Applied Information Technology And Computer Science Vol. 5 No. 1 (2024) 182-199

AITCS

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/aitcs
e-ISSN :2773-5141

MyAC Service App: Multifactor Authentication using Android Based for Air Conditioner Service Application

Fatin Nadia Mohd Yusof[1], Zubaile Abdullah[1]*

[1]Fakulti Sains Komputer dan Teknologi Maklumat,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

DOI: https://doi.org/10.30880/aitcs.2024.05.01.011
Received 23 June 2023; Accepted 26 May 2024; Available online 30 August 2024

Abstract: MyAC Service is an Android-based air conditioner service application that utilizes Multifactor Authentication (MFA) to secure user accounts. MyAC Service is employs complex passwords and One-time Password (OTP) in the login session, including the email address verification. This paper presents a proposed application aimed at addressing the limitations of existing MyAC Service application in functionality as a booking application. The objective of this study is to develop an MyAC Service application efficiently to do a reservation for air conditioner service and secure user's accounts from data breach. The application is designed using Java programming language, android technology, and prototype model methodology approach. The expected outcome of this project is the development of a capable Air conditioner service application that can functions to make air conditioner service reservation activity and to verifies a user's identity using MFA security method before access to the application.

Keywords: MyAC Service app, Multifactor Authentication, Booking Application

1. Introduction

Air Conditioner Service Application (MyAC Service) is a platform for air conditioner service providers at Parit Raja to make a booking online platform for their clients. Many air conditioner service management have been performed manually, such as get a booking a service by calling from client that want their service. There are also air conditioner businesses that are already using the online system to promote businesses.

However, the existing system is less concerned about security as it might have unauthorized access and fraudulent activities which lead to information theft and the system allows users to use a weak. The objectives of the project are to design a secured an android- based air conditioner service booking application that utilizes multifactor authentication (MFA) using an object-oriented approach and test the functionality of multifactor authentication (MFA) security of MyAC Service application for secure user accounts. The application was made for air conditioner service providers to make an online booking application to ease their client to get their service. The application uses multifactor authentication which

*Corresponding author: zubaile@uthm.edu.my

can verifies a user's identity using several factors [1]. This application employs complex passwords and One-time Password (OTP), including email address verification. MyAC Service application is designed with Java programming languages and uses prototype model methodology approaches. Besides that, the goal of developing this application is to provide a secure air conditioner service booking application that can help to prevent unauthorized access and protect against password-based attacks, data breaches and other forms of cybercrime. The MyAC Service application consist of 5 main modules: registration, login, service list, service shop and notification. There are two types of users for this application which is client and service provider.

The remainder of this paper is organized as follows; Section 2 discusses related work, Section 3 describes the methodology, Section 4 explains the design and analysis and finally, Section 5 presents the conclusion of the project.

2. Related Work

This section discusses the related work and comparison the existing system with the purpose system.

2.1 Single-factor Authentication (SFA)

Single-factor authentication (SFA) is a security solution that uses only one element of evidence or factor to verify a user's identity [2]. This element is usually something the user is familiar with, such as a password or a personal identification number (PIN).

In SFA, the user enters their selected credential (password or PIN) to get access to the system or service. The system then compares the credential given with the one stored in its database. Access is given if the credentials match; else, access is refused. Next, SFA is the most fundamental kind of authentication and has been routinely utilised for many years. It is straightforward to implement and comprehend for users. It does, however, have certain security drawbacks. It is subject to attacks such as password guessing, phishing, and keylogging since it depends only on what the user knows.

2.2 Multi-factor authentication (MFA)

Multifactor authentication (MFA) is a security system that requires more than one method of authentication from independent categories of credentials to verify the user's identity for a login or other transaction [2]. MFA is designed to provide an additional layer of security beyond traditional single-factor authentication systems, which only require a single method of authentication, such as a password [1]. The idea of multi-factor authentication (MFA), which uses more than two kinds of credentials, eventually came out in order to give a higher level of security and enable ongoing protection of technological equipment as well as additional vital services from unauthorized access [2].

2.2.1 Complex Password

A complex password is a combination of characters that is purposefully hard to guess or crack using brute force methods. Its main function is to protect access to a system or account, offering a higher level of security than straightforward and basic passwords. There are several crucial factors to consider in order to make a password complicated. First, a key factor in a password's security is its length. It is advised to have a minimum length of 8 characters because longer passwords are often more secure. Second, adding different character kinds increases complexity [3]. It is recommended to use a combination of capital and lowercase letters, numerals, and special characters like hashtags, @ symbols, and exclamation marks [3].

The use of common terms, expressions, or patterns that may be deduced or found in a dictionary should be avoided to maintain surprise. Next, it is important to preserve password uniqueness, which means users shouldn't use the same password across different systems or accounts. Users can establish

complicated passwords that considerably lower the possibility of unauthorized access by attackers by following these recommendations. In addition, suggestion enforcing an additional complex password policy [3].

2.2.2 One-Time Password (OTP)

On a computer system or other digital device, a one-time password (OTP) is a password that is only valid for one login session or transaction[4]. It is often used in two-factor authentication (2FA) to provide an extra layer of security. An OTP can be sent through various mediums like SMS, email, or generated through an authenticator app. When a user attempts to log in, they are prompted to enter the OTP along with their regular login credentials. The OTP is verified by the system, and if it matches, the user is granted access. OTPs are considered more secure than static passwords because they can only be used once, reducing the risk of someone gaining unauthorized access to an account[5]. They are also time-sensitive and expire after a certain period, usually within a few minutes.

2.2.3 Comparison of MFA and SFA

Table 1 presents the comparison between Multifactor Authentication (MFA) and Single-factor Authentication. Based on the advantage offered for each category, MyAC Service application have been develop using multifactor authentication security method. The decision is driven by a few benefits provided by MFA. MFA uses a technique of authentication that need multiple factors to authenticate user identification and more security than SFA [6].

MFA uses features such as passwords, physical tokens, or biometrics to provide enhanced security against attacks such as password guessing and phishing [2]. While MFA can be more complicated and might require the use of extra devices, it is increasingly being used by organizations to protect user accounts. SFA, on the other hand, depends entirely on a single factor, generally a password or PIN, making it less secure but easier. SFA has drawbacks that have led to the adoption of MFA in many circumstances, despite its widespread use and ease. In addition, by adding additional elements to single factor sign-on, multi-factor authentication enhances the security of online data [6].

Table 1: Comparison MFA and SFA

Comparison Category	MFA	SFA
Security	Provides an extra layer of security by combining multiple factors.	Relies on a single factor.
Complexity	Involves an additional step or factor, which may require more effort from users during the authentication process.	Simple and easy to implement, requiring only one factor for authentication.
Usability	Require additional devices or tools (e.g., smartphone, physical token)	Easy for users to understand and use, as it usually involves entering a password or PIN.
Prevent attacks	Offers greater protection against various attacks, such as password guessing, phishing, and keylogging.	More vulnerable to attacks as it relies solely on something the user knows.

2.3 Email Verification

Email verification is a security approach that is often used in systems to validate a user's email address [7]. During the registration or account setup procedure, a verification link or code is sent to the user's registered email address. The user must afterwards click on the link or input the code given to confirm that they have access to the specified email account.

Furthermore, email verification is a frequently used security mechanism that helps authenticate user email addresses, prevents false or spam accounts, and improves the application's overall security

[7]. By using this strategy, apps may increase user confidence, prevent unauthorized use, and continue to maintain accurate contact information for their users.

2.4 Existing Booking Application

This section provides an overview of several existing booking application such as GO DAIKIN, Kaodim and Smart Parking. The features, functions and operational mechanism of these application have been analyzed.

2.3.1 GO DAIKIN

The GO Daikin app will make it simple for users to book one of two air conditioner cleaning service packages from Go Clean[8]. Cleaning of your air conditioner's "indoor" is included in the Go Clean basic plan. The air filter, body and casing, motor, blower, evaporator coils with chemical solutions, drainage pan, and pump are all included in this.

The Go Clean Pro plan, on the other hand, includes the "outdoor" of your air conditioning unit as well as the "indoor" like the basic plan but also adds a chemical cleaning service for the condenser coils, body and casing, checking running current, pressure, and voltage, wiring connection, compressor capacitor, and refrigerant level.Go Clean makes sure that you are informed after scheduling its service by providing you with the specifics of the maintenance specialist upon their arrival in addition to a protected payment gateway. Figure 1 shows the home page GO DAIKIN application.

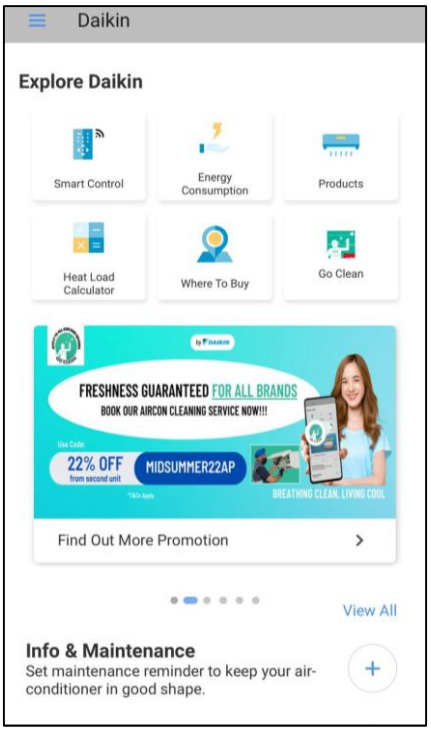

Figure 1: GO DAIKIN

2.3.2 Kaodim

Kaodim essentially offers all services like air conditioner service. The user only needs to get access to their website or mobile application to select the type of services they require based on the value proposition. Users will be directed to various pages by each category. Kaodim platform oversees hiring workers for each type of service since they are directly involved in providing the service to the users[9]. The client's job is to reserve the service by selecting a date and a backup date. For each extra demand a customer adds, the services pricing will be presented. Figure 2 shows the home page Kaodim application.

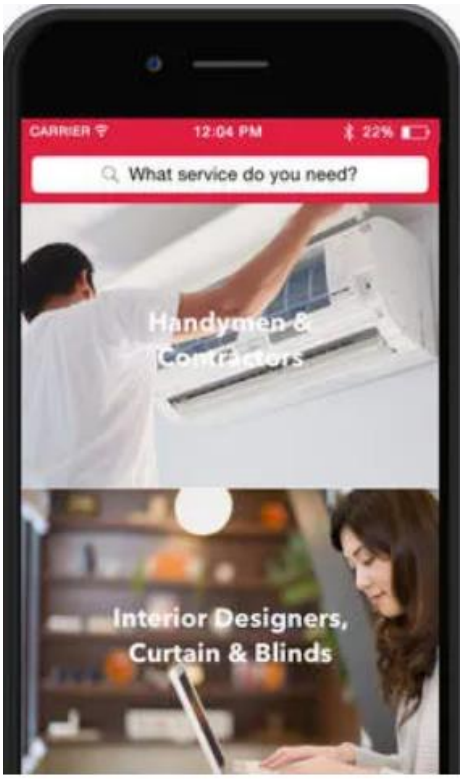

Figure 2: Kaodim

2.3.3 Smart Parking

Smart Parking is a mobile Android application for streamlining the parking process[10]. This application's major job is to ascertain the availability of parking spaces in indoor parking lots and to reserve parking spaces accordingly. Next, the application can function as a booking platform for users to book the parking slot. Users need to register before using this application which needs to input email address, password, and plat number. Figure 3 shows the home page Smart Parking application.

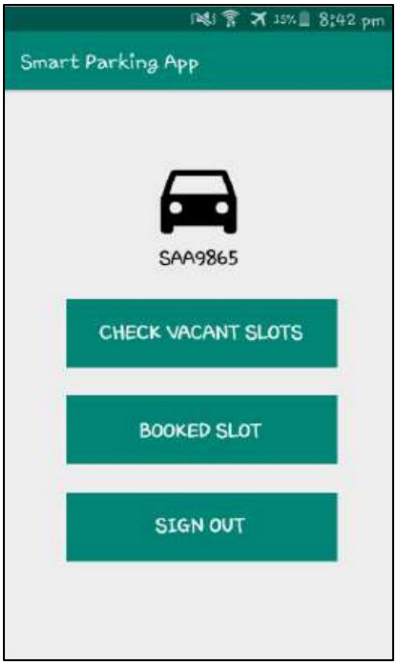

Figure 3: Smart Parking [10]

2.4 Comparison Proposed System with the Existing System

Table 2 shows the comparison of existing systems that are GO DAIKIN, Kaodim, Smart Parking and MyAC Service. All applications provide a registration and login page. The similarity between GO Daikin and MyAC Service is provided a OTP verification in login session. However, GO DAIKIN gets the OTP number via email while MyAC Service get the OTP number via SMS. Besides that, only MyAC Service application required users to input a complex password in registration session for user using it in login phase. Thus, MyAC Service is designed to be more secure compared to other existing booking applications because the application implements the MFA secure method.

Table 2: Comparison Existing application with booking

	Existing System			
Features	GO DAIKIN	Kaodim	Smart Parking	MyAC Service (Proposed system)
Registration	√	√	√	√
Login session	√	√	√	√
Using complex password	X	X	X	√
OTP verification	√	X	X	√
Booking	√	√	√	√

3. Methodology

Figure 4 present prototype model which a software development life cycle model in which a working prototype of the final product is developed and then repeatedly refined based on feedback from users. The prototype model is often chosen because it allows for a more flexible and interactive development process, where the user's needs and feedback can be considered at every stage of development. This helps to ensure that the final product is more closely aligned with the user's requirements and needs.

The prototype model typically goes through several phases: requirements gathering and analysis, prototype creation, evaluation and refinement, iteration, and final implementation. In the first phase, need to understand the user's needs and requirements for the product. In the second phase, a working prototype of the product is created based on the information gathered in the first phase. This prototype is then used as a basis for user testing and feedback. In the third phase, the prototype is evaluated by users and feedback is collected. The prototype is then refined and improved based on this feedback. This process is repeated until the final product meets the user's needs and requirements. In the final phase, the final version of the product is built. Figure 4 show the prototype model phase.

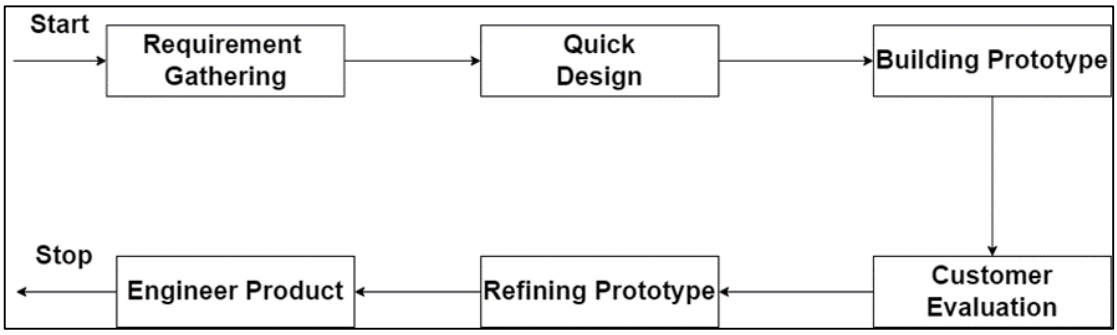

Figure 4: Prototype Model Phases [11]

3.1 System Development Workflow

There a total of five phases that are involved in this prototyping model. Table 3 shows the system development activities and the deliverables for each phase.

Table 3: System Development Phases Activities

Phase	Activities	Deliverables
Requirements gathering	Work scheduling, problem identification, scope, and objective	Gantt chart and proposal
Quick Design	Design a prototype of MyAC Service application	Design of prototype
Building Prototype	Design and develop a working prototype with MFA mechanisms	A working prototype of the air conditioner service application with MFA mechanisms
Customer Evaluation	Test and refine the prototype, focus on MFA security.	A refined and improved prototype with enhanced MFA security
Refining Prototype	Identify the problems that exist in the system and repair the existing system	System prototype
	Repetition of the planning phase until the implementation phase	
Engineer Product	Problem identification and repair the developed system	System prototype

3.2 Hardware and Software Requirement

For software requirements, we are considering developing the MyAC Service application use the Java programming language. Object-oriented programming (OOP) is a programming technique in Java that revolves around the use of objects and classes. The goal of this method is to include real-world notions, and the key principle of OOP is to mix data and the functions that act on it in such a way that no other portion of the code may access it.

The hardware requirements for an MyAC Serviceitioning application might vary depending on a number of factors. These variables include the size and complexity of the network under protection, as well as the amount and types of traffic monitored. Table 4 indicates the hardware requirements for the development of the MyAC Service application in this project.

Table 4: Hardware Requirement

Hardware requirement	Specifications
CPU	2.38 GHz AMD Ryzen 5
Memory	12 GB 2400 MHz DDR4-SODIMM
Software	Windows 11

4. System Analysis and Design

In this section, it will explain about the system architecture, functional requirements, non-functional requirement, entity relationship diagram and interface design.

4.1 System Architecture

Figure 5 shows the prototyping model of air conditioner service application. This application have two types of users which is client and service provider. From this mode, the service provider of air conditioner service that interested to register as a service provider in the application which to provide their air conditioner service for client book it while the client registration for user want to get air conditioner service. Firstly, the users need to register based on the chosen type. Then, for the registration

session, users need to enter the company details or client details to the system. Next, the users need to input an email address and password for login session.

The service provider's details, email address and password that have been encrypted are stored in the Firebase realtime database. For the other verification, the user must get a notification of one-time password (OTP) which to make sure only authenticate user can access to the application. After the user does OTP verification, the users will get the access granted to the application. Next, both users can view the application details, list of air conditioner and personal information. However, the system will only allow service providers to access the service shop page and edit, delete and update in that page while client cannot access it. Client roles may book the service and the system will give a notification of client booking to the service provider.

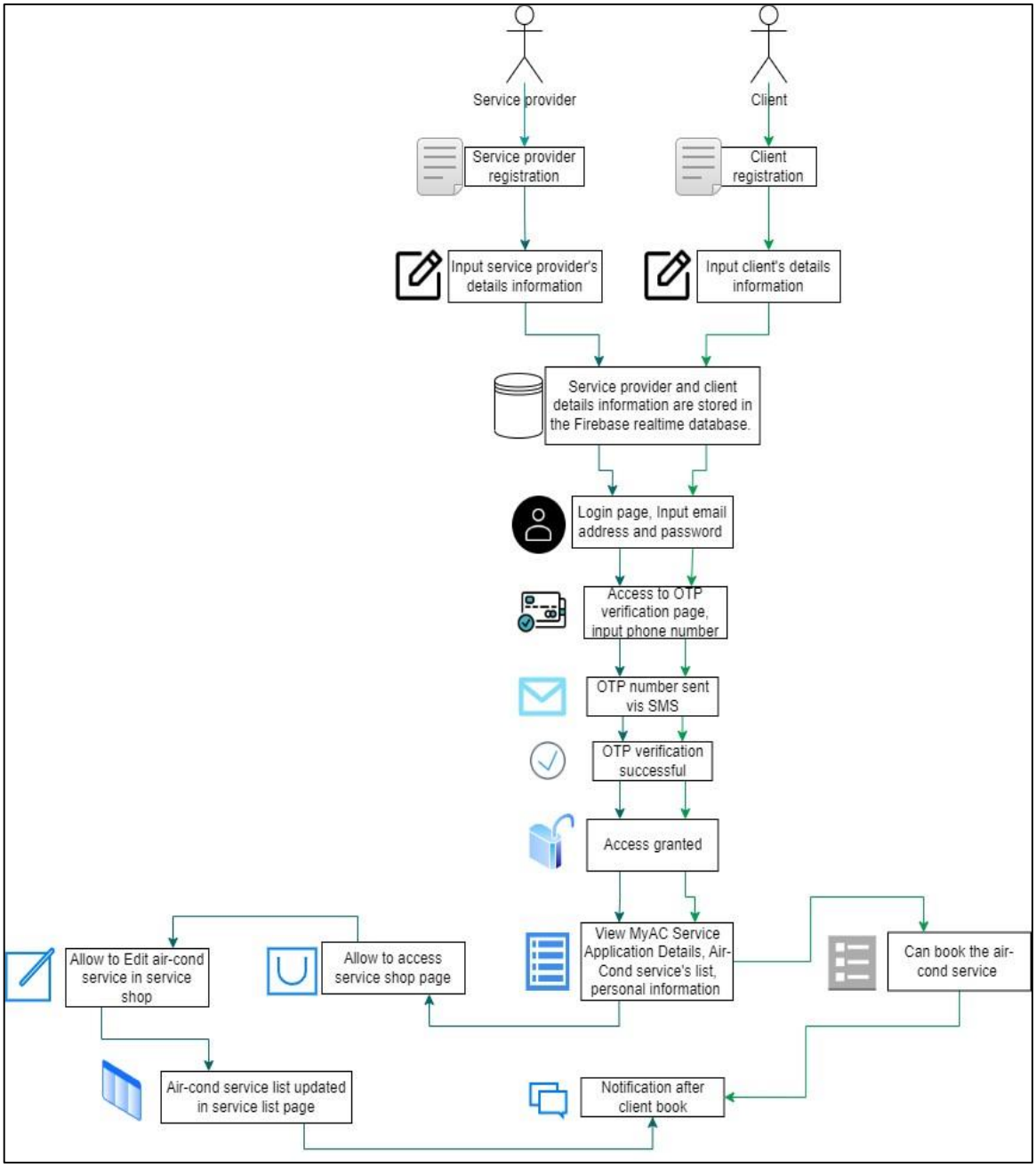

Figure 5: Architecture of Air Conditioner Service Application

4.2 Functional Requirements

The proposed system has eight modules as shown in Table 5.

Table 5: Functional requirements

Function	Functionality	User
Registration	Allow service provider to register their service provider in the application as a new service provider service	Service provider
Login	Allows to log in to their accounts using MFA which are using complex password and One-time Password.	service provider
Service Upload	Allow to enter, delete, and edit their air conditioner service information in the application.	service provider
Service List	Allow to view all the list of air conditioner services in the application	service provider
Booking List	Allow to view the successful booking from client.	service provider
Registration	Allow client to register in the application as a new user.	client
Login	Allows to log in to their accounts using MFA which are using complex password, and One-time Password.	client
Booking List	Allow to view all the list of air conditioner services in the application	client

4.3 Non-Functional Requirement

There are four categories of non-functional requirements, which are operational, performance, security, and usability shows in Table 6.

Table 6: Non-functional Requirement system

Requirement	Functionality
Operational	a) Application is only available for mobile devices with Android operating system version 10 or above. b) Application should be easily maintained. c) Application should be user friendly.
Performance	a) Application is available for users to use for 24 hours per day. b) Application's interaction between the user should not exceed 5 seconds.
Security	a) Only authorized users can login to the application using MFA method. b) Only service provider roles can get access to edit, update and delete the service shop page.
Usability	a) The application is using the English language which it is easy for the user to understand. b) Application allows users to enter many languages in order to save their information.

4.4 Data Flow Diagram Level 1 (DFD 1)

Data Flow Diagram (DFD) is a technique for organized analysis and design. It is a graphical tool for representing logic models and data transformation in a system [12]. Decomposition is supported to show the specifics of the data flows and functions. Figure 6 shows the shows data flow diagram level 1 which has four process registration, login, manage company service and manage booking. In the registration process, users have the option to become a client or service provider. As a client, need to input name, phone number, email, address, birth date, password. However, for service provider registration, system will ask to enter company name, phone number, email, address, first date of operation, password.

For login process, both users need to input email address, password, and OTP code to get access to the application. The login information will be stored in log database. Next, have a manage company service process which service provider roles can view, edit, delete, and update the air conditioner service data in service shop page. The air conditioner service will be stored in a service list database and the system will give a input viewer for both users. This application also has a managed booking process in which clients give the service type and price they choose in the service list for booking it. Then the system may store the booking data in a booking list and service provider should get notification of client booking.

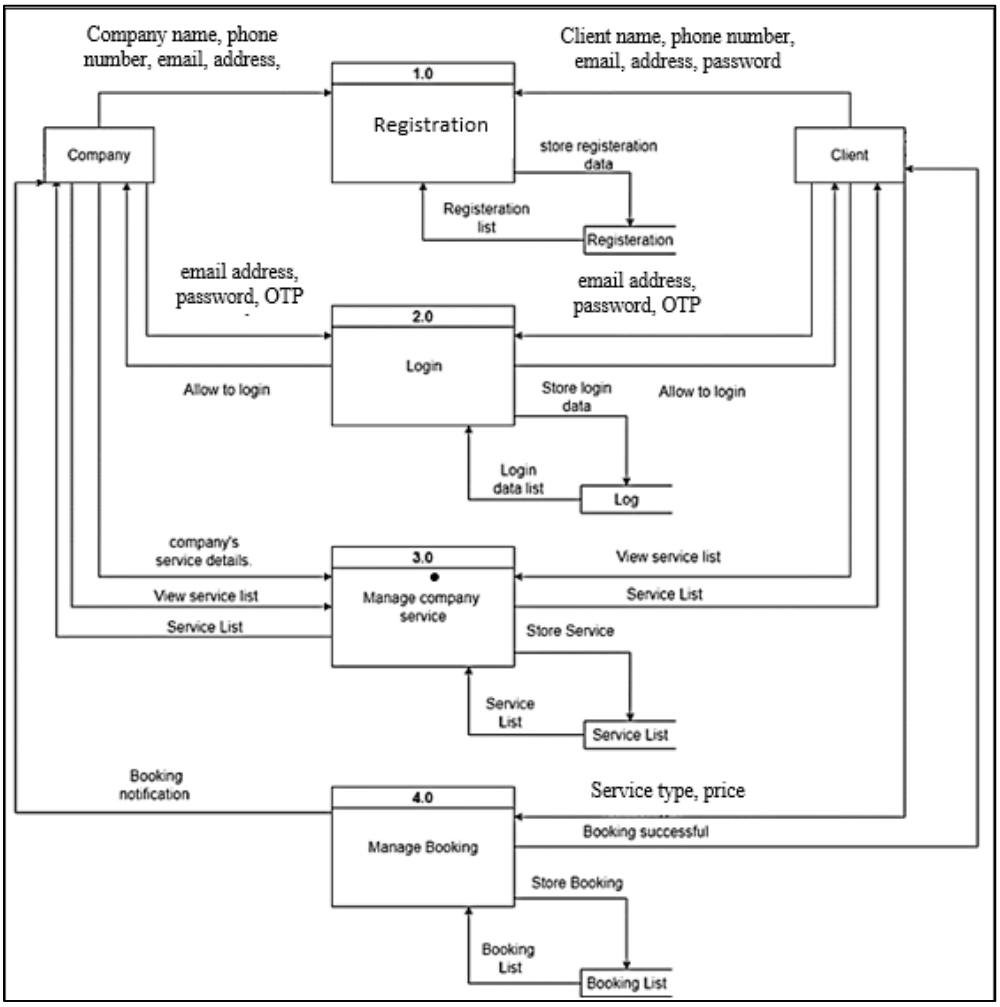

Figure 6: Data Flow Diagram Level 1 (DFD 1)

4.4 Design Interfaces

Figure 7 shows the design interfaces for MyAC Service application main page. The main page provided two options for users which they want to be a client or service provider. As a client, they can get access to the client page which they can do a register as client, see a service list, booking activity, get confirmation of booking. For service provider role, user can register as service provider, see a service list, edit service shop, give confirmation for client booking. After clicking button client, the system will ask user to register as client role before login to MyAC Service app booking session. Otherwise, for button service provider, system may direct to register as service provider role.

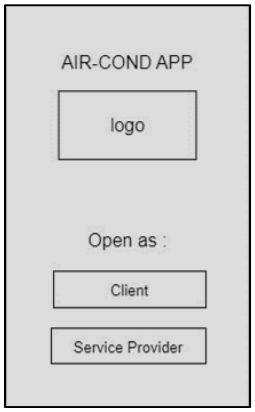

Figure 7: MyAC Service application Main page Design Interfaces

Figure 8 shows the design of the register client interface in MyAC Service application which user can register as client role. The system will ask the user to enter a few details information such as name, email address, address, phone number, password, and confirmation password. After that, click button register to go login page.

Figure 8: MyAC Service Application Register User Design Interfaces

Figure 9 illustrates the design of the login interface in MyAC Service application. This interface functions as a login session for this application. Users need to enter their email address and password then click the login button. User need to click the email verification before get access to the main interface of application.

Figure 9: MyAC Service Application Login Design Interfaces

Figure 10 shows a design for OTP verification page, which system will ask users to enter phone number and click send OTP button. Next, user need to enter the OTP number that successful send to the user and click button verify to verify the user identification.

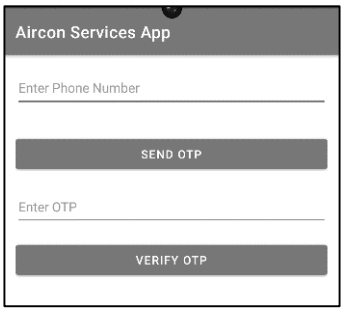

Figure 10: MyAC Service Application OTP verification Design Interfaces

Figure 11 displays the design of the list of air conditioner services in MyAC Service application. Within this interface, both users could access and view lists of services. However, only client roles can do a booking activity which client need to select their chosen services and click book now button to book the service.

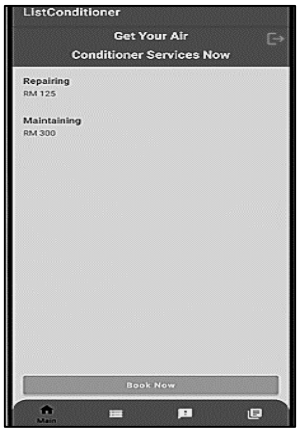

Figure 11: MyAC Service Application List Service Design Interfaces

Figure 12 displays the design of the notification page in MyAC Service application. In this interface, service provider should get booking notification after client book their company service while client will get successful booking notification.

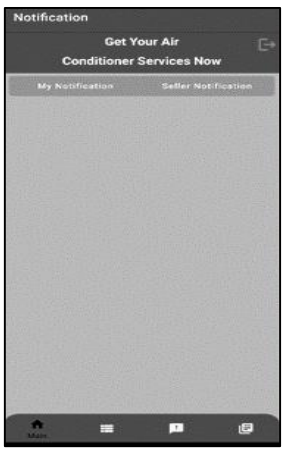

Figure 12: MyAC Service Application Notification Design Interfaces

Figure 13 presents the edit service design interface in MyAC Service application. This interface only be accessed by a registered service provider. As a service provider, the system will allow them to edit, delete their service shop.

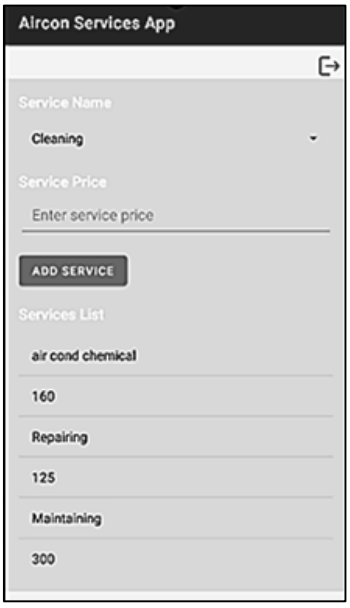

Figure 13: MyAC Service Application Edit Service Design Interfaces

5. Result and Discussion

This section presents the result and discussion of MyAC Service application, including the implementation of application and the result of the testing.

5.1 Implementation

This Figure 14 shows the validation password requirement code may be used as part of a login or password generation process to enforce a specific level of password difficulty. It verifies that the password matches the needed criteria, such as having a combination of lowercase and uppercase letters, digits, and special characters, by validating it against the provided regular expression. Implementing such complex criteria can improve user account security by encouraging users to choose tougher passwords that are resistant to brute-force attacks or guessing efforts. Figure 15 shows the alert message in registration interface ask the user to enter the password based on the requirement at least eight numeric digits, and non-alphanumeric characters if the user not enter the complexity password feature.

```
private boolean isValidPassword(String password) {
    String regex = "^(?=.*[a-z])(?=.*[A-Z])(?=.*\\d)(?=.*[#@$!%*?&])[A-Za-z\\d#@$!%*?&]{8,}$";
    Pattern pattern = Pattern.compile(regex);
    Matcher matcher = pattern.matcher(password);
    return matcher.matches();
}
```

Figure 14: The validation password requirement code

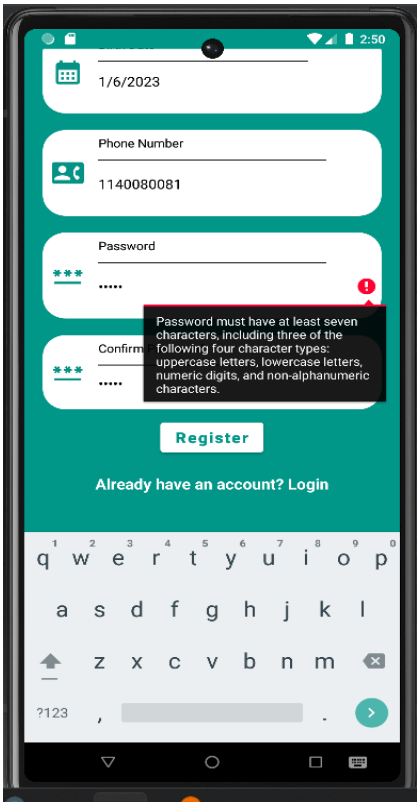

Figure 15: MyAC Service Application Alert Message in Registration Interface

This Figure 16 and Figure 17 show the signInWithPhoneAuthCredential function is called once the user inputs the OTP code and hits the "Verify" button. It accepts a PhoneAuthCredential object as an argument (which contains the verification code). The auth object is used within this function to sign in with the specified credential by using signInWithCredential. The addOnCompleteListener function is used to manage the completion of the sign-in process. If the task is completed successfully, the user is deemed authenticated. In this scenario, the code shows a toast message confirming successful verification, hides the progress bar, and navigates to the Dashboard activity. If the verification fails, the code determines if the error is of the type FirebaseAuthInvalidCredentialsException, indicating an incorrect verification code. It hides the progress bar and shows an appropriate toast message based on the outcome.

```
private void sendVerificationCode(String phoneNumber) {
    otpProgressBar.setVisibility(View.VISIBLE);

    PhoneAuthOptions options =
            PhoneAuthOptions.newBuilder(auth)
                    .setPhoneNumber(phoneNumber)
                    .setTimeout( timeout: 60L, TimeUnit.SECONDS)
                    .setActivity(this)
                    .setCallbacks(mCallbacks)
                    .build();

    PhoneAuthProvider.verifyPhoneNumber(options);
}
```

Figure 16: The send verification OTP code

```
2 usages
private void signInWithPhoneAuthCredential(PhoneAuthCredential credential) {
    auth.signInWithCredential(credential)
            .addOnCompleteListener( activity: this, new OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if (task.isSuccessful()) {
                        FirebaseUser user = task.getResult().getUser();
                        // Handle successful verification here
                        Toast.makeText( context: OtpVerification.this, text: "Verification successful", Toast.LENGTH_SHORT).show();
                        otpProgressBar.setVisibility(View.GONE);

                        // Navigate to Dashboard activity
                        Intent intent = new Intent( packageContext: OtpVerification.this, Dashboard.class);
                        startActivity(intent);
                        finish(); // Optional: Finish the current activity to prevent going back to it
                    } else {
                        if (task.getException() instanceof FirebaseAuthInvalidCredentialsException) {
                            Toast.makeText( context: OtpVerification.this, text: "Invalid verification code", Toast.LENGTH_SHORT).show();
                        } else {
                            Toast.makeText( context: OtpVerification.this, text: "Verification failed", Toast.LENGTH_SHORT).show();
                        }
                        otpProgressBar.setVisibility(View.GONE);
                    }
                }
            });
}
```

Figure 17: The code for sign in with phone authentication credential

The application offers OTP verification for user login by employing these techniques. The sendVerificationCode function starts the SMS verification process by sending the OTP to the supplied phone number, whereas the signInWithPhoneAuthCredential method processes the verification result and takes the appropriate action based on it. This guarantees that only users with a valid OTP may successfully authenticate and access Dashboard activities, hence improving login security. Besides that, when the user gets the OTP number, and the system will display message “OTP sent”. After that, user can do verification using OTP to get access to the dashboard page and the system may display a message “Verification successful” in dashboard page. The OTP verification is successful as shown in Figure 18.

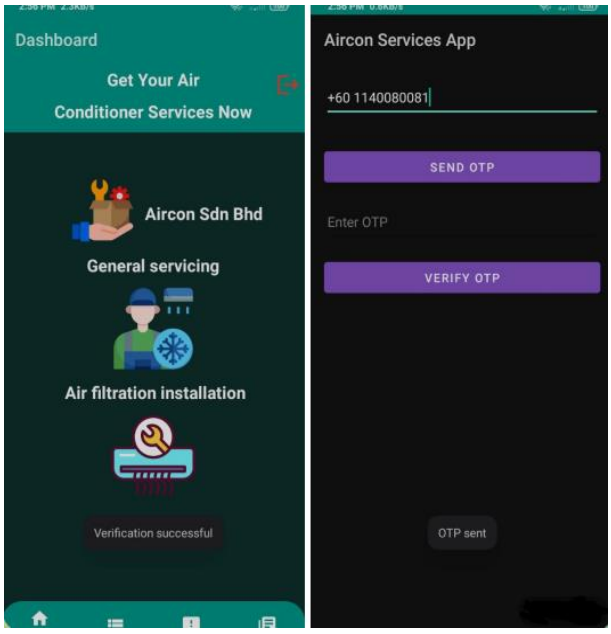

Figure 18: OTP verification successful.

5.2 System Functionality Testing

In this section, the test plan will be carried out to determine the result of MyAC Service application. Table 7 show the test plan result of MyAC

Table 7: Test Plan Result

No	Requirement	Result
1.	Both user able to login using complex password and OTP verification.	Pass
2.	User for client roles able to do a booking air conditioner service.	Pass
3.	User for service provider roles able to edit, delete, and update their service shop.	Pass
4.	Both user able to view the service list of air conditioner service.	Pass
5.	Both user able get successful booking notification.	Pass

5.3 User Acceptance Testing

In this section, the user acceptance testing is carried out to confirm that the developed application meets those desired goals. The user acceptance is created using Google Form and 30 android phone users were chosen randomly for this testing. There are three sections of testing which are interface satisfaction, functionalities satisfaction and security checklist. Figure 19 shows the results interface satisfaction level of MyAC Service. The question interface testing is easy to use and understand, navigation, text (font family, font size), layout for the content (colour, background), and interface design. Most of the responders choose option 5 which is strongly satisfy. According to the findings of the system interface testing, all respondents were satisfied with the application interface designed.

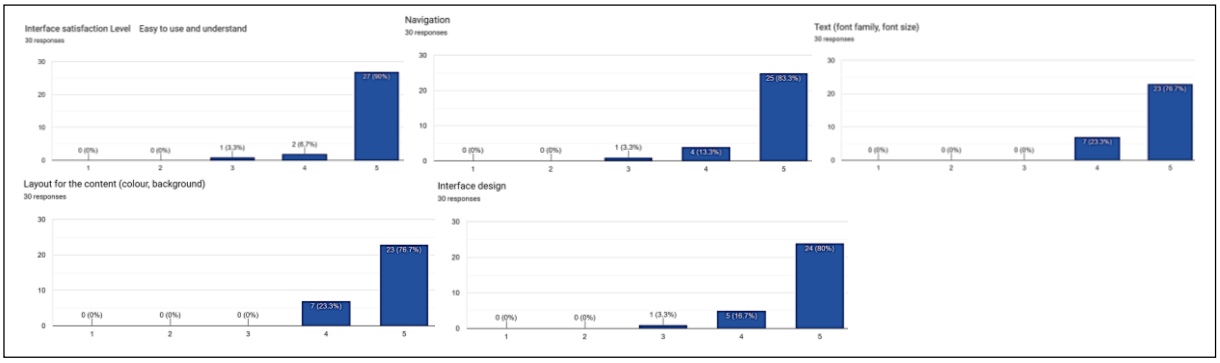

Figure 19: Results interface satisfaction of MyAC Service

Based on Figure 20, the testing section for functionalities satisfaction level. There are six questions which are the system can execute from start to end, users can register a new account, user can login to the application with their email address and password, user can view air conditioner service list, users can manage their service shop, and user can get notification after booking activity. The majority of responders are chosen strongly to satisfy in the testing section for functionalities satisfaction. Thus, from the findings of the functionality's satisfaction testing, all respondents were satisfied with the application functionality.

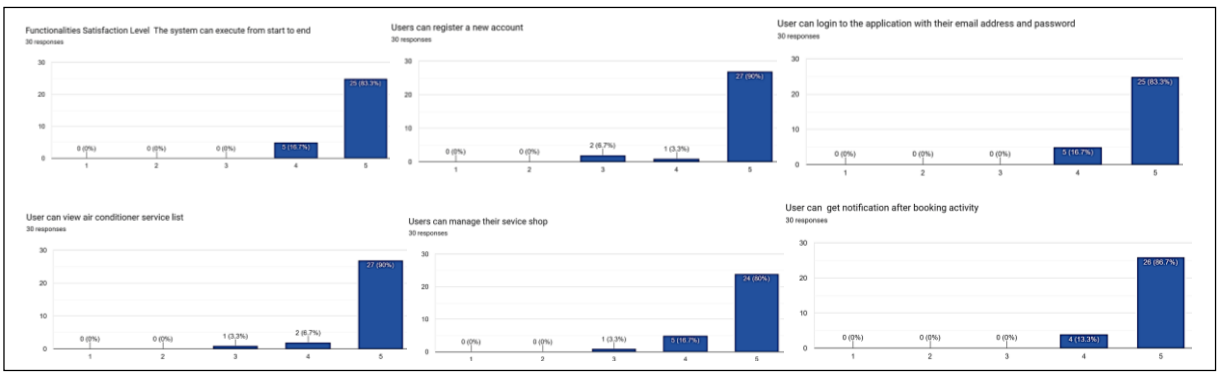

Figure 20: Results functionalities satisfaction of MyAC Service

Figure 21 displays the testing section for security checklist. There are seven testing statements which require users to make a strong password. If not, system will print out error message " Password must have at least seven characters, including three of the following four character types: uppercase letters, lowercase letters, numeric digits, and non-alphanumeric characters ", ensure the complexity of the password including uppercase letters, lowercase letters, numeric digits, and non-alphanumeric characters, enforce the length of the password at least 8 characters, successfully login after done One-Time Password (OTP) verification, print out error message "Invalid OTP code number" for invalid OTP code number, only user that verify their email can get access to home page of application and successfully get OTP code number via SMS. All of the responders are chosen pass option all the security checklists. Therefore, from the findings of the application's security testing, all respondents agreed with the implement security in the MyAC service application.

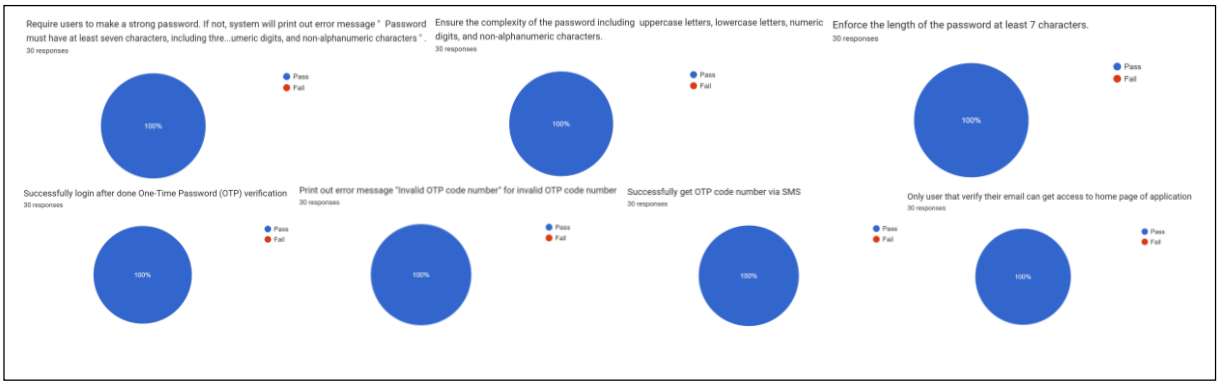

Figure 21: Results for security checklist testing of MyAC Service.

6. Conclusion and Future Works

This project aims to enhance the security and reliability of an Android-based air conditioner service application through the development and implementation of a multifactor authentication (MFA) system. The MFA system will include complex passwords and one-time passwords (OTPs) to provide multiple layers of protection against unauthorized access and cyber-attacks. The expected outcomes of the project include improved security, enhanced confidentiality, integrity, and availability of the system, improved user experience and satisfaction, and identification of the advantages and challenges of MFA in the air conditioner service.

The implementation for future work is implemented the password change functionality. From the limitations of MyAC Service application, users do not have ability to change their password after registering. Thus, developers for future work can develop an air conditioner service application feature that allows users to reset their passwords. This feature may contain a form in which users submit their current password as well as a new password. Implement validation checks to validate the new password's strength and complexity. Besides that, improvements for this application are implemented this application for IOS user which can increase the potential users of MyAC service application. The user will be blocked after three times of fail attempts to prevent a brute force attack. In addition, there are no functions for users to change their password. Thus, the future implementation is providing a link for the user to reset their password if the password is forgotten.

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] K. Zaky and D. H. Saxe, "Multi-factor Authentication," *IDPro Body of Knowledge*, vol. 1, no. 10, 2022.

[2] A. Ometov, S. Bezzateev, N. Mäkitalo, S. Andreev, T. Mikkonen, and Y. Koucheryavy, "Multi-Factor Authentication: A Survey," *Cryptography*, vol. 2, Jun. 2018, doi: 10.3390/cryptography2010001.

[3] M. Awad, Z. Al-Qudah, S. Idwan, and A. H. Jallad, "Password security: Password behavior analysis at a small university," in *2016 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA)*, 2016, pp. 1–4.

[4] K. Aravindhan and R. R. Karthiga, "One time password: A survey," *International Journal of Emerging Trends in Engineering and Development*, vol. 1, no. 3, pp. 613–623, 2013.

[5] E. Erdem and M. T. Sandıkkaya, "OTPaaS—One Time Password as a Service," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 3, pp. 743–756, Mar. 2019, doi: 10.1109/TIFS.2018.2866025.

[6] S. Das, B. Wang, Z. Tingle, and L. J. Camp, "Evaluating user perception of multi-factor authentication: A systematic review," *arXiv preprint arXiv:1908.05901*, 2019.

[7] Y. Fang, Y. Yang, and C. Huang, "EmailDetective: an email authorship identification and verification model," *Comput J*, vol. 63, no. 11, pp. 1775–1787, 2020.

[8] S. The, "Book your next air-cond cleaning service with the go daikin app," Sep. 30, 2021.

[9] H. Halid, "Journal Article Publication: Business Case Paper DoneIT : Online Platform for Services," Jun. 2018.

[10] R. R. Porle and N. N. M. Saiful, "Android-based Booking Application for Smart Parking System," in *2021 IEEE 19th Student Conference on Research and Development (SCOReD)*, 2021, pp. 290–294.

[11] S. S. Kute and S. D. Thorat, "A review on various software development life cycle (SDLC) models," *International Journal of Research in Computer and Communication Technology*, vol. 3, no. 7, pp. 778–779, 2014.

[12] Q. Li and Y.-L. Chen, "Data flow diagram," in *Modeling and Analysis of Enterprise and Information Systems*, Springer, 2009, pp. 85–97.