

Appendix D

Tunku Abdul Rahman University of Management and Technology BAIT2073 Mobile Application Development

Task Description

Name : Lim Jun Wei
Programme : RSD

ID : 24WMR09078
Group : G5

Instruction: Answer **ALL** the questions. You may insert extra pages.

1. Please briefly describe the module(s)/function(s) you engaged in the assignment.

I have taken the role of developing the **user modules** and **payment modules**. In the user modules, I have implemented the register functionalities. Within the registration function, I have integrated with the SMTP to realize the **email verification by sending OTP** to the user's mailbox. When dealing with **login and logout** functions, I have integrated those functions with the Provider and FlutterSecureStorage for maintaining the logged in user account. This can prevent users from repeatedly logging into the app every time they restarted their apps. On the other hand, within the user profile editing function, I have developed the **user profile image uploading function via selecting images from the user device's gallery and capturing images using the user device's built-in camera**. The additional image processing has been implemented for minimizing the image size and standardizing the image file type before passing the data to the backend server via FastAPI. If users have accidentally forgotten the password, the **"Forgot Password" mechanism** is also created for prompting the users their email address, verifying their email address via OTP and allowing them to reset their password.

In the **payment module**, I have developed the **payment history list** which allows the users to view their pending, unpaid or paid payment with various filters and sorting. Not only that, the **Stripe API** is also integrated into the system for bringing the user to the sandbox payment gateway API for completing the payment for each booked appointment. The Stripe API implemented is able to support 3 types of payment methods such as Card, FPX and GrabPay. Once the user has completed their payment successfully, I have developed the function for allowing the users to **view and download their invoice in PDF format**. Additionally, I have also integrated the application with the Google Firebase Messaging technology to allow the device to push notifications once the users have made the payment successfully. If the user has completed the appointment but not yet paid for it, the application will automatically check the unpaid payment by the users daily and push the notification to **remind the users to pay the appointment payment** at 10 a.m. every day.

2. What are the strengths of the modules/functions created by you?

From the aspect of the user module, one of the potential strengths is the application can always **keep the user in logged in state** once the user has been logged into the system. Via implementing the Provider and FlutterSecureStorage, the users can directly revisit the application content every time without requiring the repeatedly logging into the system using the same account after quitting or restarting from the system. On the other hand, I have applied the **hashing process when dealing with the users' registered account password**. In the backend process flow, when the user is registering their new account with password, the system will make sure that the password is hashed before storing into the Supabase cloud database. This can effectively protect the user account security and prevent the user account password from being exposed to the intruders. When it comes to user profile image uploading, the Flutter frontend site will **optimize the image size by appropriately dropping the image quality and check for the uploaded image format** before passing the data to the Python backend site via FastAPI. This can efficiently maximize the

Appendix D

performance of passing data to backend while standardizing the uploading image format. Thus, only the minimum storage capacity will be taken for storing the user profile images into the Supabase cloud database when maintaining the consistent way of storing the image files.

From the aspect of the **payment module**, the potential strength is the **Stripe API has been integrated into the system**, instead of developing the payment gateway using its own codes (more risky). This can help for protecting the user's sensitive payment information such as card number and passwords. Thus, no sensitive data will be easily stolen by the intruders since all the payment information and handling will be handled by the Stripe API. On the other hand, the users who have booked the appointments sometimes will forget to pay for them. In order to prevent this situation, the payment module has an **automatic mechanism to regularly check for the currently logged in user' unpaid payment for the completed appointments**, then push the notification to the user device every day at 10 a.m.

3. What are the weaknesses of the modules/functions created by you?

In the user profile editing function, the app does **not provide the mechanism for automatically configuring a user's location address** by locating their current location. The users have to manually key in their long location address which may require more human effort.

When it comes to notifications features, the current **notification pushing feature is only available when users are accessing the app or putting the application in background**. Else, the notification feature will not be triggered successfully.

Since the current Stripe API applied into the application is in a sandbox environment, it is **not available for making the real transaction and connecting to a real payment gateway**. The users can only enter the default specified payment details or click the "Authorize payment" or "Failed payment" button to simulate the payment process as a user view.

4. What have you learned in doing this assignment?

When dealing with the assignment, I have learned how the **Flutter framework works using the Dart programming language** and how these codes are being implemented and displaying the results on the screen of virtual or real mobile devices. Besides, I have also learned how the **Flutter framework (frontend) and Python (backend via FastAPI) are collaborating with each other to perform the application features** on the devices. Meanwhile, I have also learned how to configure and establish the connection between Flutter and Python to achieve a seamless "communication" between each other.

Moreover, I have learned how the Python backend code can be **linked with the Supabase cloud database** to store all the information into the cloud database. Not only that, I have also got to know how to **design the logic of navigation flow** from page to another page in the application. I also learned how the Flutter application **stores and "remembers" users account data** using Provider and FlutterSecureStorage so the users logged in account can always be maintained without requiring them to repeatedly log into the application. I have also learned how to **develop the notification pushing on mobile devices** in Flutter using Google Firebase Messaging.

From the communication between Flutter and Python, I have learned how to **establish the API requests** for allowing the Flutter to call API requests to the Python backend and the Python backend to respond on the request back to Flutter. Meanwhile, I also **get more familiar with designing the Model-View-Controller**

Appendix D

(MVC) code structure with the collaboration between Flutter and Python.

5. What are the challenges, if any, faced by you while working on this assignment?

When dealing with the database, I am struggling with the type of the database that is ideal for the assignment application. I am struggling with choosing the **relational database (Supabase) or NoSQL database (Google Firebase)**. Relational databases can provide more consistent and manageable database data while NoSQL databases can offer more flexible database data by storing flexible types of data in the database. Due to the consideration of the requirements for easier and standardized management of the database, I eventually **decided to apply the Supabase** which is the relational database as allowing for applying SQL statements and establishing the linking between different tables using primary and foreign keys.

When it comes to backend handling, I am struggling with how I should **implement the backend code which can collaborate with the Flutter framework** since there is lack of knowledge and experience of using different languages on the frontend and backend of a mobile application. Eventually, I solved this issue by **implementing the FastAPI to "link" the Flutter framework with Python using RESTful API and applying the MVC structure within both frontend and backend**. Eventually, the Flutter application can effectively communicate with the Python backend via sending requests to the backend server and waiting for its response.

Signature:

Date: 17/9/2025