



UNIVERSITI MALAYSIA TERENGGANU

CSM3114 FRAMEWORK-BASED MOBILE APPLICATION DEVELOPMENT (K1)

SEMESTER I 2023/2024

**BACHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING) WITH
HONOURS**

**FINAL REPORT
“TOURIST HELPER APPLICATION”**

**PREPARED BY:
NUR FATHIAH IZZATI BINTI MOHD RODI**

**PREPARED FOR:
DR MOHAMAD NOR HASSAN**

Contents	
Executive summary	3
Use Case	4
The common structure of tree widgets you used when designing and developing application	5
Flutter widget and features adopted in the application	9
Sample of interface with explanation	11
Conclusion	18
Reference	18
Link GitHub	18

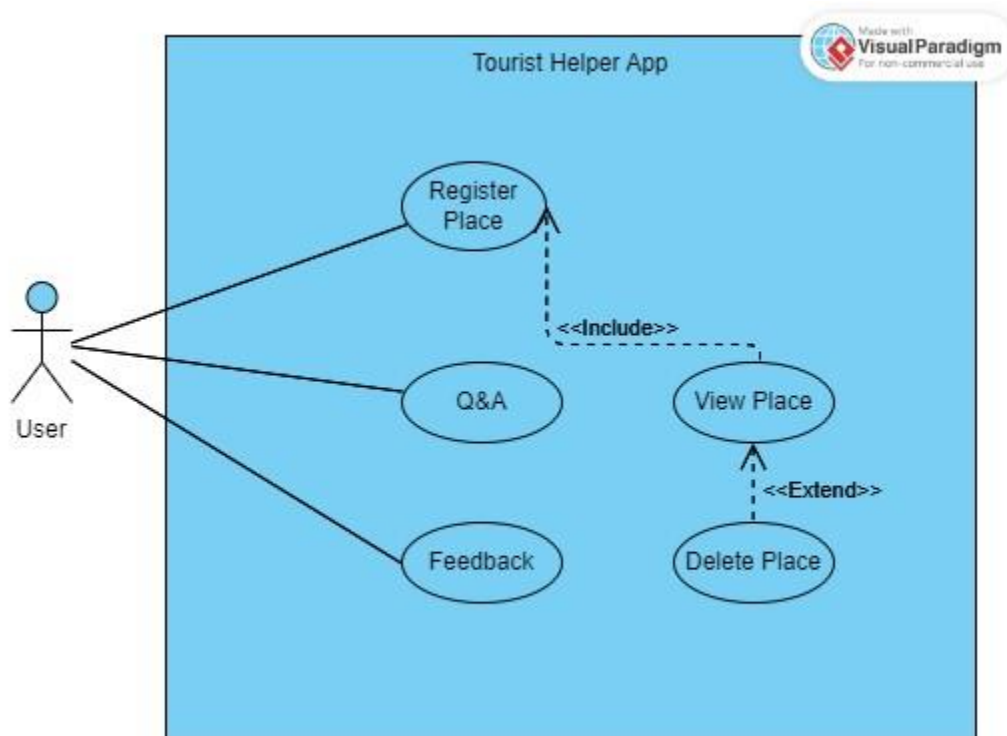
Executive summary

This application's main objective is to give users a comprehensive travel experience by allowing them to have meaningful conversations, ask for guidance, and exchange views, all the while promoting an organized method of choosing and overseeing tourist attractions.

Simultaneously, the Tourist Place Registration system emphasizes teamwork and carefully selected travel experiences by providing users with a simplified interface for registering, exploring, and managing tourist places. The application was created with the help of the cross-platform framework Flutter and follows Material Design guidelines to provide a consistent user experience on different devices. This strategy highlights the app's dedication to user-centric design, intuitive navigation, and effective functioning.

The application caters to a variety of travel requirements and positions itself as a vital resource for travellers worldwide. Tourist Helper is well-positioned to adjust, broaden, and reimagine travel-related applications as the digital world develops further, demonstrating its importance and worth in promoting more fulfilling travel experiences.

Use Case

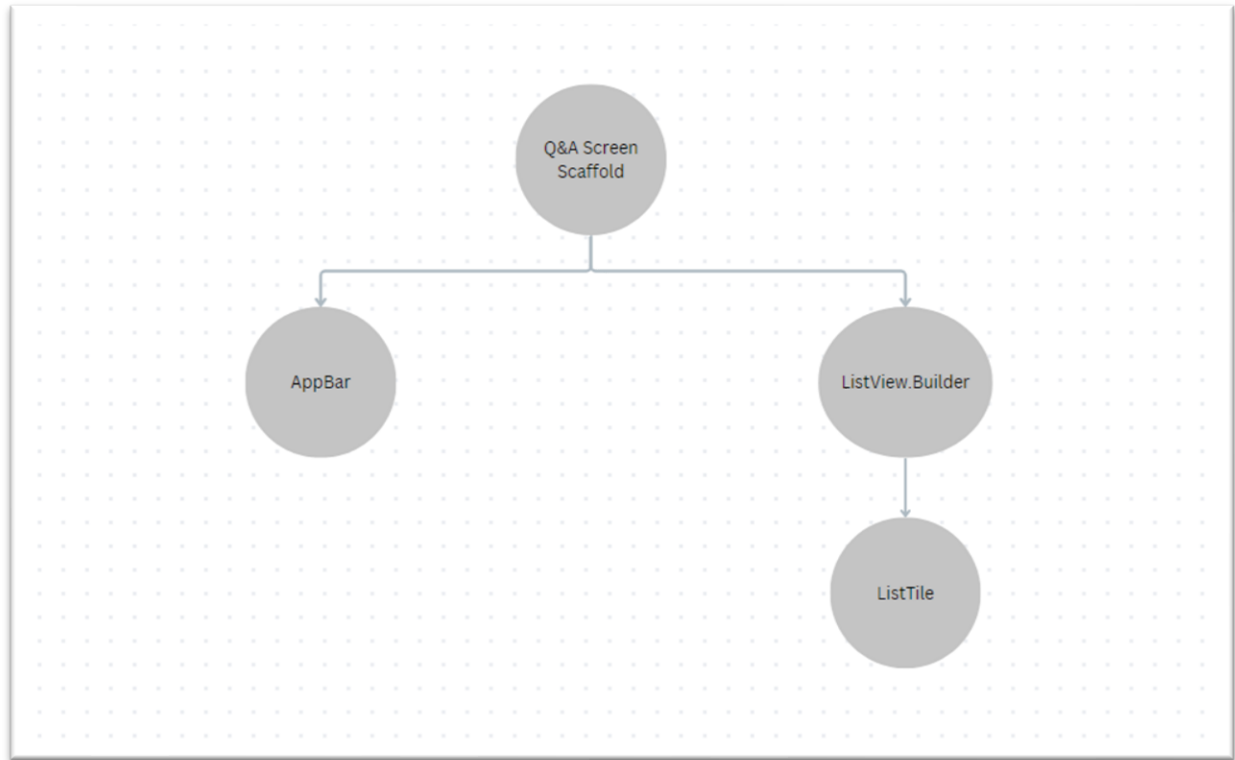


Q&A Functionality: Users can view a list of questions related to travel and tourism. By tapping on a specific question, they can navigate to a detailed view where they can read existing answers or submit their own answers.

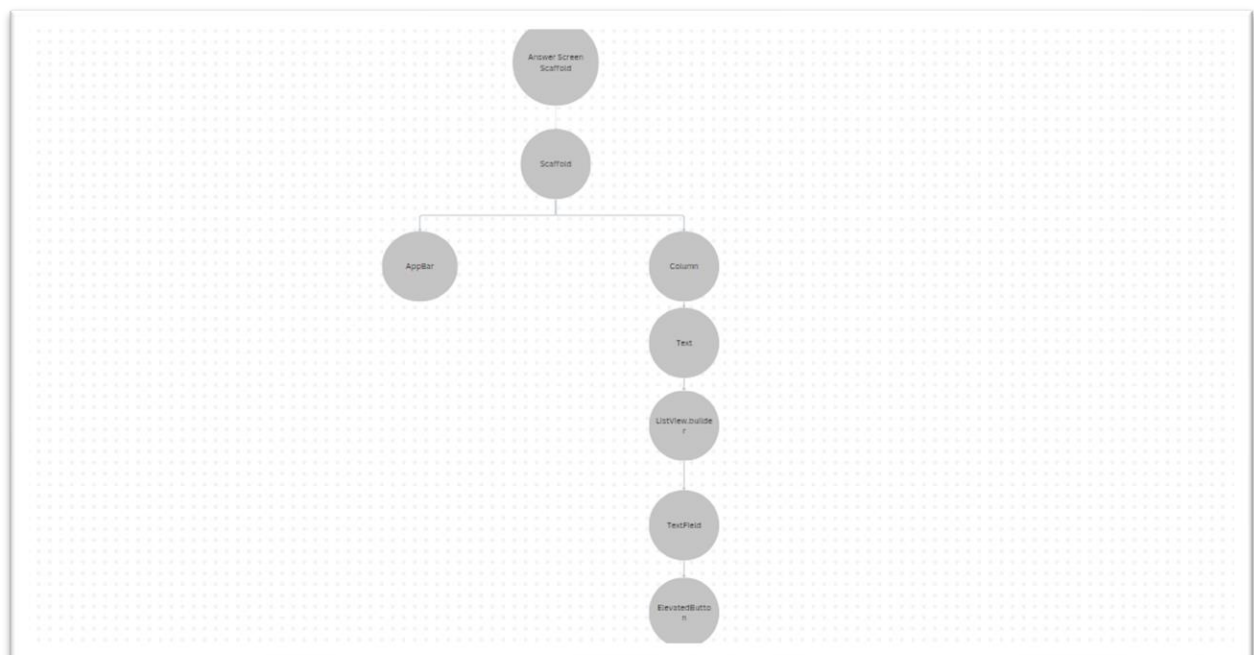
Tourist Place Registration and Viewing: Users can register new tourist places by providing details like place name, description, and location. Additionally, they can view a list of registered places and have the option to delete any of them.

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

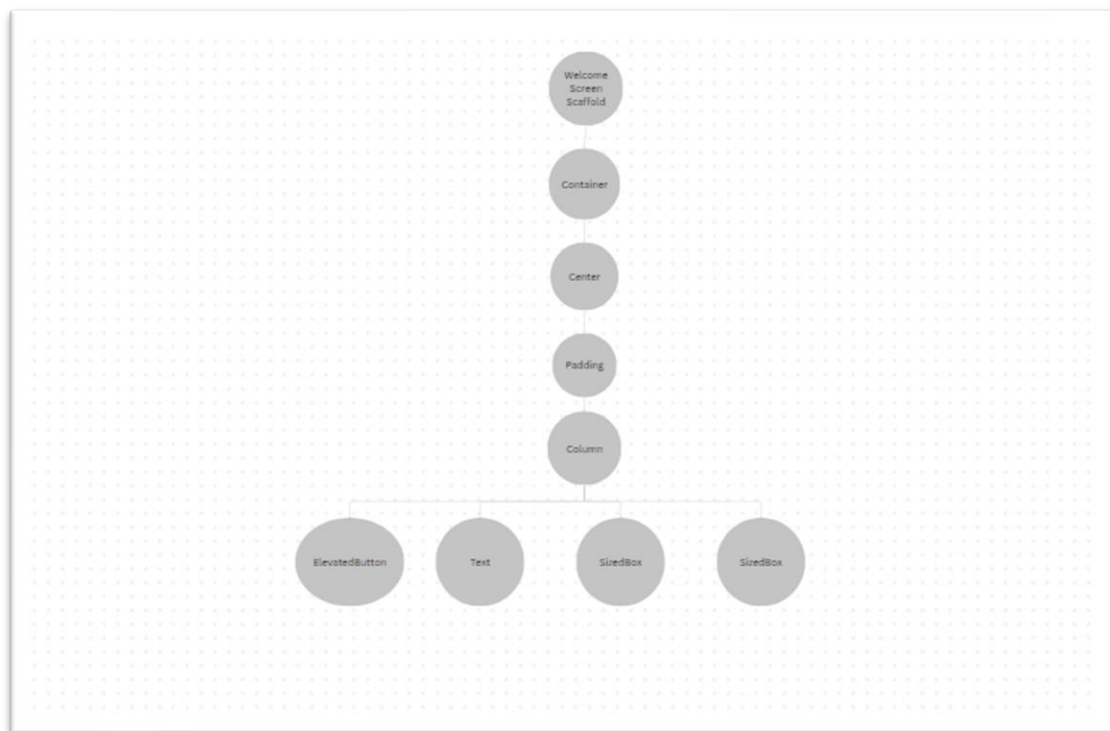
1. Q&A Screen



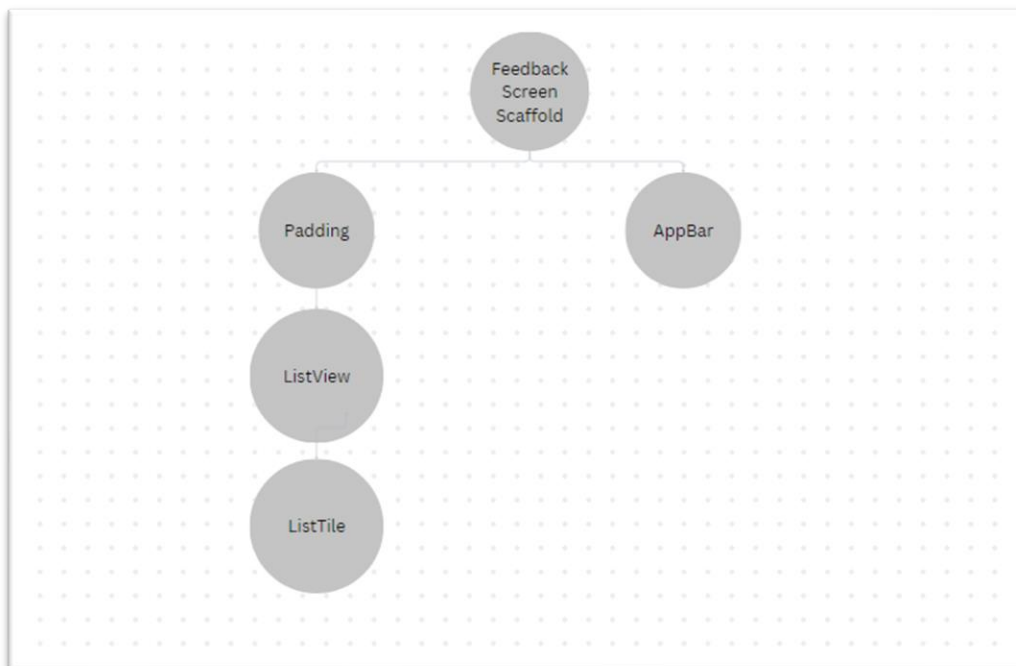
2. Answer Screen



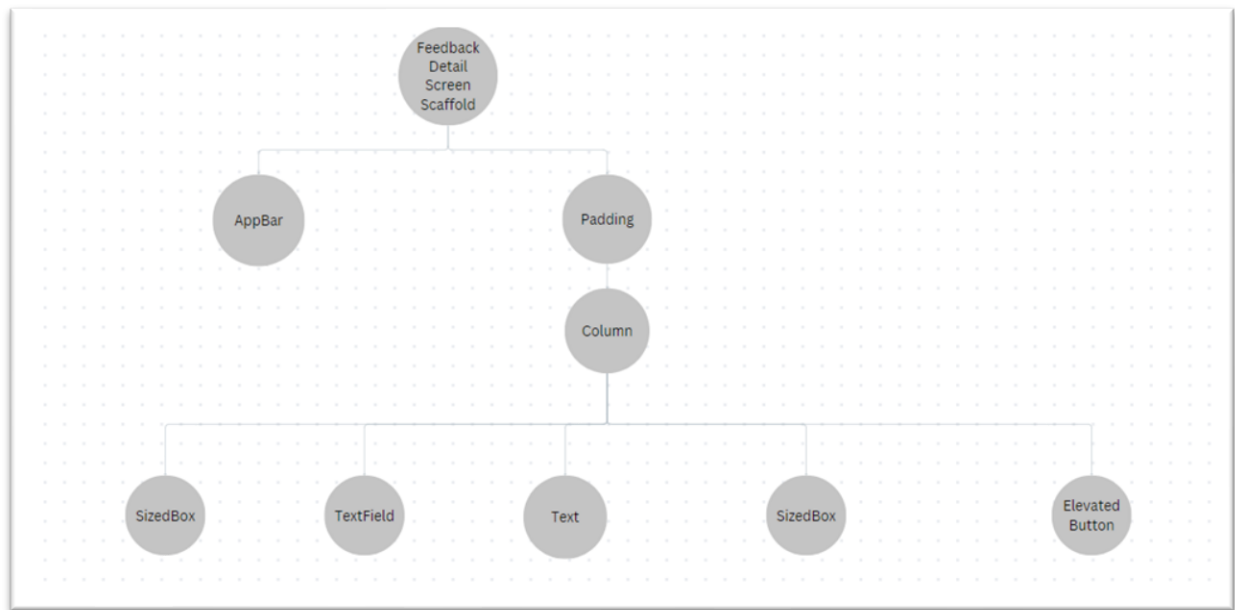
3. Welcome Screen



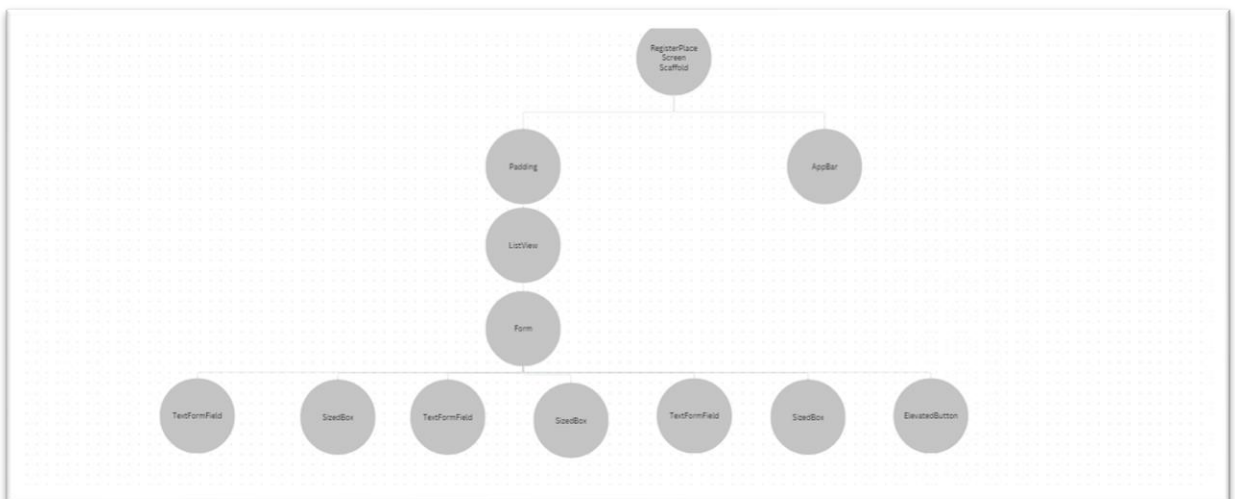
4. Feedback Screen



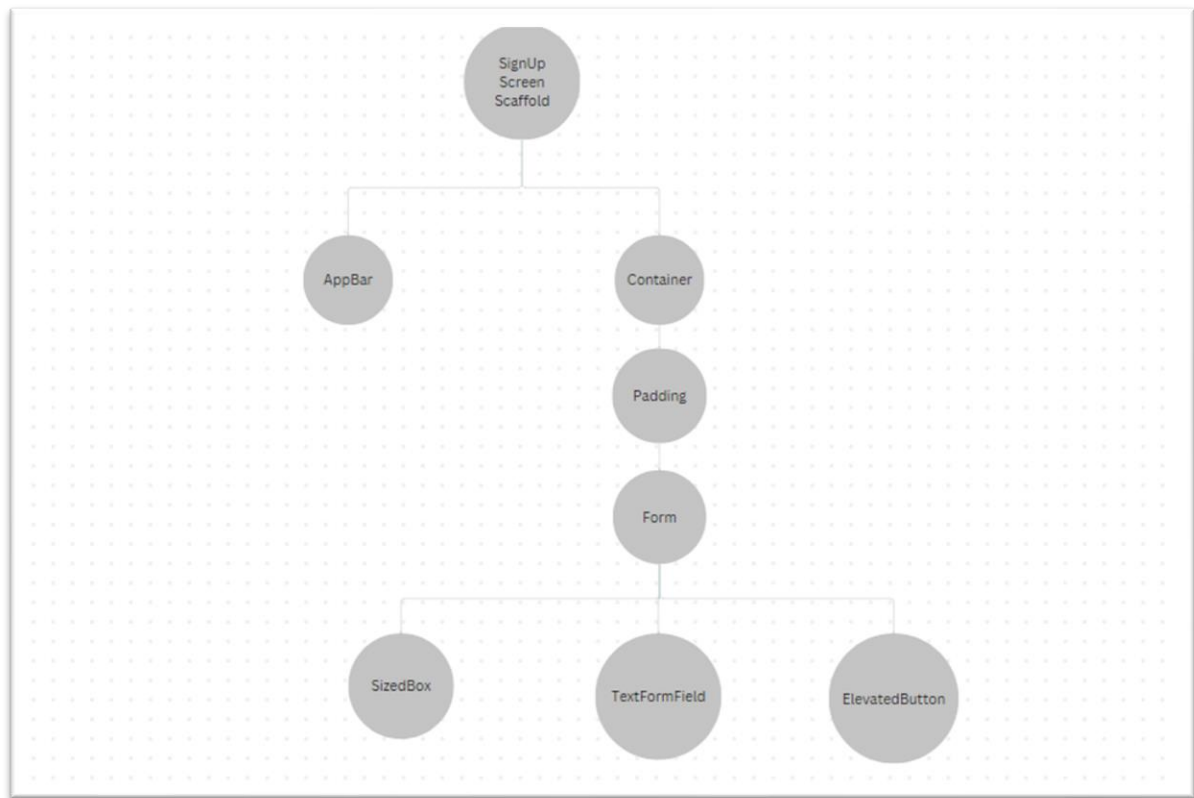
5.Feedback Detail



6. Register Screen



7.SignUp Screen



Flutter widget and features adopted in the application

Material Design Principles

The usage of Material Design as one of the fundamental components of the Tourist Helper program guarantees a unified and user-friendly user interface (UI) on both the Android and iOS platforms. The application prioritizes user experience by following known design patterns and rules, while simultaneously maintaining a visually appealing aesthetic thanks to the utilization of Flutter's built-in Material components.

Navigation Framework

User experience must be seamless, and the application makes use of Flutter's powerful navigation features to accomplish this. By using `MaterialPageRoute` with the `Navigator` widget, Tourist Helper makes it possible for users to move between different sections—like Q&A forums, place registration forms, and feedback categories—without experiencing any technical difficulties or inconsistent layouts.

HTTP Requests and Firebase Integration

To guarantee data synchronization in real time and effective data management, Tourist Helper makes use of HTTP requests, most especially the `http` package. With the help of this connection, users can easily communicate with the Firebase Realtime Database and perform tasks like retrieving current place data, adding new tourist destinations, and removing unnecessary or out-of-date records. Within the application ecosystem, this continuous connection fosters confidence and reliability by guaranteeing users have access to the most recent information.

Form Validation Mechanisms

The program has strong form validation features because it understands how important user input validation and data integrity are. Tourist Helper enforces strict validation requirements for location registration forms by utilizing Flutter's `GlobalKey<FormState>` feature. By guaranteeing users supply correct and comprehensive data, this improves the quality of data saved and cultivates a more organized and dependable database.

Implementation of SnackBar

Strategic integration of the SnackBar widget within the Tourist Helper application underscores its commitment to enhancing user experience, facilitating real-time feedback, and streamlining error handling mechanisms. Through its multifaceted functionalities—from providing immediate feedback to enabling interactive responses—the SnackBar widget serves as a cornerstone element, elevating the overall usability and responsiveness of Tourist Helper.

Sample of interface with explanation



Figure 1 Welcome page

Figure 1 defines a Welcome screen that displays a background image, a welcome message, and a button. When the button is pressed, the user is navigated to the LoginPage. The design follows a structured layout using Flutter widgets to create a visually appealing welcome interface for the app users.

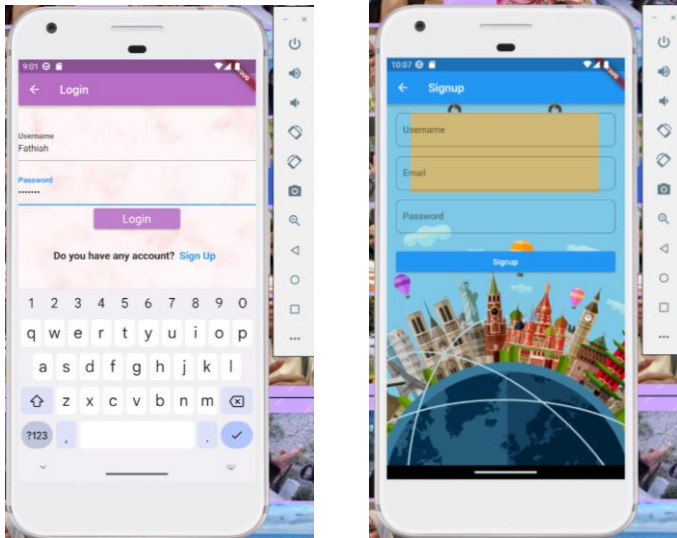


Figure 2 and 3

Figures 2 and 3 show that the 'SignupPage' allows users to register using a username, email, and password, submitting this data to a Firebase database upon validation. Once registered, they're redirected to the 'LoginPage'. Conversely, the 'LoginPage' verifies user credentials against the Firebase database using a GET request. Valid users proceed to the home page, while invalid attempts trigger an error. Additionally, the 'LoginPage' provides a "Sign Up" link for new users, ensuring smooth navigation and form management through Flutter's state system.

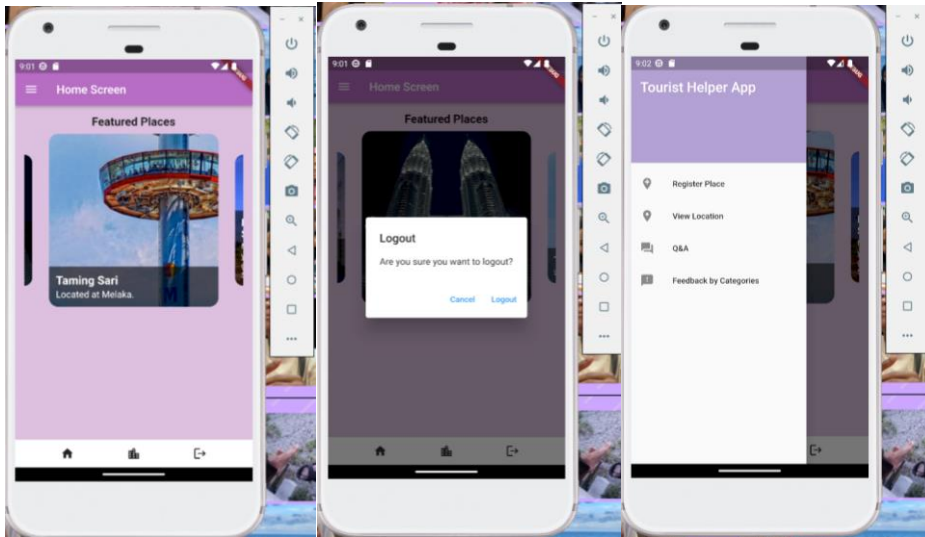


Figure 4, 5, and 6

Figure 4, 5, and 6 defines the primary interface of a "Tourist Helper App," structured around a HomeScreen widget. This home screen presents a carousel of "Featured Places," including images and brief descriptions of notable tourist locations. Users can navigate to different sections of the app using a bottom navigation bar featuring icons for home, location registration, and logout functionalities. Additionally, a drawer menu provides further navigation options such as registering a place, viewing locations, accessing Q&A sections, and providing feedback. The app maintains a cohesive design with themed colors, interactive elements, and navigational paths to enhance the user experience while exploring tourist-related content.

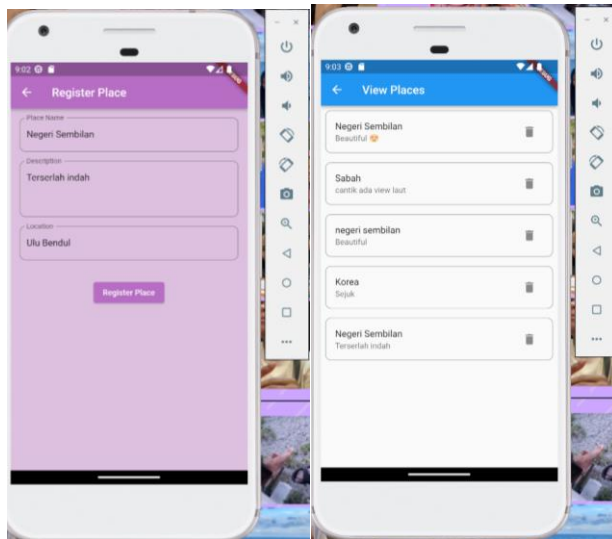


Figure 7 and 8

The Register class allows users to input details of a place, including its name, description, and location. Upon form submission, this data is sent as a POST request to a Firebase database. If the registration is successful, users are redirected to the ViewPlace class.

The ViewPlace class fetches and displays a list of registered places from the same Firebase database. Initially, a loading indicator is displayed while the data is being fetched. Once retrieved, each place's details, including its name and description, are displayed in a scrollable list. Additionally, users have the option to delete any listed place, triggering a DELETE request to the database to remove the respective entry.

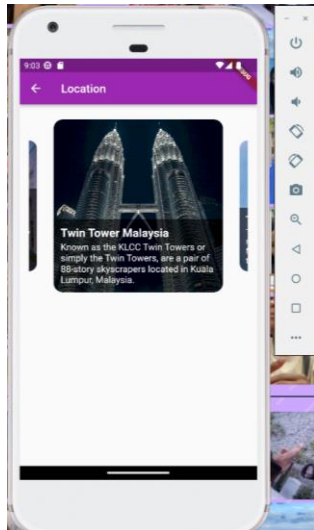


Figure 9

Figure 9 defines a Flutter Location widget that showcases a list of featured places using a carousel slider. Each featured place is represented by a map containing its title, image path, and description. The carousel displays these places one at a time, allowing users to scroll through them horizontally. Each slide in the carousel displays an image of the location, overlaid with a semi-transparent black background at the bottom, which contains the title and description of the featured place. The carousel is configured with specific options like auto-playing, enlarging the center page, and adjusting the viewport fraction for better presentation. Overall, the widget offers an interactive and visually appealing way to highlight various tourist locations within the app interface.

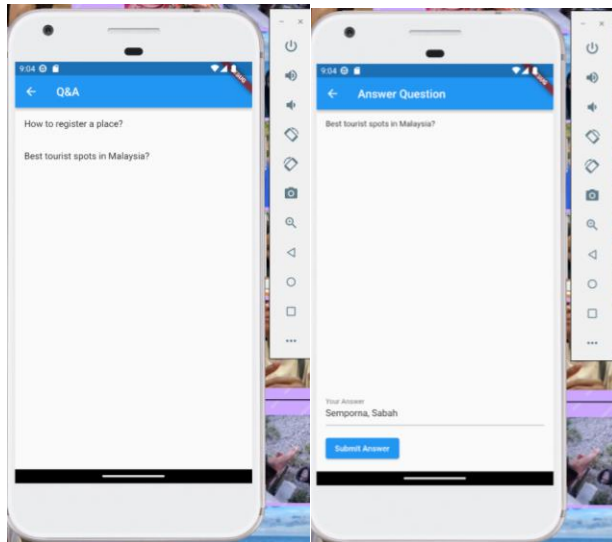


Figure 10 and 11

Figure 10 and Figure 11 implements a Question and Answer (Q&A) feature within a Flutter application, structured around the 'QA' and 'QAScreen' classes. The 'QA' class encapsulates a question and its associated answers. The 'QAScreen' displays a list of questions, and upon selection, it navigates to an 'AnswerScreen' where users can view existing answers and submit new ones. This design allows for interactive Q&A sessions, with answers dynamically updating the UI upon submission, thereby facilitating an engaging user experience.

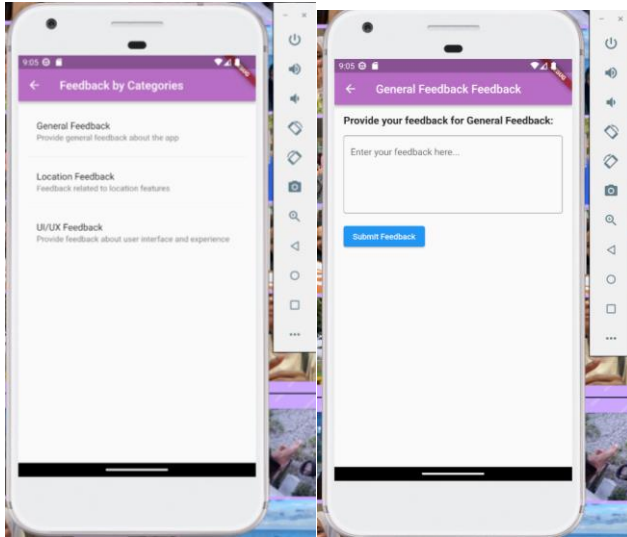


Figure 12 and 13

Figure 12 and 13 presents a feedback system categorized by specific topics within the application. The `'FeedbackScreen'` class displays a list of feedback categories like General Feedback, Location Feedback, and UI/UX Feedback. When a user selects a category, it navigates to the `'FeedbackDetailScreen'` where users can provide detailed feedback related to the chosen category. The feedback is collected through a text field, and upon submission, the user is returned to the previous screen, enabling a structured and user-friendly approach to gathering user insights and suggestions for improvement.

Conclusion

The "Tourist Helper" application amalgamates two essential functionalities: a Q&A platform for travel-related queries and a system to register and view tourist places. By leveraging Flutter's robust widget system and features, the app ensures a smooth user experience. Future iterations could focus on enhancing user interactions, integrating more advanced features, and expanding the database to include user reviews and ratings for tourist places.

In essence, the "Tourist Helper" application emerges as a pivotal tool, seamlessly connecting travel enthusiasts with invaluable insights and destinations. Leveraging Flutter's dynamic widget system and advanced features, the platform delivers a fluid and intuitive user experience, setting a robust foundation for future innovations. Moving forward, opportunities abound for enhancing user engagement through the integration of interactive maps, real-time recommendations, and community-centric features. As "Tourist Helper" continues to evolve, it stands poised to redefine travel experiences, fostering community collaboration, and empowering users with informed decision-making capabilities.

Reference

1. *Flutter documentation* Flutter. Available at: <https://docs.flutter.dev/> (Accessed: 30 December 2023).
2. (2021). *YouTube*. Retrieved January 5, 2024, from https://youtu.be/5dX3mnYJS_Y?si=qOHqerTU8vxjGeLB.
3. *ChatGPT*. (n.d.-b). ChatGPT. Retrieved January 2, 2024, from <https://chat.openai.com/share/10a8379f-d88b-4e22-9bec-5e9be7145dc6>.
4. *Add Carousel in Dart*. (n.d.). ChatGPT. Retrieved January 2, 2024, from <https://chat.openai.com/c/f2c48d3d-7089-479f-b883-0720911ac1a7>

Link GitHub

https://github.com/FathiahIzzati/tourist_helper