screen:

1. MaterialApp: Wraps the entire app. Manages the app's material design.
2. Scaffold: Provides the basic structure of the visual interface.
3. Center: Centers its child widget within the parent.
4. SingleChildScrollView: Allows the child to be scrolled when the keyboard is open.
5. Container: A box model to contain and organize child widgets.
6. Column: Organizes child widgets in a vertical column.
7. Text: Displays text on the screen.
8. Image.asset: Displays an image from the assets.
9. TextField: Allows the user to input text.
10. ElevatedButton: A Material Design raised button.
11. GestureDetector: Detects gestures, like taps, to trigger actions.
12. Navigator: Manages the navigation stack.
13. SnackBar: A small message at the bottom of the screen.

Intro Screen or Home Screen:

1. MaterialApp: Wraps the entire app. Manages the app's material design.
2. Scaffold: Provides the basic structure of the visual interface.
3. PageView: Allows swiping between pages.

4. Column: Organizes child widgets in a vertical column.
5. Text: Displays text on the screen.
6. ElevatedButton: A Material Design raised button.
7. Container: A box model to contain and organize child widgets.

Add Truck Screen:

1. Scaffold: Provides the basic structure of the visual interface.
2. AppBar: Displays the app bar at the top.
3. Form: Allows grouping of form-related widgets.
4. TextFormField: A form field that allows the user to enter text.
5. DropdownButtonFormField: A form field that allows the user to select from a dropdown.
6. ElevatedButton: A Material Design raised button.
7. SnackBar: A small message at the bottom of the screen.

Schedule Truck Screen:

1. Scaffold: Provides the basic structure of the visual interface.
2. AppBar: Displays the app bar at the top.
3. Column: Organizes child widgets in a vertical column.
4. Text: Displays text on the screen.
5. CalendarDatePicker: A date picker widget.
6. TimePicker: A widget that allows the user to pick a time.
7. ElevatedButton: A Material Design raised button.
8. SnackBar: A small message at the bottom of the screen.

Truck Utilisation Screen:

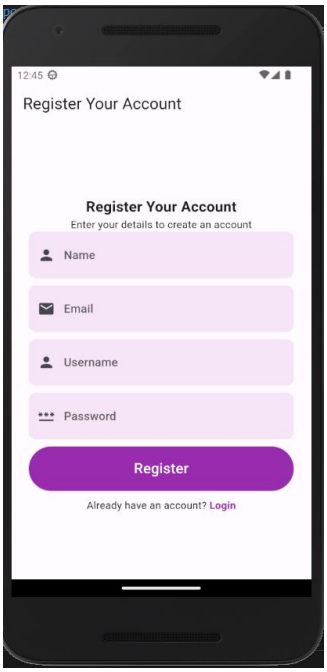
1. Scaffold: Provides the basic structure of the visual interface.
2. AppBar: Displays the app bar at the top.
3. Row: Arranges child widgets in a horizontal row.
4. Image.asset: Displays an image in the app bar.
5. Text: Displays text on the screen.
6. SingleChildScrollView: Allows the child to be scrolled when there is limited space.
7. Column: Organizes child widgets in a vertical column.

8. LinearPercentIndicator: Displays a linear progress indicator with percentage.
9. FutureBuilder: Represents a widget whose content depends on the result of a Future.
10. http: Makes HTTP requests.

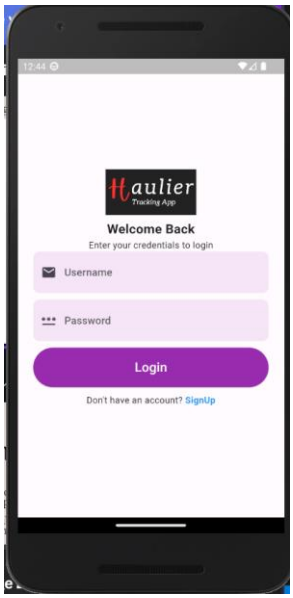
5.0 Sample of Interface

There are six screen for Haulier Tracking App which are Home Screen, Add Truck Screen, Schedule Truck Screen, Truck Utilisation Screen, Sign In Screen and Sign Up Screen. These are the sample of the interface for the application:

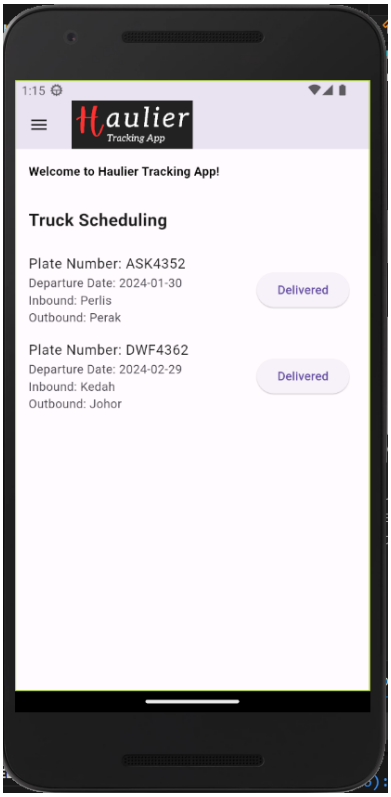
Sign Up Screen:



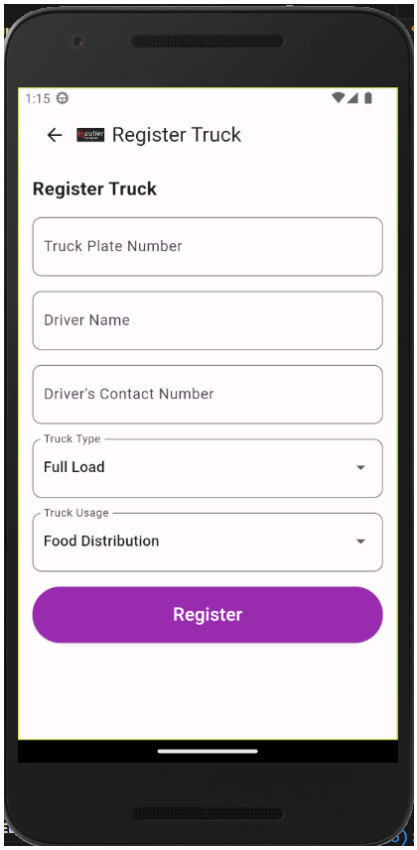
Sign In Screen:



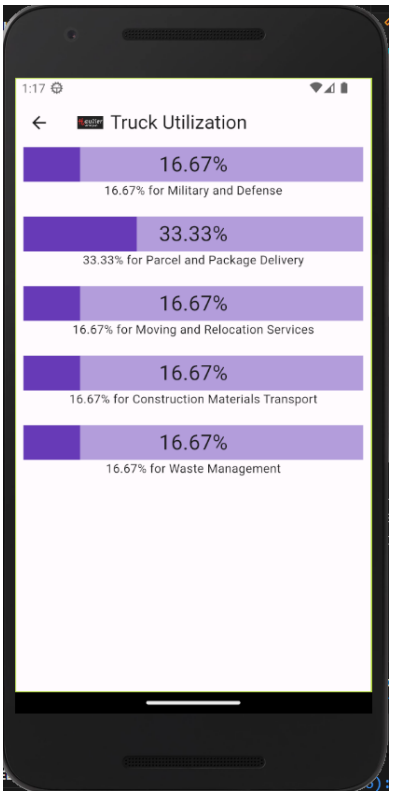
Home Screen:



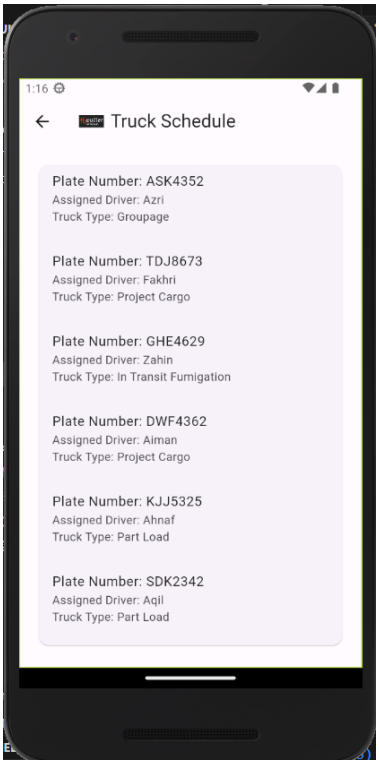
Add Truck Screen:



Truck Utilisation Screen:



Truck Schedule Screen:



6.0 Conclusion

In conclusion, the Haulier Tracking App might be a game-changer for companies engaged in logistics and transportation. With its comprehensive program, which addresses typical problems like misunderstanding, inefficiency, and delivery delays, you may improve the efficacy of cargo monitoring and align truck operations management.

With truck managers and enterprises in mind, this program has an easy-to-use interface. It provides informative data for effective logistics operations management by including sophisticated features like schedule changes and truck additions, going beyond basic tracking.

7.0 Reference

1. CHALLA, A. (2016). *FLEET MANAGEMENT SYSTEM* [Review of *FLEET MANAGEMENT SYSTEM*].
2. Flutterdynasty. (2023, September 29). *Login/SignUp Page in Flutter*. Medium. https://medium.com/@flutterdynasty/login-signup-page-design-with-code-c593e7c82a4c
3. Spark, M. (2020). *[Solved] Packages have newer versions incompatible Error*. Flutter Campus. https://www.fluttercampus.com/guide/288/packages-have-newer-versions-incompatible-error/#google_vignette
4. Ali, M. (2023, February). *Add Image to AppBar in Flutter App Tutorial | Build Siri & Alexa Clone using Open AI ChatGPT & DaleE*. Www.youtube.com. https://youtu.be/88_rNtHbv1A
5. koko, mitch. (2021). *CircularPercentIndicator/lib/homepage.dart at master · mitchkoko/CircularPercentIndicator*. GitHub. https://github.com/mitchkoko/CircularPercentIndicator/blob/master/lib/homepage.dart
6. https://chat.openai.com/c/65872e18-5d1f-4e32-a051-8dc9e66e5aef
7. Koko, M. (2021, November 3). *Really Cool Animated Icons • Flutter Widget of the Day #23*. Www.youtube.com. https://www.youtube.com/watch?v=j6f8lTV6D50
8. Lab 7 and Lab 9 and class activity which is shopping_list
9. https://chat.openai.com/c/8489b6ce-deb9-41fb-9efc-3360cc67a879
10. https://chat.openai.com/c/8defd2e2-1dd3-4a3a-9cc9-72acf8326513
11. https://chat.openai.com/c/98c880b5-7aa4-4393-951c-d1dccafd9b56
12. https://chat.openai.com/c/2933a77e-9938-4f5e-ac05-5580bd5f0c77