

## Sleeping Coach Mobile Application

Soo Xin Yi<sup>1</sup>, Nayef Abdulwahab Mohammed Alduais<sup>1\*</sup>

<sup>1</sup>Faculty of Computer Science and Information Technology,  
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

DOI: <https://doi.org/10.30880/aitcs.2024.05.01.041>

Received 24 June 2023; Accepted 18 May 2024; Available online 30 August 24

**Abstract:** Mobile-based sleeping coach application was developed for the insomniac. This project was to assist insomniac when they are starting hard to falling asleep and staying asleep. Insomniac can be more easily to getting asleep with follow the sleep coach plan and using tools. Besides that, administrator module allows admin to edit sleep coach plans and sleep tests. This project was developed with Waterfall model, the chosen development tools to build up the system were Flutter, Visual Studio Code, and Firebase as database. The developed application provided several modules included administrator module which allow admin to edit plans, and tests, while user can choose the plan, answer sleep tests to get recommend plan, use tools, view the sleep diary, and produce report. Since the application did not have check list function and notification function, thus for future implementation, the application shall implement check list function, and notification function.

**Keywords:** Insomnia, CBT-I, mobile application, sleep coach

### 1. Introduction

Mobile-based sleeping coach application is a mobile-based system, and it is developed especially for the insomniac. Insomnia is a sleep disorder that will make hard to sleep, stay asleep, or cause wake up too early and cannot fall asleep again. It usually will decrease the energy level, mood, and cause the health, work performance, and quality of life become poor [1]. This project is proposed to assist insomniac when they are starting hard to falling asleep and staying asleep. Insomniac will be more easily to getting asleep with the train a regular daily pattern, and use the tools to help in fall asleep.

Firstly, the aim of this project is to develop a mobile-based sleeping coach application which can help to improve the sleeping quality, and provide administrator module to allow admin edit the sleep coach plans, and sleep tests questions.

The objectives of this project are to design a sleeping coach application based on mobile-based approach, develop the mobile-based sleeping coach application, and implement the alpha and beta testing for the developed mobile-based sleeping coach application.

The scopes of this project are making the administrator module to edit the sleep coach plans, and edit sleep tests questions, and user can follow the sleep coach plan, answer the sleep tests, use tools,

---

\*Corresponding author: [nayef@uthm.edu.my](mailto:nayef@uthm.edu.my)

| This is an open access article under the CC BY-NC-SA 4.0 license.

view the sleep diary that display the sleep time chart and awake time, and produce the report of sleep time and awake time.

The expected result of this project is to allow user can follow the sleep coach plan, answer the sleep tests, use tools, view and get the report of sleep time and awake time, while admin can edit the sleep coach plans, and edit sleep tests questions.

## 2. Related Work

### 2.1 Literature Review

During the literature review, I have found that to develop a sleep coach application should use the CBT-I which is refer to Cognitive Behavioral Therapy – Insomnia method. To support the development of mobile application, the Flutter, Visual Studio Code, Dart programming are important. To build up the database management system, the Firebase is suitable to support the database of the mobile application. Besides that, to provide different sleep coach to the users, the users have to provide the behavior information, and get the recommendation sleep coach plan.

The sleep loss which also called Insomnia affect every significant system in the human body [3]. Cognitive behavioral therapy for insomnia (CBTI) is an effective treatment for insomnia. In-person CBTI is replaced by technology-based solutions and lifestyle modifications like physical activity (PA), which are both accessible and affordable self-management options [4]. According to American and European recommendations, cognitive behavioral therapy for insomnia (CBT-I) is the most effective behavioral therapy available [5]. To improve the quality of sleep, CBT-I tries to modify sleep hygiene education, stimulus control, sleep restriction, relaxation training, and cognitive therapy components [6].

According to American and European recommendations, cognitive behavioral therapy for insomnia (CBT-I) is the most effective behavioral therapy available [7]. To improve the quality of sleep, CBT-I tries to modify sleep hygiene education, stimulus control, sleep restriction, relaxation training, and cognitive therapy components [8].

The goal of cognitive therapy is to identify and change problematic sleep-related ideas and thought patterns, such as worrying. Then a series of techniques are used to get rid of unwanted sleep-related thoughts and irrational concerns. The three most popular techniques are thought-stopping, paradoxical intention, and cognitive restructuring.

In the final session which is evaluation and relapse prevention, a summary of all the elements is given, and post-measurements are conducted to assess the result. The intention is to stop the recurrence of symptoms brought on by sleeplessness [9].

### 2.2 Comparison of three existing applications and proposed system

One of three chosen existing application is Insomnia Coach Mobile Application which provide functions include the training plan which provide step to enhance sleep quality, the sleep coach which allow users to feedback about personal sleeping, the sleep diary to allow users observe the change in sleep, and seventeen tools which help users to get in sleeping [10].

The other existing application is SleepSpace Mobile Application which provides the functions include connect to smart bedroom which also is a special function compare with other sleeping coach applications [11].

The last of three existing application is CBT-I Coach Mobile Application which allow the users do the self-management to the sleep habit and sleeping time.

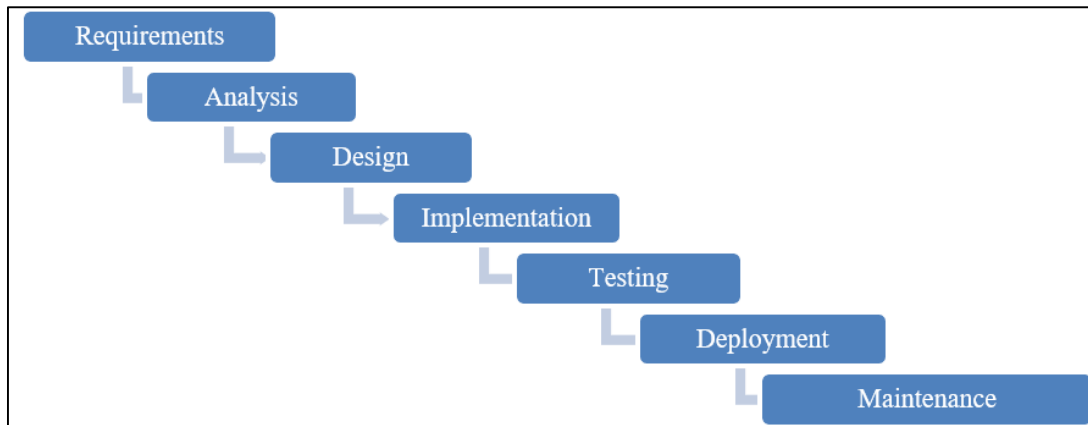
**Table 2.1: The comparison table between three existing applications and the proposed application.**

<b>Features</b>	<b>Insomnia Coach Mobile Application</b>	<b>SleepSpace Mobile Application</b>	<b>CBT-I Coach Mobile Application</b>	<b>Mobile-Based Sleeping Coach Application</b>
Admin Panel	Not available	Not available	Not available	Available
Operating System	iOS and Android	iOS and Android	iOS and Android	iOS and Android
Mobile-Based Application	Yes	Yes	Yes	Yes
Register and Login module	Not available	Available	Not available	Available
Sleep Time chart	Available	Available	Available	Available
Sleep Plan module	Available	Not available	Not available	Available
Allow to choose different plan	Not available	Not available	Not available	Available
Tools module	Available	Not available	Available	Available
Purchase	Free	Purchase in app	Free	Free

### 3. Methodology/Framework

#### 3.1 Methodology

The methodology chosen in this project was the waterfall methodology. Methodology is a system of methods that used in a particular area of study or activity. Waterfall is a linear and sequential approach model that used in system development cycle [12]. It usually involves seven stage which are requirement, analysis, design, implementation, testing, deployment, and lastly maintenance [13]. I choose waterfall model as my project methodology because it is simple to understand since I already have a clearly vision for my project. Since the requirements proposed system are well-known, clear, and fixed and need not to adapt to change rapidly, thus the waterfall model is suitable for this project.



**Figure 3.1: Waterfall model**

### 3.2 System Analysis and Design

The requirement analysis is the step to define the user expectation of a system before develop and modify. The analysis had been defined and the data flow is explained with the form of use-case diagram, sequence diagram, activity diagram, classes design, and the entity relationship diagram. This system need one user is enough because this system is concentrated in helping the user to improve the sleep quality with follow the provided sleep coach plan autonomously, and giving the user liberty to make their own decisions, and self-monitoring.

The requirement can be divided into user requirement and system requirement which contain functional requirement and non-functional requirement. Table 3.1 has explained the user requirements of the insomniac for the proposed system. Table 3.2 has explained the functional requirements for the proposed system. Table 3.3 has explained the non-functional requirements for the proposed system.

**Table 3.1: User requirements of insomniac**

No	User requirements
1	Users are able to login to the application.
2	Users are able to manage and update the profile details such as first name, last name, and phone number.
3	Users are able to observe the sleep time and awake time with chart form.
4	Users are able to use the tools to help in sleeping.
5	Users are able to view to follow the steps in plan.
6	Users are able to get the recommend plan.
7	Users are able to do the tests.
8	Users are able to receive the notification from alarm, and email report.
9	Users are able to logout the application.
10	Admins are able to login to the application.
11	Admins are able to create, read, update, and delete the steps of plans.

---

12 Admins are able to create, read, update, and delete the questions of tests.

---

13 Admins are able to logout the application.

---

**Table 3.2: Functional requirement for the proposed system**

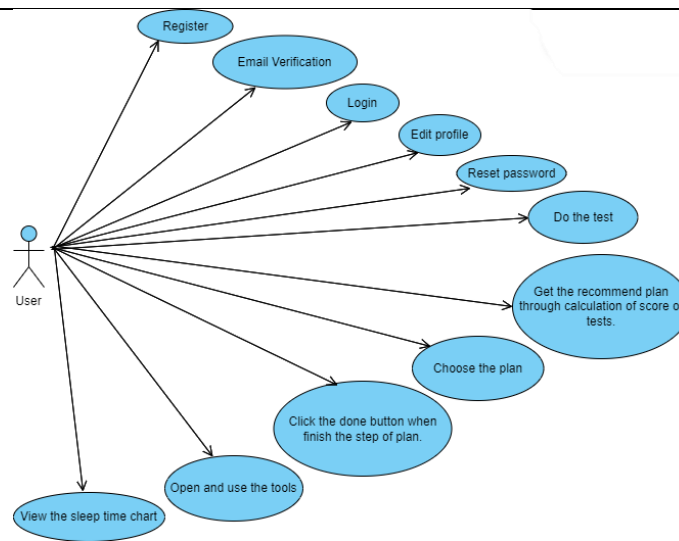
Function	Functionality	User
Input sleep tests	Answer the sleep tests question to get the recommendation of sleep coach plan and report of sleeping.	Insomniac

**Table 3.2: Functional requirement for the proposed system (continuous)**

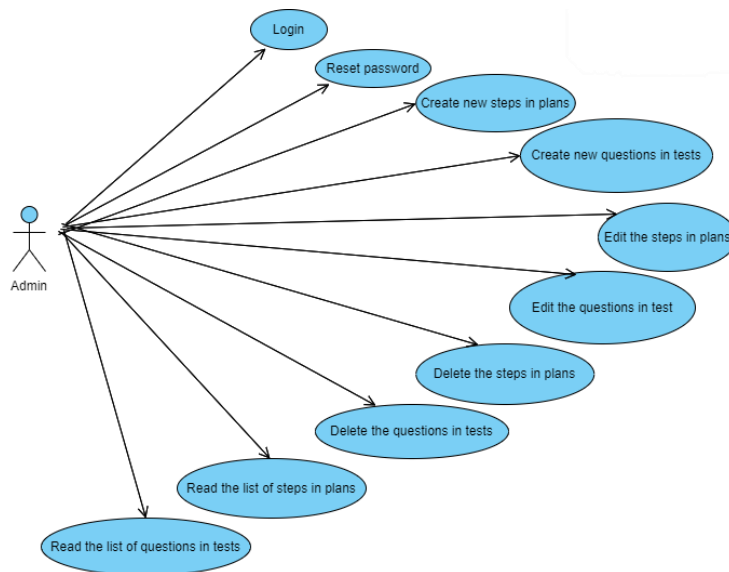
Function	Functionality	User
Manage and update profile	Update the account information such as first name, last name, phone number.	Insomniac
Check the sleep time and awake time	Check the chart that show the sleep time and awake time.	Insomniac
Reset the password	Send the email link to allow the user to reset the password.	Insomniac
Use sleep tools	Open and use the sleep tools for helping in falling and staying asleep.	Insomniac
Receive email verification	Send the request for get sleeping report, then receive the report from the email.	Insomniac
Select sleep coach plan	Select the sleep coach plan according own selection or recommendations.	Insomniac
Login	Allow user to login via using email, password.	Insomniac
Register	Allow to register as a user of system via using email, password, security phase, email, gender, age, role, and phone number.	Insomniac
Edit plans	Allow admins to create, read, update, and delete the steps of the plans.	Admin
Edit tests	Allow admins to create, read, update, and delete the questions of the steps.	Admin

**Table 3.3: Non-functionality requirement for the proposed system**

Requirement	Description
Security	<ol style="list-style-type: none"> <li>1. Information of user is secured.</li> <li>2. User need to enter the correct username, password to login.</li> </ol>
Operational	<ol style="list-style-type: none"> <li>1. This application can used in any mobile device such as Android and IOS.</li> <li>2. The system will have a database for the requirement to store all the data.</li> </ol>
Performance	<ol style="list-style-type: none"> <li>1. The system response time should be short.</li> </ol>
Usability	<ol style="list-style-type: none"> <li>1. The system is simple and easy to understand and use since the system is user-friendly.</li> </ol>



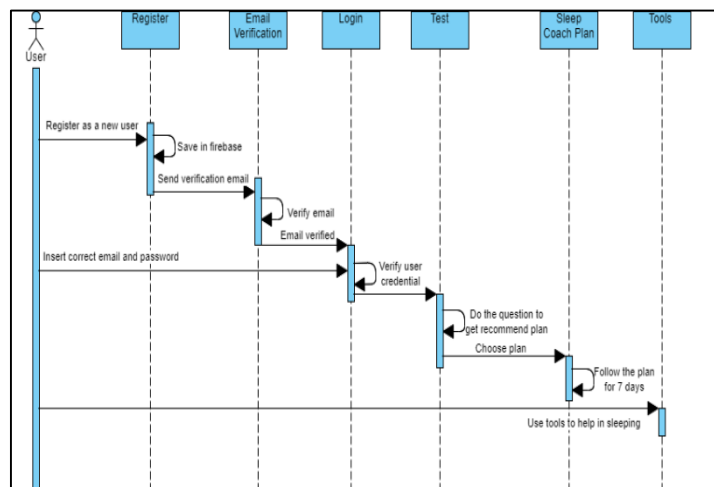
**Figure 3.2: User use-case diagram**



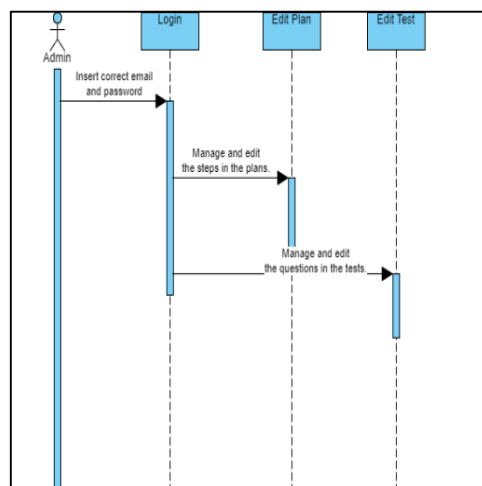
**Figure 3.3: Admin use-case diagram**

Figure 3.2 and 3.3 has shown the use-case diagram for the user and admin. There are eleven use case for the users. The use cases are register, login, email verification, edit profile, reset password, do the

test, get the recommend plan, choose the plan, click the done button when finish the step of plan, open and use the tools, and view the sleep time chart. There are eleven use case for the admins. The use cases are login, reset password, create, read, update, and delete the steps in plans, and the questions in tests.



**Figure 3.4: Sequence diagram for user**



**Figure 3.5: Sequence diagram for admin**

Figure 3.4 and 3.5 review the sequence diagram for user and admin. Users have to provide their own information to the system to register as a user of the proposed system. The information will save into the database. To login into system, the users need to provide the correct email, password, and security phase. After login into application, users have to complete the tests questions to get the recommendations sleep coach plan. After users choose the sleep coach plan, users have to complete the plan. Besides that, the users can view use the tools to help in sleeping. For admin, admin need to insert correct email and password to login. Then, admin can create, read, update, and delete the steps in plans, and questions in tests.

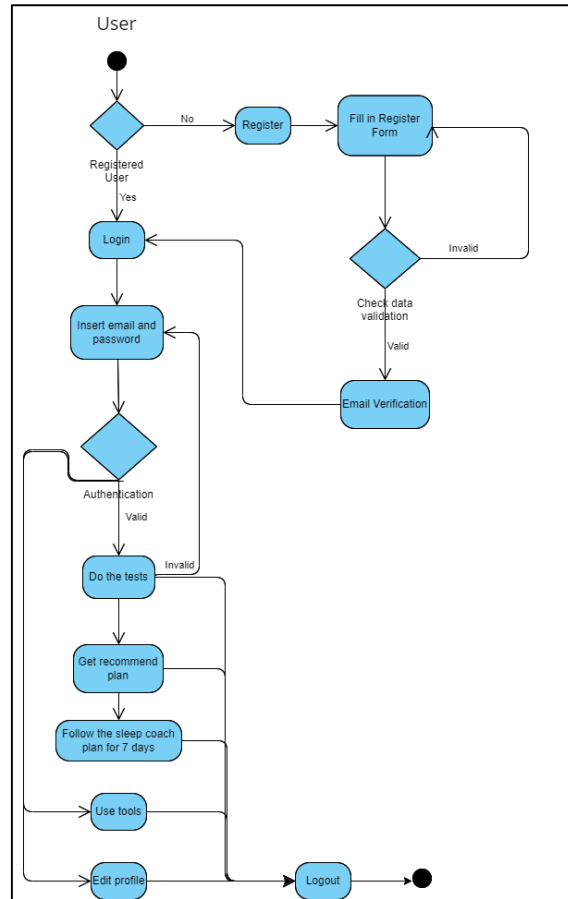


Figure 3.6: Activity diagram for user

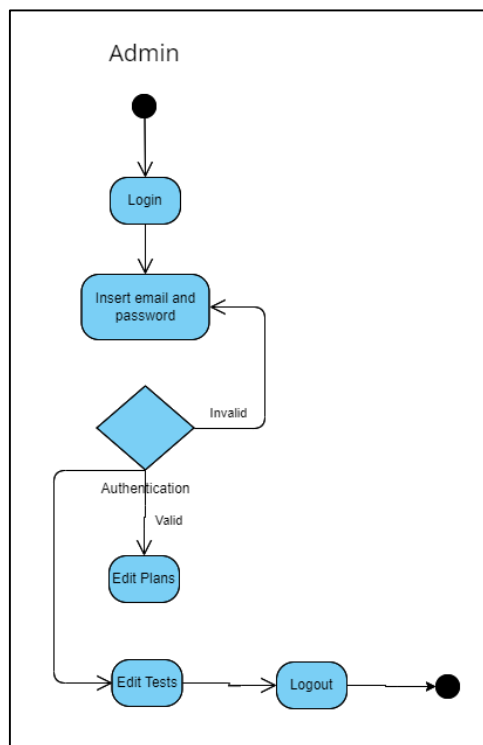
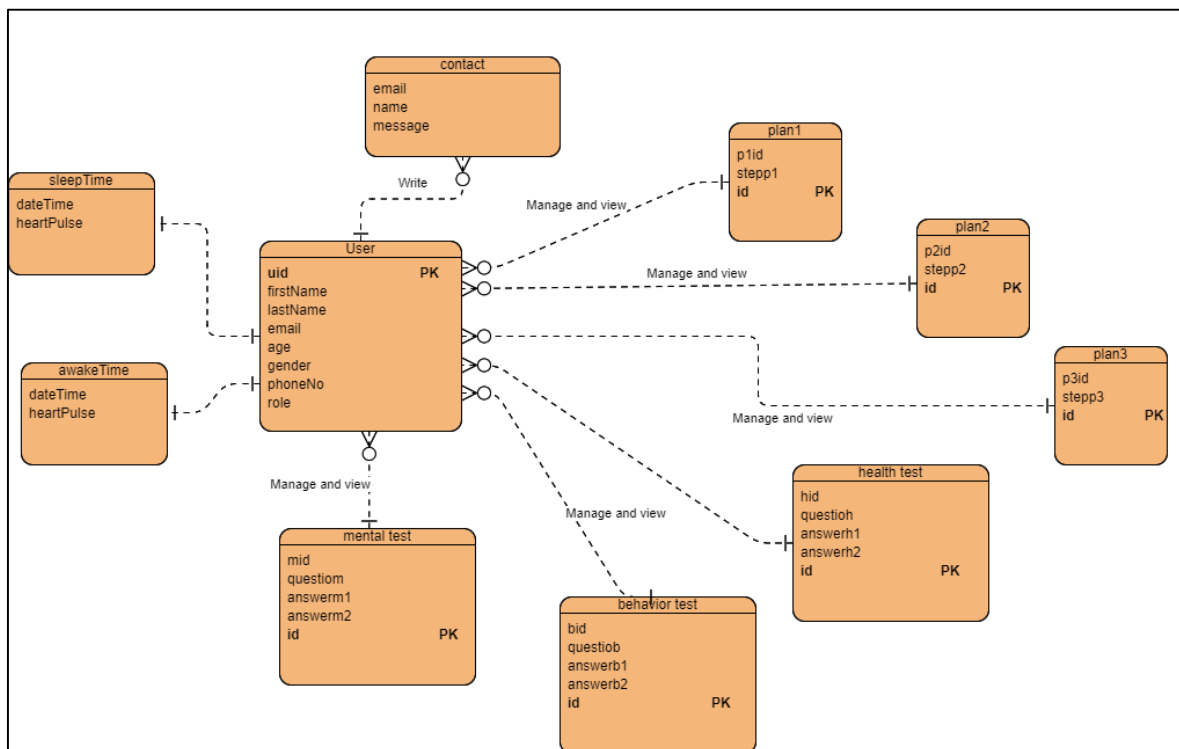


Figure 3.7: Activity diagram for admin

Figure 3.6 and 3.7 shows the activity diagram of the user and admin. The activity is starting from registration of the new user by providing their information in the register form. After registering, the users have to verify the email via email verification link. After verify the email, the user can login into the system, and complete the sleep tests to choose the sleep coach plan. User need to complete the sleep coach plan step every day. Besides that, user can use tools to help in sleeping and edit the profile. The activity is ended while the user is logout from the application. For the activity of admin is starting from login. The admin can create, read, update, and delete the steps in plans, and questions in tests. The activity is ended while the admin is logout from the application.

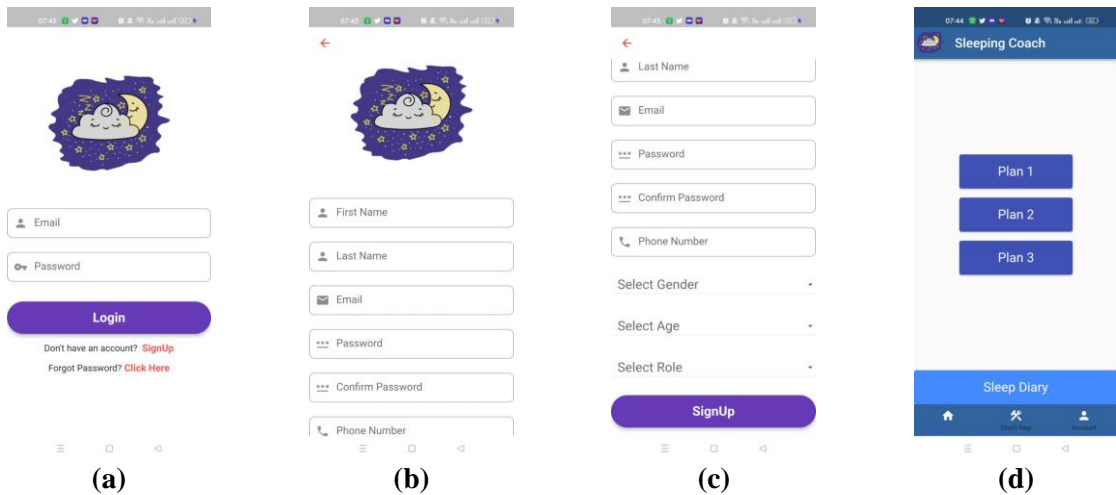


**Figure 3.8: Entity relationship diagram**

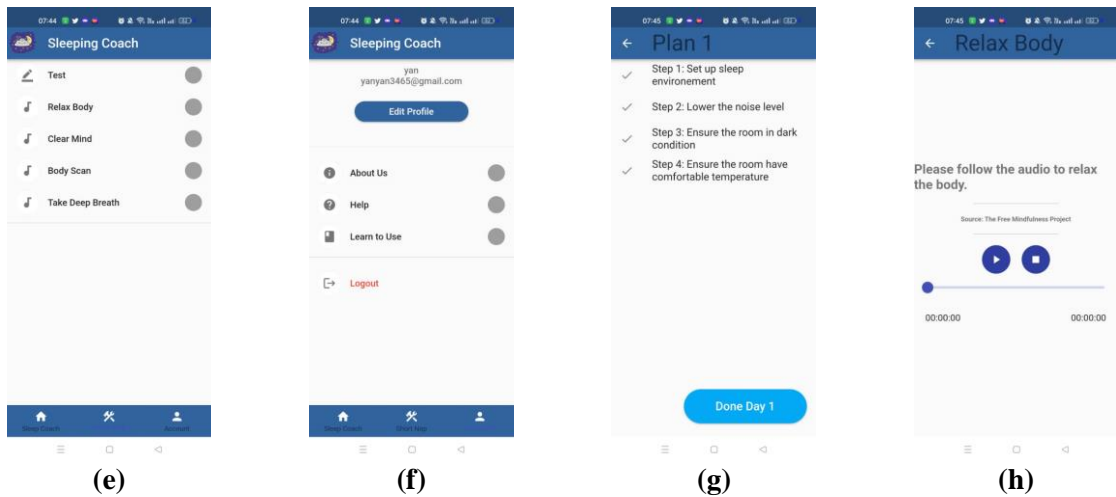
Figure 3.8 shows the entity relationship diagram which has eight tables. The tables are user, mental test, behavior test, health test, plan1, plan2, plan3, sleepTime, awakeTime and contact. User entity table contain sixteen attributes which are uid (PRIMARY KEY), firstName, lastName, email, age, gender, phoneNo, and role. While in mental test table contain four attributes which are id (PRIMARY KEY), mid, questionm, answerm1, answerm2. In health test table contain four attributes which are id (PRIMARY KEY), hid, questionh, answerh1, answerh2. In behavior test table contain four attributes which are id (PRIMARY KEY), bid, questionb, answerb1, answerb2. In plan1 table contain three attributes which are id (PRIMARY KEY), p1id, and stepp1. In plan2 table contain three attributes which are id (PRIMARY KEY), p2id, and stepp2. In plan3 table contain three attributes which are id (PRIMARY KEY), p3id, and stepp3. In sleepTime and awakeTime table contain the same attributes which are dateTime, and heartPulse. Finally, in contact table which contain three attributes which are email, name, and message.

#### 4. Implementation

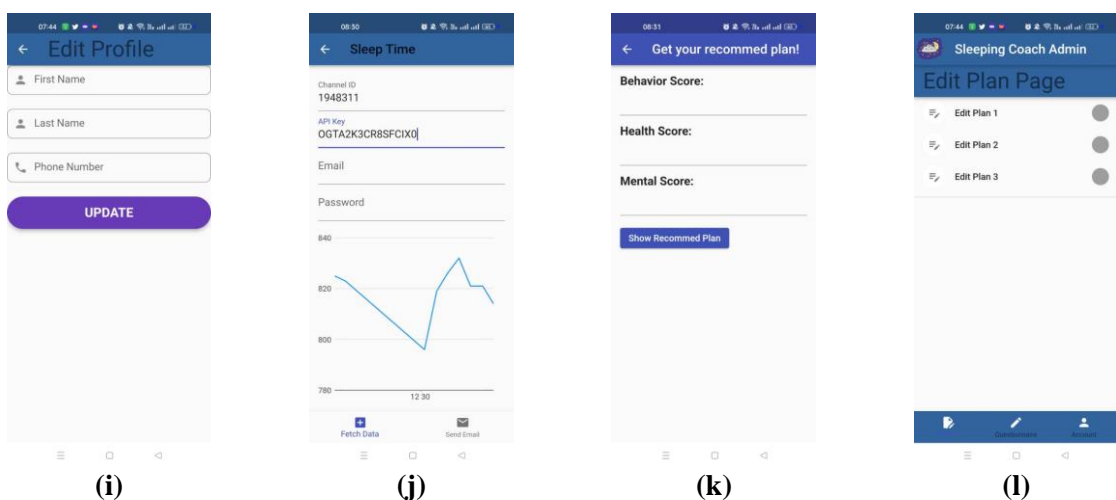
##### 4.2 Implementation of Graphical User Interface



**Figure 4.1: Login Page (a), Sign Up Page (b), Sign up Page continuous (c), Sleep Coach Page (d)**



**Figure 4.2: Tools Page (e), Account Page (f), Plan Page (g), Relax Body Tool Page (h)**



**Figure 4.3: Edit Profile Page (i), Sleep Time Page (j), Test Page (k), Edit Plan Page in Admin Module (l)**

Figure 4.1, 4.2, and 4.3 show the graphical user interface of the application. The graphical user interface that showed were Login Page, Register Page, Sleep Coach Page, Tools Page, Account Page,

Plan Page, Relax Body Tool Page, Edit Profile Page, Sleep Time Page, Test Page, and Edit Page in Admin Module.

## 4.2 Implementation of Sleep Coach module

### 4.2.1 Implementation of Sleep Coach Plan

```
final Stream<QuerySnapshot> plan1Stream = FirebaseFirestore.instance.collection('plan1').snapshots();

CollectionReference plan1 =
    FirebaseFirestore.instance.collection('plan1');
```

**Figure 4.4: Code for connect with Firebase**

Figure 4.4 shows the code to connect with Firebase to retrieved the data in plan1 collection.

```
body: StreamBuilder<QuerySnapshot>(
  stream: plan1Stream,
  builder: (BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot) {
    if (snapshot.hasError) {
      print('Something went Wrong');
    }
    if (snapshot.connectionState == ConnectionState.waiting) {
      return Center(
        child: CircularProgressIndicator(),
      ); // Center
    }
    final List storedocs = [];
    snapshot.data!.docs.map((DocumentSnapshot document) {
      Map a = document.data() as Map<String, dynamic>;
      if (a['plid'] != null && a['stepp1'] != null) {
        a['id'] = document.id;
        storedocs.add(a);
      }
    }).toList();

    storedocs.sort((a, b) => a['plid'].compareTo(b['plid']));
```

**Figure 4.5: Code for display screen while connecting with Firebase**

Figure 4.5 show the code of the screen when connect with Firebase, if the connection has problem, it will show the message 'Something went wrong'. While connect successfully, it will display the data that store in Firebase with the order by plid.

```

return ListView.builder(
  itemCount: storedocs.length,
  itemBuilder: (context, index){
    return ListTile(
      leading: const Icon(Icons.check),
      title: Text('Step ${storedocs[index]['plid']}: ${storedocs[index]['stepp1']}'),
    ); // ListTile
  },
); // ListView.builder
  }, // StreamBuilder
  floatingActionButton: _doneButton(),
); // Scaffold
}

```

**Figure 4.6: Code for display the list of steps in plan**

Figure 4.6 show the code to display the list view of the steps in plan with order by plid.

#### 4.1.2 Implementation of Sleep time Chart

```

Future<void> fetchData() async {
  final apiKey = 'OGTA2K3CR8SFCIX0'; // Replace with your ThingSpeak API key
  final channelId = '1948311'; // Replace with your ThingSpeak channel ID
  final url =
    'https://api.thingspeak.com/channels/$channelId/feeds.json?api_key=$apiKey&results=10';
  final response = await http.get(Uri.parse(url));
}

```

**Figure 4.7: Code for connect with ThingSpeak**

Figure 4.7 shows the code to connect with ThingSpeak to retrieved the data with http link.

```

if (response.statusCode == 200) {
  final data = json.decode(response.body);
  final feeds = data['feeds'];

  for (var feed in feeds) {
    final heartPulse = feed['field1'].toString();
    final dateTime = DateTime.parse(feed['created_at'].toString());

    setState(() {
      if (int.parse(heartPulse) < 50) {
        heartPulses.add(heartPulse);
        dateTimeList.add(dateTime.toString());
      }
    });
  }
}

```

**Figure 4.8: Code for retrieved data from ThingSpeak**

Figure 4.8 shows the code to retrieved the heartPulse and dateTime when the heartPulse is below than 50.

### 4.3 Implementation of Admin module

#### 4.3.1 Implementation of Create

```
CollectionReference plan1 =
    FirebaseFirestore.instance.collection('plan1');

Future<void> addPlan1() {
    return plan1
        .add({'p1id': p1id, 'stepp1': stepp1})
        .then((value) => print('Step Added'))
        .catchError((error) => print('Failed to Add step: $error'));
}
```

**Figure 4.9: Code for create new document and store in Firebase**

Figure 4.9 shows the code to connect with Firebase to create new document in the plan1 collection.

#### 4.3.2 Implementation of Read

```
return StreamBuilder<QuerySnapshot>(
    stream: plan1Stream,
    builder: (BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot) {
        if (snapshot.hasError) {
            print('Something went Wrong');
        }
        if (snapshot.connectionState == ConnectionState.waiting) {
            return Center(
                child: CircularProgressIndicator(),
            ); // Center
        }

        final List storedocs = [];
        snapshot.data!.docs.map((DocumentSnapshot document) {
            Map a = document.data() as Map<String, dynamic>;
            if (a['p1id'] != null && a['stepp1'] != null) {
                a['id'] = document.id;
                storedocs.add(a);
            }
        }).toList();

        storedocs.sort((a, b) => a['p1id'].compareTo(b['p1id']));
    });
```

**Figure 4.10: Code for read document which store in Firebase**

Figure 4.10 shows the code to connect with Firebase to read the fields in the plan1 collection.

### 4.3.3 Implementation of Update

```

CollectionReference plan1 =
    FirebaseFirestore.instance.collection('plan1');

Future<void> updatePlan1(id, p1id, stepp1) {
    return plan1
        .doc(id)
        .update({'p1id': p1id, 'stepp1': stepp1})
        .then((value) => print("Step Updated"))
        .catchError((error) => print("Failed to update step: $error"));
}

```

**Figure 4.11: Code for update document and store in Firebase**

Figure 4.11 shows the code to connect with Firebase to update the data of the field in the plan1 collection.

### 4.3.4 Implementation of Delete

```

CollectionReference plan1 =
    FirebaseFirestore.instance.collection('plan1');

Future<void> deleteUser(id) {
    // print("User Deleted $id");
    return plan1
        .doc(id)
        .delete()
        .then((value) => print('Step Deleted'))
        .catchError((error) => print('Failed to Delete step: $error'));
}

```

**Figure 4.12: Code for delete document and store in Firebase**

Figure 4.12 shows the code to connect with Firebase to delete the document in the plan1 collection.

## 5. Results and discussion

This section discusses about the testing result of the proposed application. The testing is based on the check list that designed in design phase. The user acceptance test form is sent to the target user and SME and the result is displayed and analyzed using the graph form.

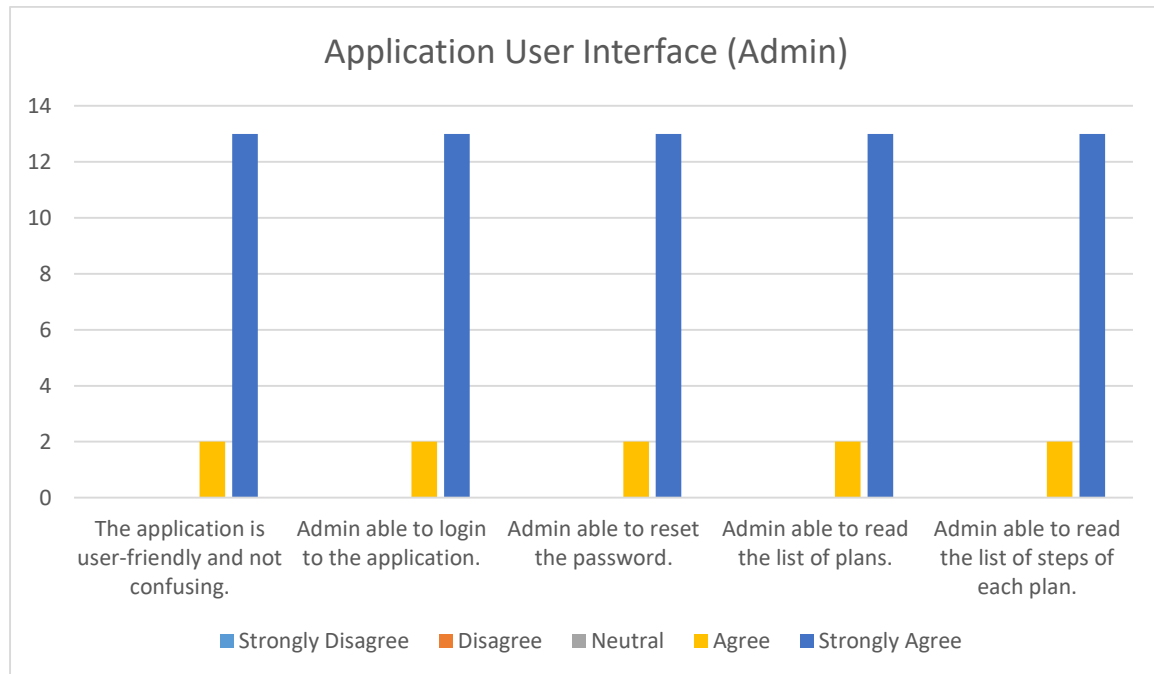
### 5.1 Security Requirement Testing Result

Security requirement test plan is to test whether the security feature of the developed application is function as the expectation. Table 5.1 show the result of the security requirement test plan.

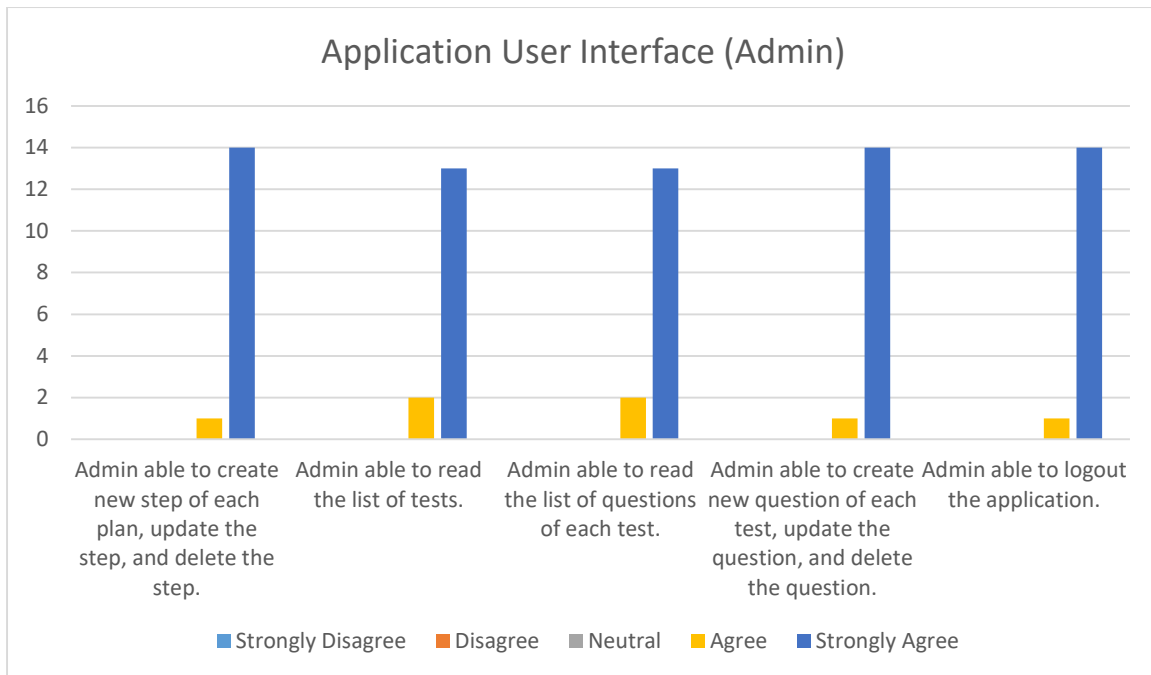
**Table 5.1: Security requirement test plan result**

No	Check List	Actual Result
1	Ensure the error message not direct indicate which part of the authentication data incorrect. For example, error message should not show “incorrect username” or “password”	Pass
2	Specify the password length inside the policy. For example, must have combination of at least 1 alphabet, and 1 numeric with minimum length 6.	Pass
3	Password should be obscured in the text box.	Pass
4	The account has to authentication with email.	Pass

6.2 User Acceptance Result

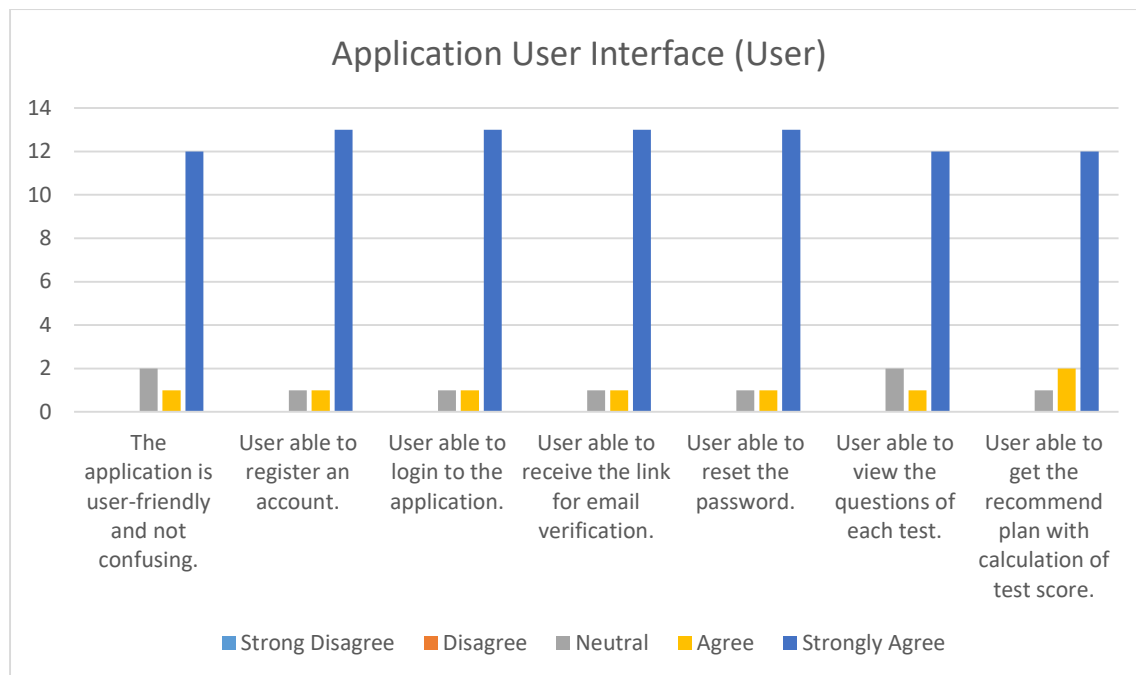


**Figure 6.1: Application User Interface Testing of the Admin**

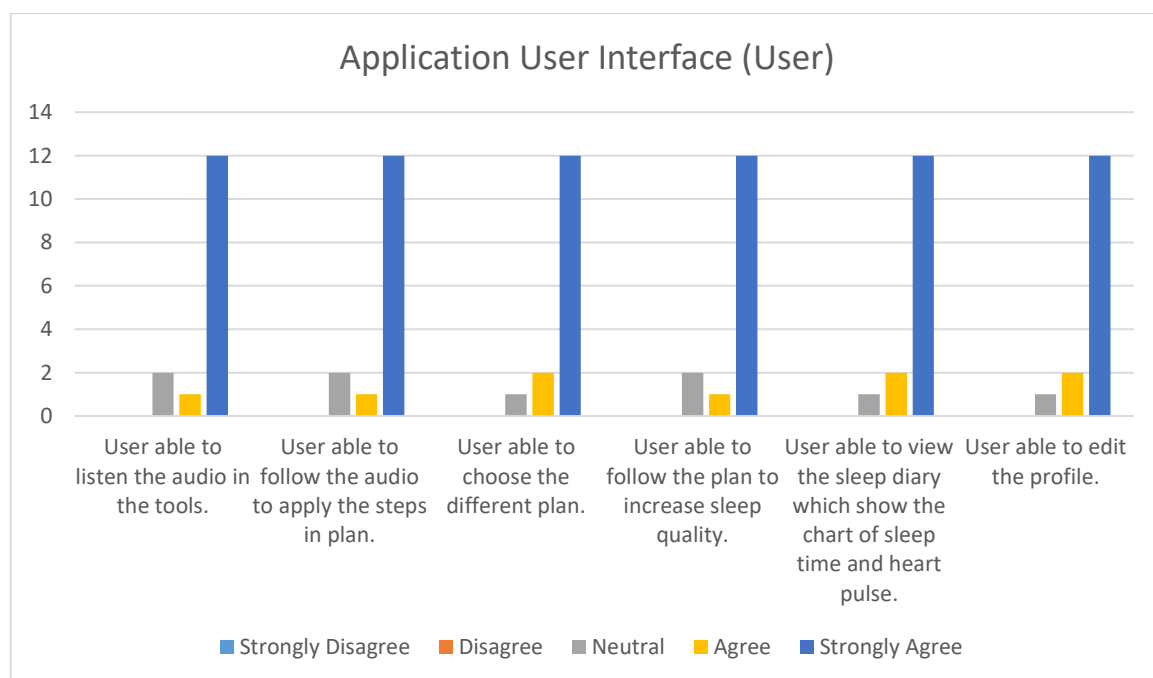


**Figure 6.2: Application User Interface Testing of the Admin**

Figure 6.1 and 6.2 shows the result of the application user interface testing of the admin. Two respondents and thirteen respondents agree that the application is user-friendly and not confusing. Two respondents agree, and thirteen respondents strongly agree that the admin able to login to the application. Two respondents agree, and thirteen respondents strongly agree the admin able to reset the password. Two respondents agree, and thirteen respondents strongly agree the admin able to read the list of plans. Two respondents agree, and thirteen respondents strongly agree the admin able to read the list of steps of each plan. One respondent and fourteen respondents agree admin able to create, new step of each plan, update the step, and delete the step. Two respondents agree, and thirteen respondents strongly agree the admin able to read the list of tests. Two respondents agree, and thirteen respondents strongly agree the admin able to read the list of questions of each test. One respondent and fourteen respondents agree admin able to create, new question of each test, update the question, and delete the question. One respondent and fourteen respondents agree admin able to logout the application.



**Figure 6.3: Application User Interface Testing of the User**



**Figure 6.4: Application User Interface Testing of the User**

Figure 6.3 and 6.4 shows the result of the application user interface testing of the user. Two respondents neutral, one respondent agree, and twelve respondents strongly agree the application is user-friendly and not confusing. One respondent neutral, one respondent agrees, and thirteen respondents strongly agree the user able to register an account. One respondent neutral, one respondent agrees, and thirteen respondents strongly agree the user able to login to the application. One respondent neutral, one respondent agrees, and thirteen respondents strongly agree the user able to receive the link for email verification. One respondent neutral, one respondent agrees, and thirteen respondents strongly agree the user able to reset the password. Two respondents neutral, one respondent agree, and twelve respondents strongly agree the user able to view the questions of each

test. One respondent neutral, two respondents agree, and thirteen respondents strongly agree the user able to get the recommend plan with calculation of test score. Two respondents neutral, one respondent agree, and twelve respondents strongly agree the user able to listen the audio in the tools. Two respondents neutral, one respondent agree, and twelve respondents strongly agree the user able to follow the audio to apply the steps in plan. One respondent neutral, two respondents agree, and thirteen respondents strongly agree the user able to choose the different plan. Two respondents neutral, one respondent agree, and twelve respondents strongly agree the user able to follow the plan to increase sleep quality. One respondent neutral, two respondents agree, and thirteen respondents strongly agree the user able to view the sleep diary which show the chart of sleep time and heart pulse. One respondent neutral, two respondents agree, and thirteen respondents strongly agree the user able to edit the profile.

## 5. Conclusion

In conclusion, the mobile based sleeping coach application allow admin to edit the plans and tests, and allow the user to follow the sleep coach plan, answer the tests to get the recommend plan, view the sleep time chart to observe the sleep quality through the chart form, edit profile, use tools to help in sleeping.

This application has four advantages. This application provides three different plans to allow the user to choose according their own need. Besides that, the administrator module allows the admin to edit the plans and tests directly to change the content of plans and tests in user module. The application also contains different tools to allow user to follow the tools to help in sleeping. The application also can connect to specific cloud platform to retrieved the data to view the sleeping time of user.

However, the application also has two disadvantages. The application cannot allow the user to tick the check list the steps of plan step by step. The application does not have notification function.

Since the application do not have function to tick the check list of plans, and notification function. For future implementation, the application should add the function to allow the user to tick the check list the steps of plan step by step, and notification function.

## Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

## References

- [1] M. CLinic, "Insomnia," Mayo Clinic, 15 October 2016. [Online]. Available: <https://www.mayoclinic.org/diseases-conditions/insomnia/symptoms-causes/syc-20355167#:~:text=Insomnia%20is%20a%20common%20sleep,tired%20when%20you%20wake%20up>. [Accessed 13 November 2022].
- [2] B. Z. J. P. R. M. M. A. J. M. G.-G. S. T. Y. G. & L. F.-L. Ignacio Perez-Pozuelo, "The future of sleep health: a data-driven revolution in sleep science and medicine," *npj Digital Medicine*, vol. 3, p. 42, 2020 March 23.
- [3] A. O. I. Erin Dawna Reilly 1, 5. A. O. I. Stephanie A Robinson 4, B. A. P. 4. A. O. Image, M. M. G. 6. A. O. Image, 5. A. O. I. Renda Soylemez Wiener 4 and C. C.-S. 7, "Mobile Intervention to Improve Sleep and Functional Health of Veterans With Insomnia: Randomized Controlled Trial," *JMIR Form Res*, vol. 5, p. 12, 2021 December 09.
- [4] Qaseem, D. Kansagara, M. Forcica, M. Cooke, T. Denberg, M. Barry, C. Boyd, R. Chow, N. Fitterman, R. Harris and e. al., "Management of chronic insomnia disorder in adults: A clinical practice guideline from the American college of physicians.," *Ann. Intern. Med.*, vol. 165, p. 125–133, 2016.
- [5] M. Sharma and C. Andrade, "Behavioral interventions for insomnia: Theory and practice.," *Indian J. Psychiatry*, vol. 54, p. 359–366, 2012.
- [6] & S. P. Khawas, "Application of firebase in android app development-a study.," *International Journal of Computer Applications*, vol. 179, no. 46, pp. 49-53, 2018 June. A. Qaseem, D. Kansagara, M. Forcica, M. Cooke, T. Denberg, M. Barry, C. Boyd, R. Chow, N. Fitterman, R. Harris and e. al., "Management of chronic insomnia disorder in adults: A clinical practice guideline from the American college of physicians.," *Ann. Intern. Med.*, vol. 165, p. 125–133, 2016.
- [7] M. Sharma and C. Andrade, "Behavioral interventions for insomnia: Theory and practice.," *Indian J. Psychiatry*, vol. 54, p. 359–366, 2012.
- [8] L. F. a. J. H. Begum Erten Uyumaz \*, "A Review of Digital Cognitive Behavioral Therapy for Insomnia (CBT-I Apps): Are They Designed for Engagement?," *International Journal of Environmental Research and Public Health*, vol. 18, no. 6, p. 2929, 2021 March 12.
- [9] *Mobile Mental Health*. [Online]. Available: <https://myvaapps.com/#/insomnia/>. [Accessed 20 November 2022].
- [10] "Sleepspace: Sleep operating system that ends sleeplessness 🌙," *SleepSpace on Wefunder*. [Online]. Available: <https://wefunder.com/sleepspace/>. [Accessed 20 November 2022].
- [11] "Va.gov: Veterans Affairs," *Mobile App: CBT-i Coach*, 08-May-2013. [Online]. Available: [https://www.ptsd.va.gov/appvid/mobile/cbticoach\\_app\\_public.asp](https://www.ptsd.va.gov/appvid/mobile/cbticoach_app_public.asp). [Accessed 20 November 2022].
- [12] L. Sharma, "WaterFall Model," Tools, 8 October 2021. [Online]. Available: <https://www.toolsqa.com/software-testing/waterfall-model/>. [Accessed 16 December 2022].
- [13] S. Lewis, "Waterfall Model," TechTarget, February 2019. [Online]. Available: <https://www.techtarget.com/searchsoftwarequality/definition/waterfall-model>. [Accessed 16 December 2022].