

MyJomChangeYourHabit: An Android Application for Quitting Bad Habit

Azrul Asaari, Salama A. Mostafa*

Faculty of Computer Science & Information Technology,
Universiti Tun Hussein Onn Malaysia, 86400 Johor, MALAYSIA

DOI: <https://doi.org/10.30880/aitcs.2021.02.02.058>

Received 09 July 2021; Accepted 29 September 2021; Available online 30 November 2021

Abstract: A bad habit is a problem that all people in the world will face in their life. A bad habit is a negative lifestyle, whether it is physical or mental: Smoking, pornography, smartphone, and social media addictions are just few examples. Nowadays some so many peoples have so many bad habits in their normal life. The bad habit also can come from the smartphone itself. This application aims to design MyJomChangeYourHabit using object-oriented analysis design, to help the user to reduces or stop themselves from their bad habit and lastly to perform alpha and beta testing on the application. The methodology that was used for this application development is Object-Oriented Methodology. This object-oriented methodology has six phases: requirement, analysis, modelling, design, implementation, and lastly, evaluation. In analysis phase there were three related work that have been studied to help in determining the requirement for this application. This application was implement using Java programming language, Android studio as the environment platform and Firebase Database as the database. The application underwent two testing that are alpha and beta testing. This project, the MyJomChangeYourHabit application, has been completed and can be used by users of different ages. This application can help other people who have bad habits to change themselves or avoid these habits. Although there is a limitation that have been figured out. For the future work there is many improvisations can be done with the application such as how to increase the effectiveness in help user to reduce or stop smoking, add app blocker to reduce the usage of the application or smartphone itself. These improvements towards the system will eventually make the system more effective, and more user-friendly.

Keywords: Habit, Android Object-Oriented Methodology

1. Introduction

A bad habit is a negative behavior pattern that happens to all people in the world. Each person in this world must have a bad habit. A bad habit can really affect their daily lives, for example, when someone has a bad habit of using social media. They will use so much time for scrolling something that not beneficial for them. If they can change their bad habit into something that is better than their bad habit such as focusing do their work or exercising and spend time more with their family. It is a better way of life for them to live. This MyJomChangeYourHabit can help them to minimize their bad habit or stop their bad habit. The existing application was run on an Android based application. This is

because in this developing world, based on Statista that was published by Joschka [1], almost 96.86% people in Malaysia have their own smartphones. So, everybody in Malaysia can use this application easily. Many bad habits were natural come from the usages of smartphone itself. So, the objective for this application is:

- To design MyJomChangeYourHabit using object-oriented analysis design
- To develop and implement MyJomChangeYourHabit that works to help people reduce their bad habits.
- To perform alpha and beta testing on the MyJomChangeYourHabit application.

This objective can solve the problem where the existing application for stopping a bad habit is many in the market right now. The existing application lack of function that we need to stop or reduce the bad habit. In the current application lack of time restriction for using an application on the smartphone. A function allows users to set a reminder to help them stop using the application when the time has run out. Next, there is a problem in the current application where it does not have any features or functions that can help users remind them what to do throughout the day. In this, MyJomChangeYourHabit will give a daily schedule through notification. The user needs to follow in their free time, such as at night after they have finished work the app will give notification to remind to do some tasks such as pray for Muslim and exercise for non-Muslim. Lastly, the current application does not have a function that can help users ease their minds from negative thinking. This application will help the user ease their mind by using music such as zikr to help them ease their mind and be more focused on stopping from their bad habit.

2. Related Work

This part has three comparisons of the similar existing system with some modules that almost the same and function but with different approaches and techniques. This study is for an improvement toward the system that wants to be developed. In the 1 table, there has a summary table for the comparison between the developed system and the similar system.

2.5.1 Forksy

Based on Forksy term of service, [2] Forksy is a chatbot that uses communication platforms such as Facebook Messenger, Telegram, Viber and etc. This system provides a service that the user can communicate with the chatbot by texting a message or sending a food or drink photo that the user has consumed throughout the day. The system can create a diary of the food that you eat throughout the day and a diary you can share with your friend or your dietic trainer from the text and the picture. This Forksy chatbot also can communicate friendly with the user. The chatbot can respond the user's text based on what the user wants to tell the chatbot. This chatbot also can help the user to stop the user from their bad eating habit. It helps the user by only assist the user effort in changing their bad habit.

2.5.2 NutriWalking (NW)

Based on Hartzler et al. [3] NutriWalking (NW) is a mobile application that was built for United States of America citizens. This application was created because many individuals have a chronic health conditions such as type 2 Diabetes Mellitus that have not following the food and physical activity lifestyle guidelines. This application was a team-based application where the apps support one to ten peer team members. There is a functionality in this program where there is a digital "coach," and the software will also allow users to communicate openly through team posts and via the dashboard of the app to provide private peer-to-peer messaging.

2.5.3 Personal health Assistance via Smartwatch Based Motion

Based on R. Lutze and K. Waldhör [4], Personal health assistance is an application where the application will monitor, check user health, and give suggestions or ways to help the user to overcome their sick. A personal health system is like a private doctor to the user, but this personal health system is easy to use rather than you have a private doctor. Because the private doctor cannot be with you all day long, but with this system, it can be with you anywhere and anytime. In this application, it applies personal health assistant in the smartwatch. Smartwatch is a watch, but in the watch, they have so many sensors to monitor well-being and health parameters.

2.3 Comparison of Related System

Table 1: Comparison of Existing System

| | Forksy | NutriWalking | Personal health Assistance | MyJomChange YourHabit |
|----------------------|-------------------------------------|-----------------------|----------------------------|----------------------------------|
| Login | No | Yes | Yes | Yes |
| Usability platform | All website, androids, iOS, windows | iOS and android | Smartwatch | android |
| Focus Habit | Eating | Eating and lifestyles | Lifestyle | Smoking, eating, lifestyle, etc. |
| Daily recommendation | Yes | Yes | No | No |
| Fitness tracker | No | Yes | Yes | No |
| Audio module | No | No | No | Yes |
| Add Habit | No | No | No | Yes |
| Daily Reminder | Yes | Yes | No | Yes |
| Set Reminder | No | No | No | Yes |

From the table 1, we can conclude that all the system has their own unique trait and have their own focusing group of users who they want them to use the system or an application. For examples, the Forksy, NutriWalking, and the smartwatch-based system were focusing on eating and lifestyle. But for the MyJomChangeYourHabit was focusing on many groups of users. An example, the application was for eating, lifestyle, smoking, etc., so the application can be generally used for the user who wants to stop a bad habit. The application is not focusing only on one bad habit for the user to change, but the apps can have many bad habits that the user can choose or customize their apps to follow their suitable needs.

3. Methodology/Framework

The methodology that we use for developing JomMyChangeYourHabit is an object-oriented methodology. The object-oriented methodology or approach is a modern approach to device architecture that encourages and enables the reuse of software components [5]. This Object-Oriented Methodology (OOM) is using Unified Modelling Language (UML) as one of the modeling tools. The Unified Modelling Language is a part of the international standard Object Management Group (OMG). By using this Object-Oriented method or approach, it is easy to develop the system by using a component basis that enables the effectiveness of reusing the existing components as in Figure 1.

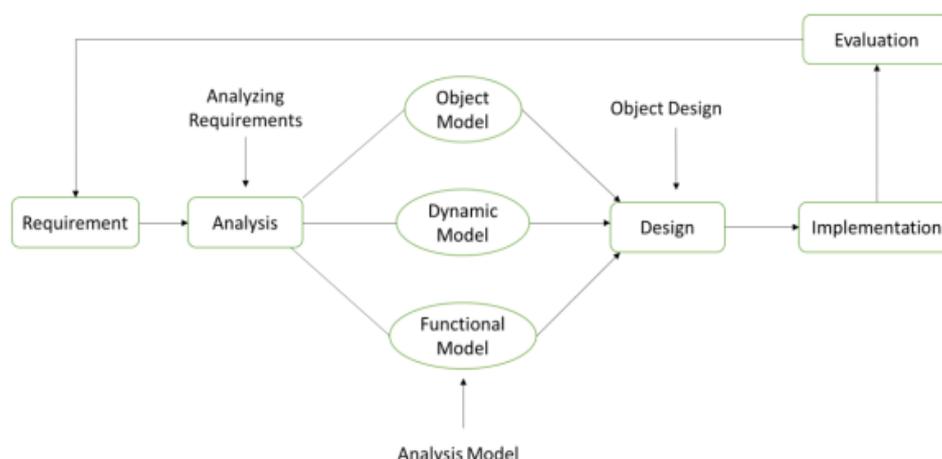


Figure 1 Object-Oriented Modelling Technique Methodology [5]

For the software development methodology, there are five (5) general phases that were used to develop the software. The phases are analysis, design, implementation, and lastly, testing or evaluation. In the Object-Oriented methodology, the phases were summarized in Table 2.

Table 2 : Summary of every phase in OOM

| Phases | Activities | Result |
|-------------|--|---|
| Requirement | <ol style="list-style-type: none"> 1. Identify the background of the project, problem statement, objectives of the project, scope, and the methodology that are used in this development of MyJomChangeYourHabit. 2. Collecting the data by using an observation method. 3. Make a brainstorming session to take out the requirement for this project. 4. Search previous documents or an existing system and try to find the requirement based on the research. 5. Preparing the schedule for the project. | <ol style="list-style-type: none"> 1. Proposal 2. Gantt chart |
| Analysis | <ol style="list-style-type: none"> 1. Determining the functional and non-functional requirement. 2. Determine the hardware and the software that is used in this system development. | <ol style="list-style-type: none"> 1. Functional and Non-Functional requirement 2. Document of the system requirement |
| Modelling | <ol style="list-style-type: none"> 1. Object Model (It displays the groups of objects and their relationships as a class diagram, reflecting the system's static structure.). 2. Dynamic Model (It displays the changes that occur with the events that might occur in the system in the states of different artifacts.). 3. Functional Model (It explains the data flow and changes that occur in the system to the data). | |

Table 2: continue

| Phases | Activities | Result |
|------------|---|--|
| Design | <ol style="list-style-type: none"> 1. Designing the Use Case diagram, Sequence diagram, Activity diagram and lastly Class diagram. 2. Designing the user interfaces for the system. 3. Designing the database system | <ol style="list-style-type: none"> 1. Use case diagram 2. Sequence diagram 3. Activity diagram 4. Class diagram 5. User interfaces design 6. Database Tables |
| Implement | <ol style="list-style-type: none"> 1. Start to implement the system coding. 2. Implement the database follow the database design | <ol style="list-style-type: none"> 1. Programming Code 2. Prototype 3. Working application |
| Evaluation | <ol style="list-style-type: none"> 1. Perform testing for the functionality and the user interfaces. 2. Perform a user acceptance test | <ol style="list-style-type: none"> 1. Feedback from the user |

3.1 Analysis

System analysis is the process where the requirement for the system or software requirement is determined from studying the user needs [6]. The requirement and the result have already been determined in Section 2, Literature Review. The Unified Modelling Language (UML) was used to describe how the system wanted to be developed worked and the flow for each of the functions in the system. Three diagrams describe or explain the functions for this system that want to be developed. The diagram is use case diagram, activity diagram, and class diagram. Figure 1 shows the process or activity in the object-oriented analysis process. The first activity is creating the system definition. This is where the system functionality was described through the use case diagram. After that is build the class diagram, activity diagram, and lastly finalizing the analysis documentation.

The system requirement that being gathered is stated below:

1. User and Admin should be able to log in and be directed to their specific function.
2. User should be able to register.
3. User should be able to add habit they want to
4. User should be able to play, pause, skip, and previous the music from applications database.
5. User should be able to set a reminder for a certain time.
6. Users should be able to see their bad habit history.
7. Users should be able to manage their bad habit history.

3.2 Modelling & Design

3.2.1 Use Case Diagram

There is six use case related to the user to conduct in this project. The user has to login into the system using registered information [6]. If the user does not have registered yet, the user can register at the registration pages. After that, they can use the information that the user provides in the registration before. Next, the user can add their bad habit into the database, and they can retrieve the bad habit data on the history page. On the history page, they can manage the data by deleting the data. Then there is a music player that they can retrieve and play the song with function play, pause, next, and previous the song. Next, the user can set the reminder based on the specific times that have been set. And after that, the notification will appear after the reminder time has finished. This can be referred to the Figure 2 below.

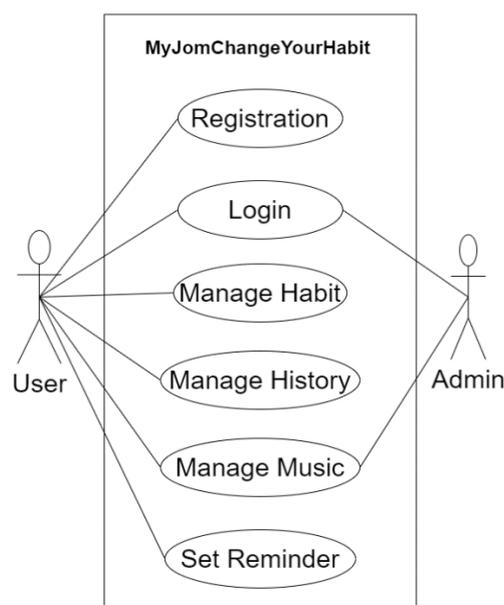


Figure 2 : The use case of the MyJomChangeYourHabit

There is two use case related to the admin. Admin can log in into the system with the required credential that is fixed account username and password. After login, they are automatically directed to upload the song page. On the page admin need to insert the title of the song and choose the music file; after that, upload the song into the database by clicking the upload button. This can be referred to in the figure above.

3.2.2 Class Diagram

The class diagram is one of the Unified Modelling Language (UML). The diagram shows class, attributes, operations, methods, and the relationship among the objects. The class diagram is a static structure of a system. The class diagram is the main part of object-oriented modeling. This diagram was used as conceptual modeling of the systematics of the application. The diagram also was a model for the programming code model. The class diagram also can be used for data modeling. Appendix B shows the class diagram for the system. Figure 3 shows the class diagram for MyJomChangeYourHabit. There is five class in this application. The class is Habit, User, Music, History and Admin.

3.3 Database Design

MyJomChangeYourHabit application will use the Firebase database as the Database Management System (DBMS). The purposes of the DBMS based [7] is to provide a data-safe storage environment that enables information extraction from existing data. Below is the table for the database for MyJomChangeYourHabit application.

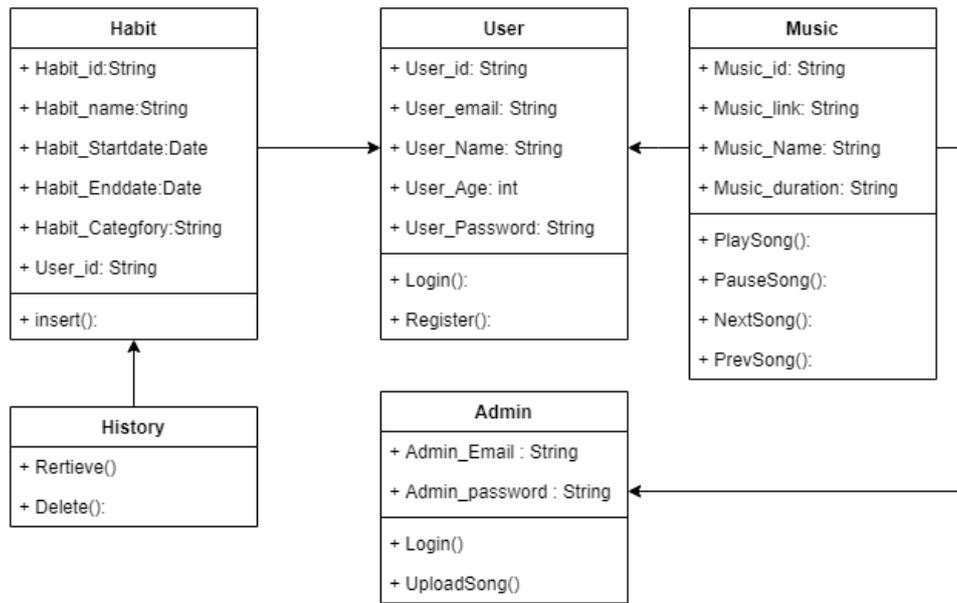


Figure 3: The class diagram of the MyJomChangeYourHabit

Table 2 User Database

| Attribute | Data Type | Size | Details |
|---------------|-----------|------|-----------------------|
| User_Name | VARCHAR | 255 | Username for the user |
| User_email | VARCHAR | 255 | Email of the user |
| User_Password | VARCHAR | 255 | User password |

Table 3 show the user database. The attribute in the database is the User_id, User_Name, User_email, and lastly User_Password.

Table 3 : Habit Database

| Attribute | Data Type | Size | Details |
|-----------------|-----------|------|-------------------|
| Habit_name | VARCHAR | 255 | Name of the habit |
| Habit_Category | VARCHAR | 255 | Habit category |
| Habit_startdate | DATE | 3 | Habit start date |
| Habit_enddate | DATE | 3 | Habit end date |

Table 4 show the database for the habit. The attribute in the database is the Habit_id, Habit_name, Habit_Note, Habit_Category, Habit_startdate, and lastly Habit_enddate.

Table 4 : Music Database

| (2018). | (2018). | (2018). | (2018). |
|------------|---------|---------|-------------------|
| Music_id | INT | 11 | Id for the music |
| Music_name | VARCHAR | 255 | Name of the music |
| Music link | VARCHAR | 255 | Type of the music |

Table 4.3 show the database for the music. The attribute is Music_id, Music_name and lastly Music_type.

3.4 Hardware and Software Specification

The hardware and software used are listed out in the Table 5 and 6:

Table 5 : List of Hardware.

| Hardware | Description |
|----------------------------|--|
| Computer | ASUS TUF FX-505DT |
| Processor | AMD Ryzen 7 3750H with Radeon Vega Mobile Gfx 2.30 GHz |
| Random Access Memory (RAM) | 8GB |
| Solid State Drive (SSD) | 512MB |
| Hard Disk Drive (HDD) | 1TB |
| Other Peripheral Device | Mouse and Keyboard |

Table 6 : List of Software

| Software | Description |
|----------------|--|
| Windows 10 | Operating system for ASUS TUF FX-505DT |
| Android Studio | Designing the application interfaces |
| SQLite | Designing the user database |
| Team Gantt | Design the Gantt chart |
| Microsoft Word | Documenting the work |

3.4 Design

After all the functional and non-functional requirement has been determine. The interfaces of the application have been design. Figure 4 shows example of the interfaces that have been design.

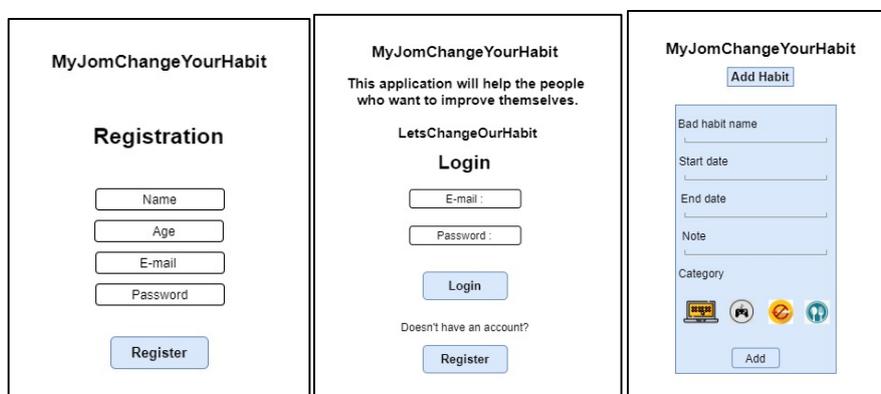


Figure 4 : Example of Interfaces Design

4. Results and Discussion

4.1 Implementation Result

This part will show an example of the modules of MyJomChangeYourHabit application which develop using Android Studio platform to develop this project. In this project the language that were used is Java programming language. The database for this application is using Firebase Realtime Database. In the Firebase Database we also have the authentication function for login and registration. The main activity in this implementation phase is the coding of the programming. Figure 5 shows an example of the implemented interfaces for the application.

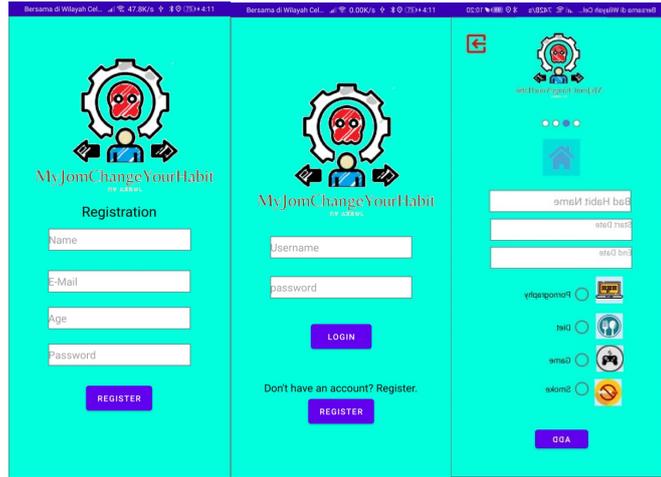


Figure 5: Example of Implemented Interfaces

4.2 Testing

Testing is crucial since it ensures that an application is ready for release in the post-implementation testing of the application's functionality. Furthermore, it is carried out to ensure that the program operates and functions properly in accordance with the interface design. The proposed application is designed with an interface that allows users to interact with it directly. The test strategies for the suggested applications are described in the sections below. The input specification, which is discussed in the next section, is used to test the test cases.

4.2.1 Alpha Testing

Table 7 shows the result of the testing for all functions in the application are presented in this section. There are seven (7) functions that have been tested for this application. The function is registration, login, add habit, manage history, music player, music upload, and lastly, set the reminder.

Table 7 : Shows The Result of The Alpha Testing.

| Module | Test cases | Expected Output | Actual output |
|---------------------|---|---|---------------|
| Registration Module | Page display all the text fields and registration button. | Display all the textbox and registration buttons. | Success |
| | Show error when a user does not enter name field | Error message is displayed | Success |
| | Show error when a user does not enter age field | Error message is displayed | Success |
| | Show error when a user does not enter email field | Error message is displayed | Success |
| | Show error when a user does not enter password field | Error message is displayed | Success |
| | User registered when clicking the button when all the information complete. | User registered | Success |
| | User registrations fail when user enter the wrong email format | Error message is displayed | Success |
| Login Module | Application display login button, username, and password textfield. | Displays username and password text field | Success |
| | Enter correct username and password | Success login | Success |
| | Enter correct username and wrong password | Unsuccessful login | Success |

Table 7 : (cont.)

| Module | Test cases | Expected Output | Actual output |
|---------------------|---|--|---------------|
| | Enter wrong username and correct password | Unsuccessful login | Success |
| | Login without entering username and password | Error message is displayed | Success |
| | Registration button directs the user to the registration page | directed to registration page | Success |
| | Insert admin credential | directed to admin page | Success |
| Add Habit Module | Display all the text fields and all the function in the page. | Displays add habit page textfield | Success |
| | Let the habit name text field empty | Error message is displayed | Success |
| | Let the Start Date Empty | Error message is displayed | Success |
| | Let the End date text field empty | Error message is displayed | Success |
| | Insert add habit into the database | Database successfully saved | Success |
| | logout button | Directed to the login page | Success |
| History Module | Display the list of user bad habit | Displays all users bad habits. | Success |
| | User delete the bad habit data from the list and the data. | Data successfully delete | Success |
| Music Player Module | Display all the song in the database into the application | Display a list of music | Success |
| | Click the song | Song played | Success |
| | Song title is shown when the song played | Title follows the song title. | Success |
| | Click pause button | Song paused | Success |
| | Click play button | Song played | Success |
| | Click the previous button | Song change to previous song | Success |
| Upload Music Module | Let the song title text field empty | Error message is displayed | Success |
| | Click the upload song button | go direct to admin phone music file | Success |
| | Let song file empty | Error message is displayed | Success |
| | Click upload button | Song upload and display message | Success |
| | Click the upload button while uploading song in progress | Error message is displayed | Success |
| | logout button | Directed to the login page | Success |
| Set Reminder Module | Display all the button | all button displayed | Success |
| | Click the 15 minutes button | All button will disable and show a success message | Success |

Table 7: (cont.)

| Module | Test cases | Expected Output | Actual output |
|--------|-----------------------------|--|---------------|
| | Click the 30 minutes button | All button will disable and show a success message | Success |
| | Click the 1 hour button | All button will disable and show the success message | Success |

4.2.2 Beta Testing

Testing can carry out a total of seven functions of functional requirement and four functions which are the non-functional requirement. The functional requirement is that the user should register, log in, add bad habit, watch history, manage history, music player, and lastly, the set reminder function. The non-functional requirement is the design, performance, security, and reliability of the application. Figure 6 shows the user acceptance testing result for functional requirements, and Figure 7 shows the user acceptance testing result for non-functional requirements.

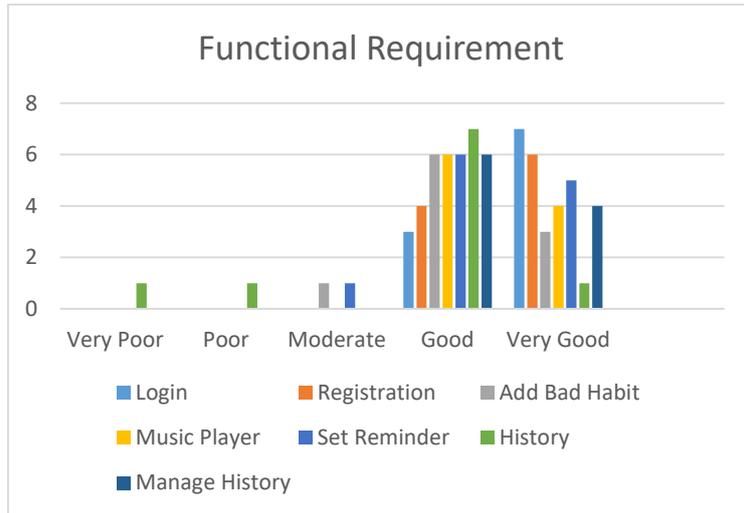


Figure 6: The User Acceptance Testing Result For Functional Requirement

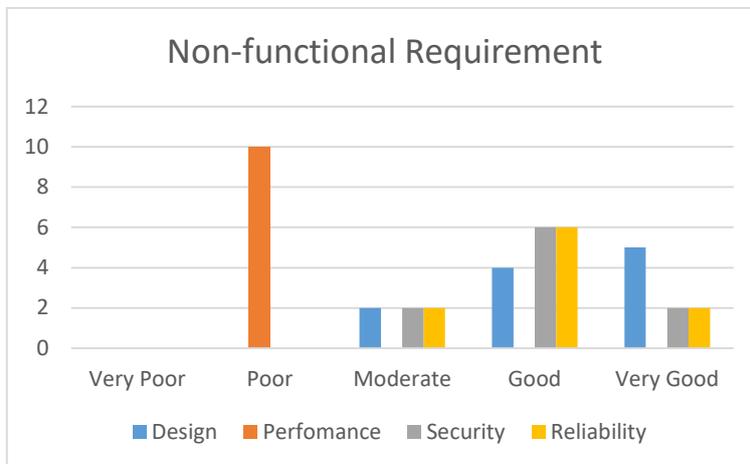


Figure 7: The User Acceptance Testing Result For Non-Functional Requirements

4.3 Discussions

According to the testing results, the functionality and user interface design are both functional and error-free. Besides, all the modules passed the testing segment because all the expected outputs was met for all the tests. The user acceptance testing or beta testing result for functional and non-functional requirements are satisfied because all the response majority are moderate to very good.

5. Conclusion

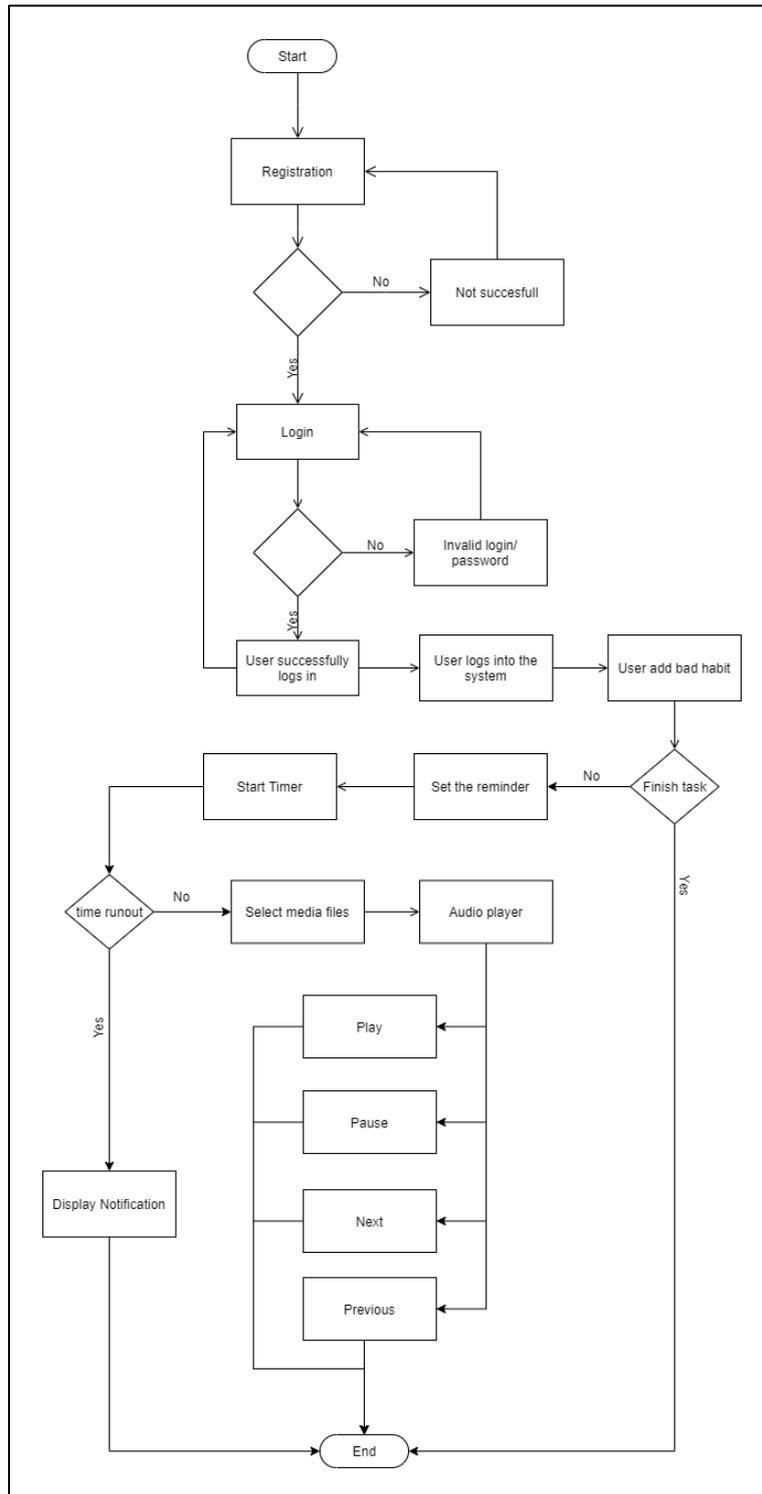
In conclusion, this application that was built by using Android Studio, Firebase database, and Java programming language has been successfully implemented. It took almost six months to complete implementing the application. The music player, add habit function, ser reminders, history, Sins sweeper game has successfully developed. All the functional requirements and non-functional requirements, objectives, scope, and goals have been fully achieved. This application is testing to ensure that it fulfills the requirement of the project and check all the functions are working as expected output. Although, there are a few limitations in this application such as there is still a limitation in helping the user to prevent, control and cure the bad habit for the smoking and eating. For future this application can be a more interactive and interactive.

Acknowledgement

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

Appendix A

Flow Chart for MyJomChangeYourHabit



References

- [1] Statista. (2021, April 7). Smartphone penetration as share of population in Malaysia 2015–2025. <https://www.statista.com/statistics/625418/smartphone-user-penetration-in-malaysia/>
- [2] Forksy. (2018, April 16). Nutrition advice automation platform. Retrieved on (2020, November 18th) at <https://getforksy.com/>
- [3] Hartzler et. al. (2016). Acceptability of a team-based mobile health (mHealth) application for lifestyle self-management in individuals with chronic illnesses. Conference proceedings: ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference. 2016. 3277-3281. 10.1109/EMBC.2016.7591428.
- [4] R. Lutze and K. Waldhör, "Personal Health Assistance for Elderly People via Smartwatch Based Motion Analysis," 2017 IEEE International Conference on Healthcare Informatics (ICHI), Park City, UT, 2017, pp. 124-133, doi: 10.1109/ICHI.2017.79.
- [5] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., & Lorenson, W. (1991). Object oriented modelling and design
- [6] Dwyer, B. (2016). Systems Analysis and Synthesis: Bridging Computer Science and Information Technology. Morgan Kaufmann Publishers. <https://doi.org/10.1016/C2015-0-04158-2>
- [7] Mostafa, S. A., AbuSalim, S. W., & Saringat, M. Z. (2020). A Comparative Study of Data Management Systems. Journal of Soft Computing and Data Mining, 1(1), 10-16.