Applied Information Technology And Computer Science Vol. 5 No. 1 (2024) 905-921

AITCS

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

Barber Booking Mobile Application

Sivaneeswaran Krishnan[1], Suhaila Mohd. Yasin[1*]

[1]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.2024.05.01.051
Received 24 June 2023; Accepted 18 June 2024; Available online 30 August 2024

Abstract: Barbers offer haircuts and trimming to customers at the barbershop. Barbers traditionally offer haircuts and trimming on a first-come, first-served basis. However, this can lead to long queues and a disorganized process. To address this, a mobile application called "Barber Booking Mobile Application" was developed. The application allows customers to book their time slots in advance, view prices, and make payments through online banking. This streamlines the booking process for both customers and barbers and provides a more efficient and organized experience for everyone involved. The application was developed using the Scrum methodology and object-oriented approach. It is accessible via an Android mobile device and is used by both customers and barbers. The overall outcome of the Barber Booking Mobile Application is a more efficient and organized process for barbershops, enhanced customer experience, and improved customer and appointment management.

Keywords*: Barbering Services, Mobile Booking Application, Database*

1. Introduction

The first barbering services were performed by Egyptians in 5000 B.C. with instruments they had made from oyster shells or sharpened flint [1]. In modern times, barbershops are places where people go to get a haircut or a beard shave as well. Commonly, a barbershop is patronized by men. Barbershops are different from hair salons in terms of their simple, unsophisticated interior and their use of universally recognized trademarks like a 'barber's pole'.

Monkey Barbershop [2] is a unique barber concept located in Taman Miharja, Kuala Lumpur. The barber is trained and has experience in cutting, trimming, styling, shaping, and maintaining the hair or facial hair of the customer. In order to provide high-quality customer service, the barber must be able to manage customers well. The barbershop uses the "first come, first served" principle, which means that a group of people or items will be handled or provided in the order in which they arrive [3]. It is not uncommon for several customers to visit the barbershop at the same time. Therefore, the barber is unable to keep track of customers and welcome them properly. The barbershop is unable to keep track of its customers and financial performance because there is no digital way to process payments. In order to schedule an appointment, regular clients must call the barbershop directly. Customers must wait to receive barber services if there are customers during the arrangement time.

**Corresponding author: ysuhaila@uthm.edu.my*

Thus, a barber booking mobile application helps connect potential clients with various local barbers in an online community. This allows users to browse for barbers, book haircut appointments, and cancel existing appointments. An appointment can be reserved if a barber and a slot is available at the time. Consequently, barbers can receive a list of booked customers and notify them before barbering services are provided. As a result, the administrator has access to a convenient format for reviewing and managing appointments. The main purpose of the barber booking mobile application is to save time and reduce the complexity of finding an available barber.

The objectives of the project are to design, develop, and test an online barber booking mobile application for Android device users. The application will be designed to be user-friendly and intuitive, and will make it easy for users to book appointments, view their booking history, and track their spending. The application will be developed using Android Studio, and will be tested for functionality and usability. In addition to these objectives, the project will also focus on security, scalability, and maintainability.

This paper contains five sections. Section 1 describes the project background. Section 2 covers the related works. Section 3 elaborates on the methodology, while Section 4 summarizes the result and discussion. Lastly, Section 5 gives the conclusion.

2. Related Work

A barber booking system refers to a system that efficiently manages customers and the availability of barbers or barbershops. A barber booking system is used to book and schedule appointments for barbers or barbershops. Barbers who are sometimes called hair stylists, cut and style hair and perform other beauty services [4]. A barber is responsible for providing customers with the best price and service. The barbershop is one of the places where people get their hair cut.

Mobile computing is now a reality as a result of the fusion of two technologies: the introduction of potent portable computers and the growth of quick, dependable networks. Mobility and portability will give rise to a completely new category of applications and perhaps even totally new, sizable markets for the convergence of personal computing and consumer electronics [5]. Typically, mobile technologies use less energy and are physically smaller than desktop alternatives. Despite not performing as well as desktop goods, they are nonetheless offered at a range of prices and conveniences. As the use of edge computing, Internet of Things (IoT), and 5G technologies increases, mobile computing is anticipated to become more and more significant in people's lives. Information systems will become more integrated as a result of mobile and distributed computing.

The online booking systems manage reservations for services. The world of online business is practically inseparable since most businesses have a recognized online presence, regardless of their industry [6]. In order to connect people and businesses, the Internet plays an important role [7]. Online booking systems enable businesses to convert customers into revenue through modern technology [8]. The online booking system certainly helps to boost businesses as it is open 24 hours a day. The booking process can be completed anywhere and at any time. Subsequently, customers can save time by booking and using it easily. There is a need for better-embedded processors as mobile devices become more robust and adaptable than before [9]. Mobile applications are simple, user-friendly, reasonably priced, downloadable, and compatible with the majority of mobile phones, including entry-level and budget models. Internet-based mobile applications in particular are useful [10]. Since the development of mobile devices, mobile-based applications have seen a greater increase in usage than web-based ones. Utilizing and understanding technology has been streamlined.

The barber industry has developed over time to become barbershops as we know them today [10]. A university or college now offers a program that focuses on hairdressing and beauty as the barber industry has grown [11]. Monkey Barbershop uses the manual method of conducting business, which defines the barbershop as inefficient handling and pressuring on the barber [2]. Several existing systems

are analysed to produce a solution to the current issues. Firstly, BookedIN system automates all those tedious appointment scheduling tasks and allows one to spend more time enjoying the workday [12]. The system facilitates tasks like arranging client appointments, reminding clients, getting clients to confirm, collecting deposits, organizing the team, rescheduling, and cancelling. Next, the Barber's Room is a conventional grooming establishment for the contemporary man [13]. The system offers a website-based and online booking mechanism. The website will offer a list of services along with their costs and wait times. The available barbers will be displayed together with the date and hour that enables the customer to select an appropriate time. Finally, JustBookSalon is an online salon portal that enables customers to browse, choose, and reserve the salon services that the salons have to offer [14]. Through this website, customers can access their treatment packages in the salon profiles that correspond with them as well as receive email and SMS reminders about their reservations.

The three similar systems were investigated and compared with the proposed system. The results of this study are important to identify the features that can be implemented in the proposed system. Table 1 summarizes the comparison between BookedIN: Appointment Scheduling & Reminders, The Barber's Room System, JustBookSalon System and Barber Booking Mobile Application.

Table 1: Comparison of the existing system

System / Features	BookedIN: Appointment Scheduling & Reminders	The Barber's Room System	JustBookSalon System	Barber Booking Mobile Application
Mobile-based	Yes	No	Yes	Yes
Registration	Yes	Yes	Yes	Yes
Login	Yes	Yes	Yes	Yes
Slot booking/cancellation	Yes	Yes	Yes	Yes
Manage customers details and barbershop	Yes	No	No	Yes
Provide location searching for customers	No	Yes	Yes	Yes
Display price	No	Yes	No	Yes
Can make advance payment	No	No	No	Yes

Based on Table 1, In today's era of digital technology, when all information must constantly be available, the features are crucial. Taking this into consideration, the Barber Booking Mobile Application will include all the functionality required to run an online booking system effectively and efficiently. If there is a booking, the application will let everyone know about it, and it will also let clients know what time is open. Customers can schedule a time slot to receive haircut services without having to wait, and the application will show all haircut pricing. The consumer only needs to log in, look for a barbershop that is open nearby, reserve a time slot, pay in advance for the reservation, and then arrive at the time period selected.

3. Methodology/Framework

In developing the Barber Booking Mobile Application, the Scrum model is chosen because it is the best concept for the development as it quickly expanded to handling big, complicated projects that might have otherwise taken a lot of time to finish [15] and it provides a clear view of an entire project, works involved, budget estimates, as well as timelines. Phases of the Scrum model include product backlog, agile sprint backlog and planning, daily scrum meeting, sprint, agile sprint review and retrospective and next sprint planning [15].

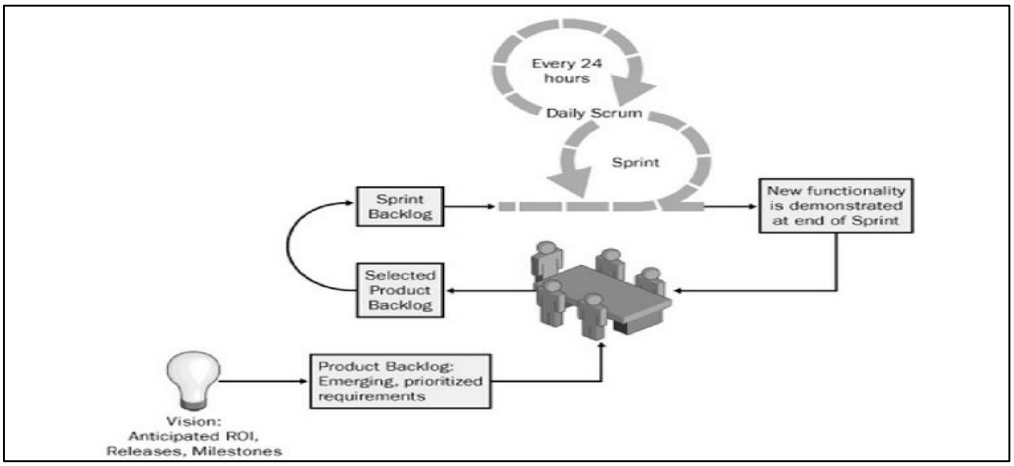

Figure 1: Scrum process overview [15]

The level of requirements details can be classified into two types, which are functional and non-functional requirements [15]. Functional requirements are defined as a set of actions or features that must be included in an application to support the purpose of the system. Non-functional requirements are defined to present the performance of the tasks in terms of the overall application [16]. The functional requirements are summarized in **Table 2**, while the non-functional requirements are summarized in **Table 3**.

Table 2: Functional requirements.

Modules	Function
Registration	• The administrator will accept the registration after checking the barber or barbershop details. • The user/customer must register in the mobile application.
Login	• A username or password required the customer to enter the mobile application. • A username or password required the barber/barbershop to enter the mobile application.
Slot booking/ cancellation	• The customer is able to book or cancel slot for barber services.
Manage customers	• Barber/barbershop will have a view of booked slot and send the notification to the next customers.
Manage location	• The customer and barber will decide the location. • For the barbershop, the shop location will be used.
Payment/Refund	• The customer must pay for booking. • The refund is direct to the customer.
Logout	• Logout user from the system.

Table 3: Non-functional requirements.

Requirements	Description
Performance	• The barber booking mobile application can multiple users at a time. • The barber booking mobile application load time for users should not exceed 10 seconds.
Scalability	• The barber booking mobile application should be able to adapt to growing usage or manage additional data as time goes on.
Responsiveness	• The barber booking mobile application must respond quickly to user input and any external interruptions with the highest priority and should then resume its previous state.
Usability	• Users should be able to use the barber booking mobile application without instructions or assistance from professionals or manuals.
Reliability	• User activity that is significant should be acknowledged with confirmation by the barber booking mobile application.
Security	• Users can only access their own account with a username and password. • Access to the customer personal information and barbershop transactions is restricted to authorized users only. • The barber booking mobile application must preserve backups of all database updates for each record transaction in order to maintain data integrity.
Extensibility	• The barber booking mobile application needs the ability to add new functional requirements to an app in the future.
Availability	• The system ought to be accessible around the clock. • The system should have simple access to the Internet. The system should have simple access to the Internet.

User requirement analysis is essential to discover and list out the user requirements. The description of requirements might be the functions or settings that will support the application operation. The requirements can be presented in terms of organization, usability, or function mapping [17]. User requirements are listed in Table 4. These are the functionality expectation from the users.

Table 4: User requirements.

No	User requirements
1.	New customers can register to the system.
2.	Existing customers can login to the system.
3.	Customers can view all the shops that are listed in the application.
4.	Customers can filter by location.
5.	Customers can view the available slot in the preferred barbershop.
6.	Customers can select their preferred time slot.
7.	Customers can view the pricing of the services available in the barbershop.
8.	Customer will receive notification.
9.	Customers will have an advanced booking for the services.
10.	Customers can pay using an online banking payment method.
11.	Customers can cancel the booking.
12.	The customer can logout of the system.

13.	The barber can login using given username and password by administrator.
14.	Barber can view all the booked slot in their barbershop.
15.	Barber can view customer name and selected services.
16.	Barber can send notification to the user.
17.	Barber can update the list.
18.	Barber can cancel any bookings.
19.	Barber can update their information including location.
20.	Barber can change the services price.
21.	Barber can logout from the system.

The use case diagram from object-oriented approach is used to present the system requirements with use cases, identify the actors and show the interaction of actors with the use cases [18]. Figure 2 shows the use case diagram of the proposed application with two actors: Customer and Barber. There are total seven use cases which are login, registration, manage customers, manage slot booking/cancellation, manage location, manage payment/refund, and logout.

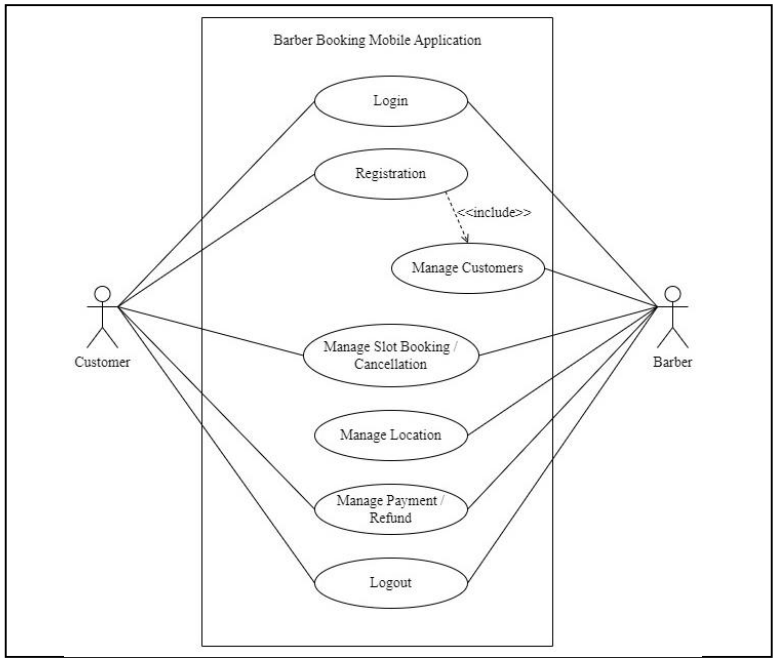

Figure 2: Use case diagram

Class diagram is an overview of application's attributes and functions that are expressed as classes and the relationship between the classes [19]. According to Figure 3, the application has seven classes, including Login Details, Services, Booking, and Payment aggregated to Online Banking, as well as three generalizing classes for new customer to customer and barbershop users.

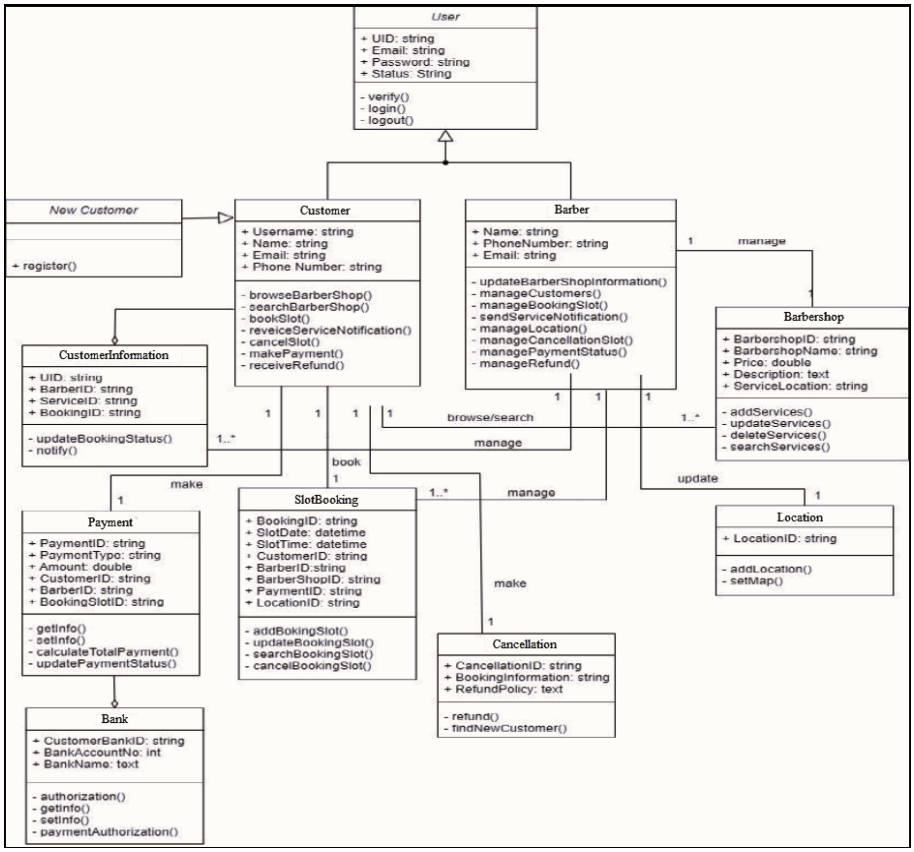

Figure 3: Class diagram

The proposed application seems to be more effective as it implemented the solutions for the problems stated. As part of the registration process, the customer will provide some information about themselves. The barber can view the list of customers whose names are registered with the barbershop. In the next step, the customer can search for preferred barber shops and book a time slot. After booking the slot, the customer can pay the barber service owner through online banking. Customers who cannot attend the barbering service can cancel the slot. It is easier for the barber to manage the customers, booking slots, cancellations issue, payments or refunds issue through the application. Accordingly, the To-Be diagram in Figure 4 shows the integration of the existing data management process.

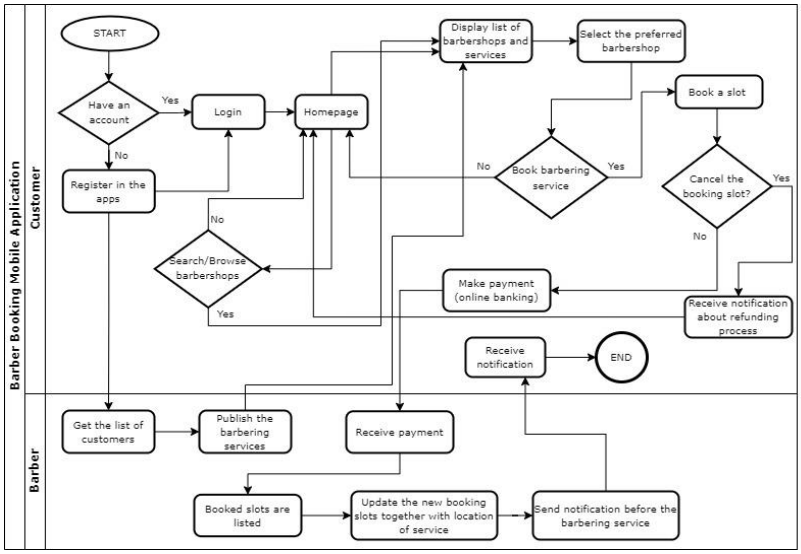

Figure 4: Flowchart

The designs are created according to the requirements information gathered at product backlog phase. The multi-layered architecture is a client-server software architecture pattern in which the user interface (UI-presentation), process logic ("business rules"), storage and access to data are developed and maintained as independent modules, most often on separate platforms. A multi-layered architecture allows for upgrades and replacements of any of the three tiers in response to changing requirements or technology. Figure 5 depicts the basic multi-layer system architecture for the Barber Booking Mobile Application.

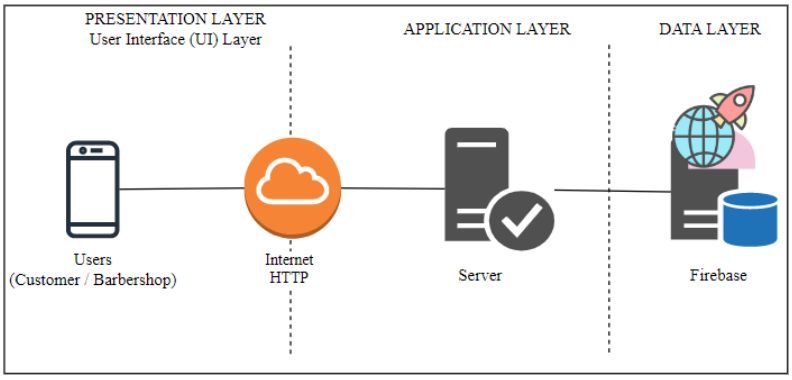

Figure 5: System Architecture

Database schema is listed as follows:

i. `Login(UID,Email,Password,Status)`
ii. `Customer(Username,Email,PhoneNumber,UID)`
iii. `Barbershop(BarbershopID,BarbershopName,ShopLocation,ShopTime,UID)`
iv. `Booking(BookingID,UID,BarbershopID,ServiceID,PaymentID,Date,Time,TotalPrice)`
v. `Services (ServiceID, ServiceName, ServicePrice)`
vi. `Payment(PaymentID,PaymentType,PaymentMade)`

As a preview of the application, Figures 6 to 17 illustrate the design of the interface, from the login interface to the logout interface. The Barber Booking Mobile Application will be used by both the customer and the barber.

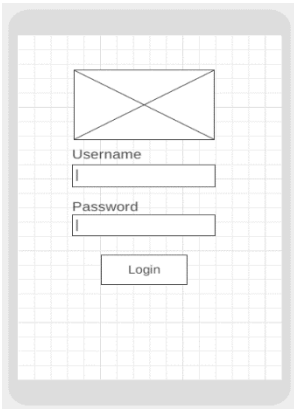

Figure 6: Login Interface

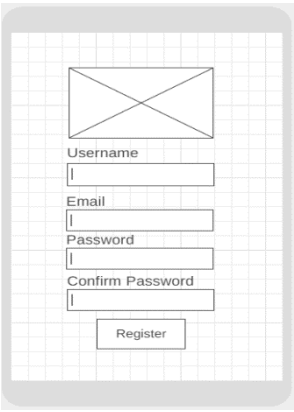

Figure 7: Registration Interface

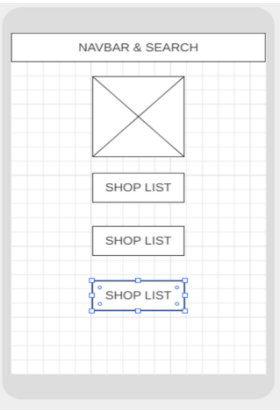

Figure 8: Homepage Interface

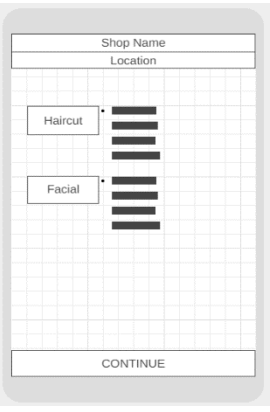

Figure 9: Services Interface

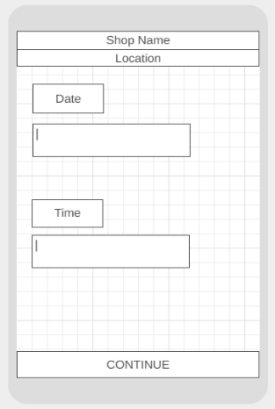

Figure 10: Appointment Interface

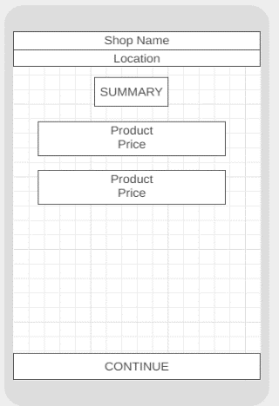

Figure 11: Summary Interface

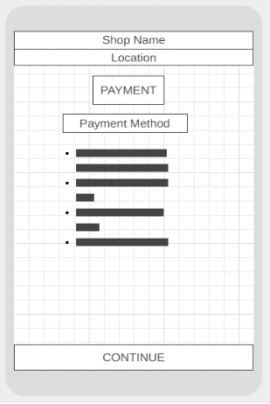

Figure 12: Payment Interface

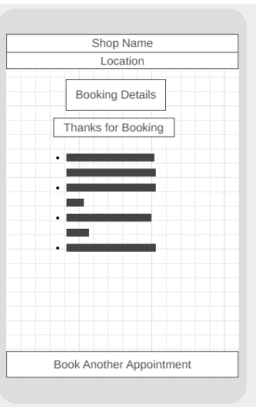

Figure 13: Booking Details Interface

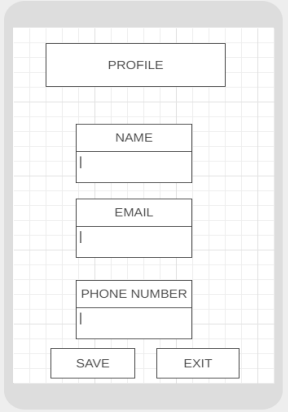

Figure 14: Profile Interface

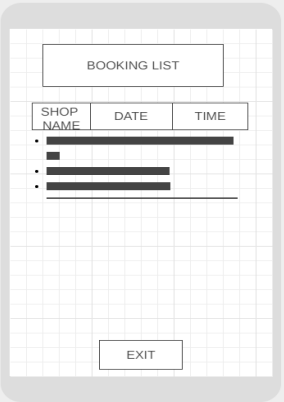

Figure 15: Booking List Interface

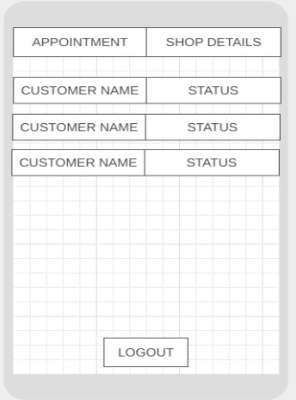

Figure 16: Barber Appointment List Interface

Figure 17: Barbershop Details Interface

4. Results and Discussion

The Barbershop Booking Mobile Application consists of two sides, customer, and barbershop. It is developed using Android Extensible Markup Language (XML) and Java, with the layout designed in XML and the functionality implemented in Java. The Firebase Database is used as the project's database, allowing for real-time data storage and synchronization among users. Android Studio is the Integrated Development Environment (IDE) used for programming, and the Android emulator within Android Studio enables testing on a PC. The results of the user testing are explained in the other subsection.

4.1 Module Implementation

(i) Registration Module

The registration module was implemented to allow users to create an account within the system. Figure 18 illustrates the coding segment of the registration module. In Figure 18, the process of gathering user-entered data and creating an account is demonstrated. The code retrieves the necessary information from the user, such as username, email address, password, and confirmation password. The `addOnCompleteListener` is utilized to determine the success of the account creation process. Upon successful account creation, an email verification is automatically sent to the user's provided

email address. This verification email contains a link that the user must click to validate their email address and gain access to the system. Figure 19 showcases the registration layout of the system. The user interface prompts the user to input their username, email address, password, and confirm password to register. These details are mandatory and essential for creating an account within the system. Overall, the registration module provides a seamless process for users to create an account, complete with email verification to enhance security and ensure valid user credentials.

```
RegistrationInfo registrationInfo = new RegistrationInfo(username.getText().toString(),email.getText().toString(),
        password.getText().toString(), role: "customer");
RegistrationInfo registrationInfo2 = new RegistrationInfo(username.getText().toString(),email.getText().toString(),
        password.getText().toString(), role: "customer", phoneNumber: "-");
mAuth.createUserWithEmailAndPassword(email.getText().toString(), password.getText().toString())
        .addOnCompleteListener( activity: Register.this, new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if (task.isSuccessful()) {
                    // Sign in success, update UI with the signed-in user's information
                    Log.d(TAG, msg: "createUserWithEmail:success");
                    FirebaseUser user = mAuth.getCurrentUser();
                    mDatabase.child("Login").child(mAuth.getUid()).setValue(registrationInfo);
                    mDatabase.child("Customer").child(mAuth.getUid()).setValue(registrationInfo2);
                    user.sendEmailVerification().addOnCompleteListener(new OnCompleteListener<Void>() {
                        @Override
                        public void onComplete(@NonNull Task<Void> task) {
                            Toast.makeText( context: Register.this, text: "Register successful.Please check your email",Toast.LENGTH_SHORT).show();
                            Intent intent = new Intent( packageContext: Register.this,MainActivity.class);
                            startActivity(intent);
                        }
                    });

                } else {
                    // If sign in fails, display a message to the user.
                    Log.e(TAG, msg: "createUserWithEmail:failure", task.getException());
                    Toast.makeText( context: Register.this, text: "Register unsuccessful",
                            Toast.LENGTH_SHORT).show();
                }
            }
        });
```

Figure 18: Registration code segment

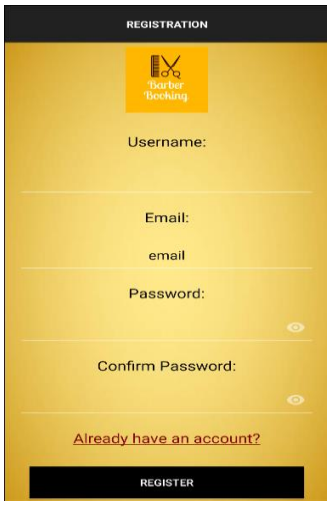

Figure 19: Registration interface

(ii) Login Module

The login module was implemented to enable users to log into the system. Figure 20 illustrates the coding implementation and user interface of this module. In Figure 20, the `addOnCompleteListener` is demonstrated with the `signInWithEmailAndPassword` method. This method is responsible for logging in using the email and password entered by the user, as shown in Figure 21. The `signInWithEmailAndPassword` method verifies if the email and password match the information stored in the Firebase authentication. If the email and password are correct, the login process proceeds. However, if there is an error or the provided credentials are invalid, an error message will be displayed to the user. Furthermore, the login module incorporates an additional check to verify if the user has already verified their email address. If the email is not yet verified, a message prompting the user to verify their email will be displayed. Figure 21 represents the user interface of the login module. Users are required to enter their email and password in the designated fields to initiate the login process. In summary, the login module utilizes Firebase authentication to validate user credentials and provides feedback to users regarding the success or failure of the login attempt. Additionally, it includes an email verification check to ensure the user has completed the necessary email verification step before accessing the system.

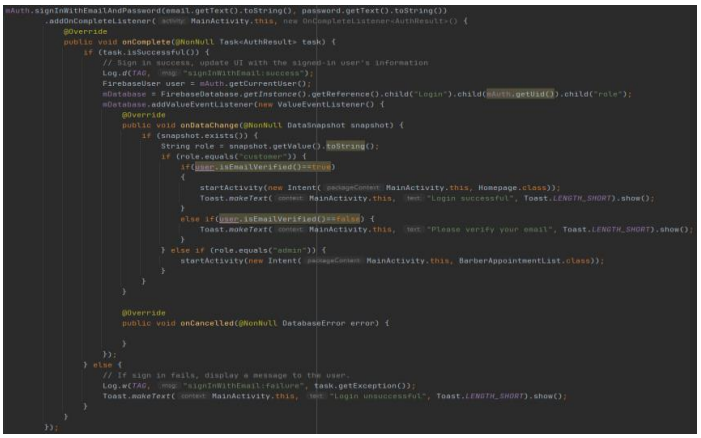

Figure 20: Login code segment

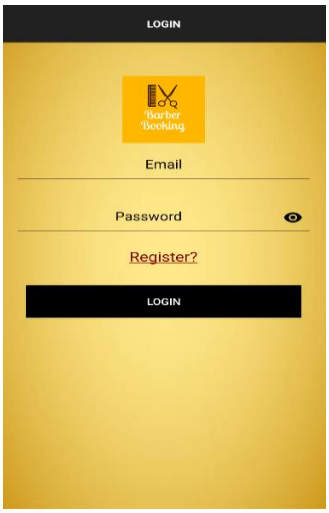

Figure 21: Login interface

(iii) Slot booking/ Cancellation Module

The slot booking/cancellation module was implemented to allow users to book or cancel slots within the system. Figure 22 demonstrates how the application inserts the booking details into the Firebase database, ensuring that the information is stored accurately in the database. Figure 24 showcases the layout for a confirmed booking, which displays all the necessary information related to the booking. In Figure 23, the cancellation process is depicted. When a booking is cancelled, the application changes the status of the current booking to `"cancel"` in the database. The application also checks the availability and status of slots to determine if they have been cancelled, completed, or are still pending. Figure 25 provides an example of a cancelled booking. Once a booking is cancelled by the admin or customer, the slot becomes available for other customers to book. The cancelled slot is displayed to both the customer and admin for record-keeping purposes. Overall, the slot booking/cancellation module allows users to book and cancel slots while ensuring accurate data storage and availability updates. The system provides confirmation of bookings and handles cancellations effectively, allowing for efficient management of available slots.

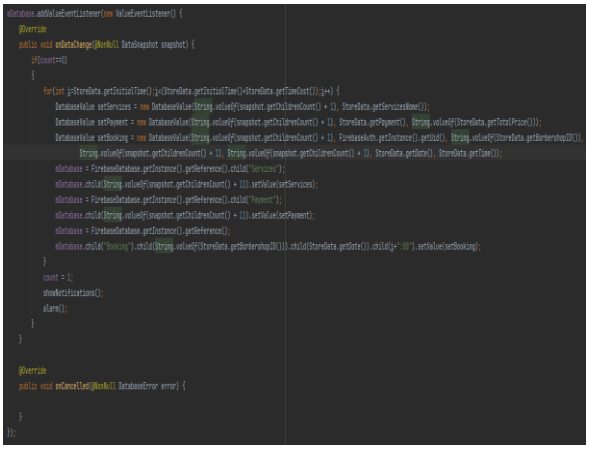

Figure 22: Slot Booking Code Segment

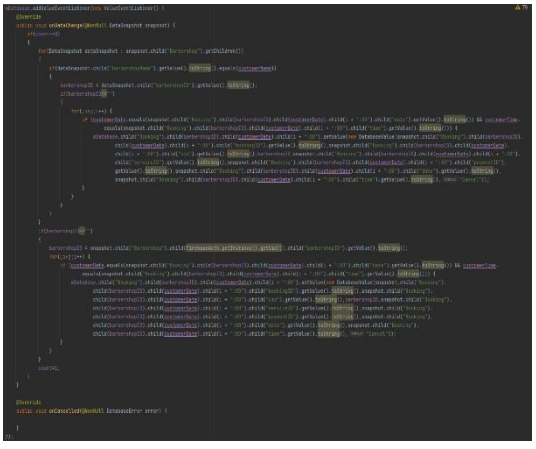

Figure 23: Slot Cancellation Code Segment

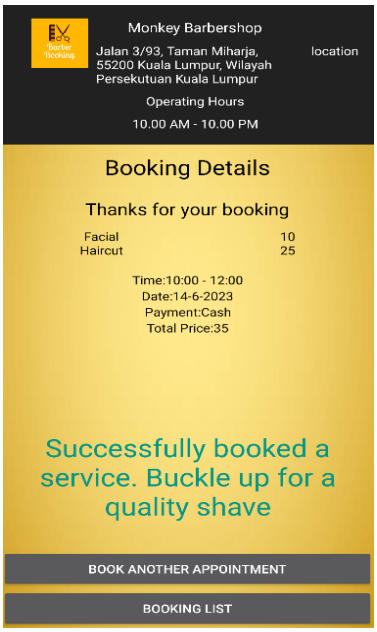

Figure 24: Slot Booking Interface

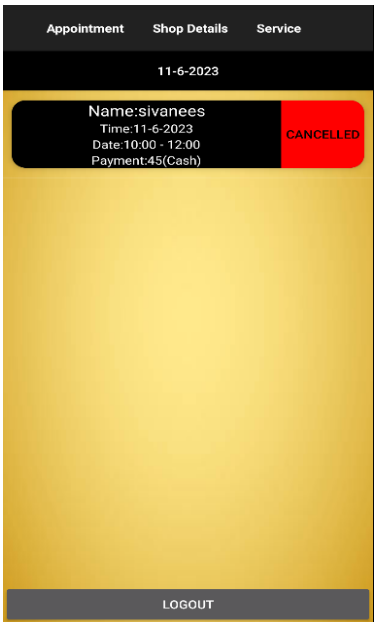

Figure 25: Slot Cancellation Interface

(iv) Manage Customers Module

The manage customers module allows barbershops to view and update customer bookings. In Figure 26, the details of the customer data are displayed to the barbershop based on the current `"BarbershopID"` and the booking's `"status"`. The application checks if a booking exists for the barbershop and if it does, the booking is displayed according to the date. The barbershop can update the status of pending bookings as either `"done"` or `"cancelled"`. When an order is completed, the barbershop needs to update the booking status as `"done"`. Figure 27 provides an example of a pending booking and presents options for the barbershop to update the status. The `">"` button notifies the user with a mobile notification for the next customer in line to receive the barbershop services. The manage customers module streamlines the process of viewing and updating customer bookings for barbershops, facilitating efficient management of appointments, and providing a seamless experience for both the barbershop and the customer.

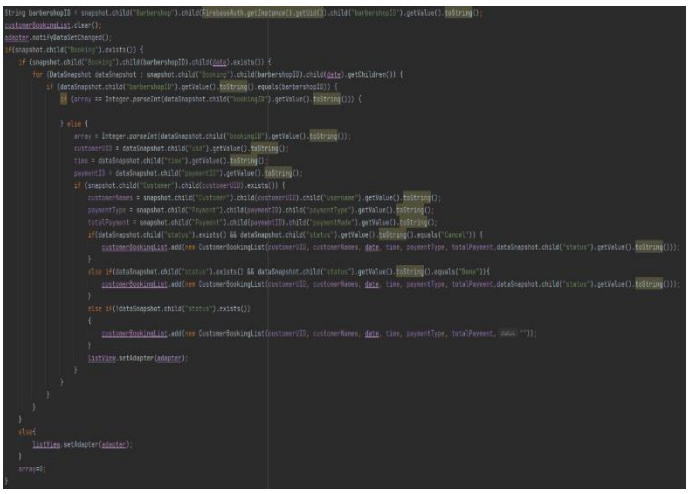

Figure 26: Manage Customer Code Segment

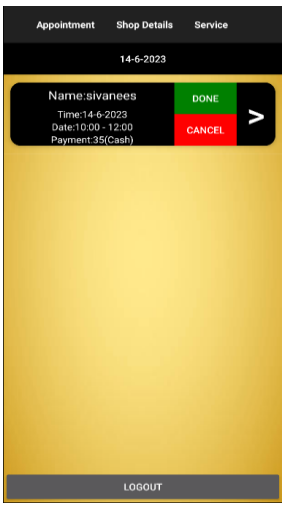

Figure 27: Manager Customer Interface

(v) Manage Location Module

The manage location module encompasses the coding implementation and user interface for updating and displaying barbershop information and location. In Figure 30, barbershops have the capability to update their information, including the location, through the provided user interface. Figure 28 showcases the implementation of storing the updated information in the Firebase Database, utilizing the `"setValue()"` function. Within the customer layout, Figure 29 illustrates the utilization of the `"shopState"` variable to determine and display barbershops in the same state as the user. This enables customers to view and connect with barbershops specific to their respective states. Additionally, Figure 31 demonstrates the process of retrieving the user current location. This information is leveraged to display nearby barbershops, as depicted in Figure 31, where only barbershops located in Johor state are shown. Overall, the manage location module offers enhanced functionality and convenience by allowing barbershops to update their information and showcasing relevant barbershops to customers based on their location and preferences.

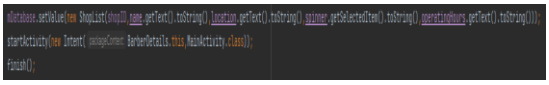

Figure 28: Location barbershops code segment

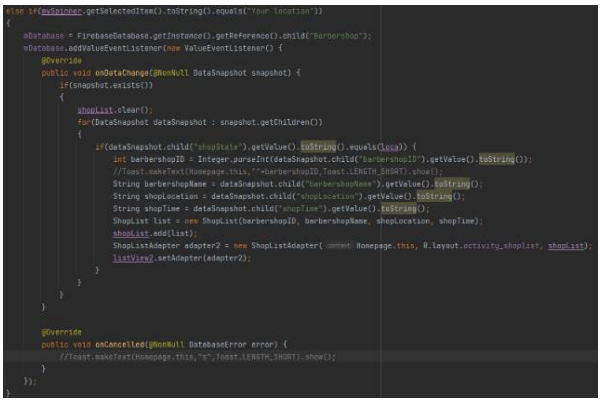

Figure 29: Access Location Customer code segment

Figure 30: Location barbershop interface

Figure 31: Location customer interface

(vi) Payment/ Refund Module

The payment/refund module involves the coding implementation and user interface related to payment processing and refunds. In Figure 32, the payment module's layout is loaded using "webView" and "`loadUrl()`" functions, which display the payment page of ToyyibPay, a payment gateway. The mobile application loads the specified Uniform Resource Locator (URL), allowing users to make payments through the payment gateway. Once the payment is completed successfully, a payment receipt is generated, and the booking is placed. If the payment fails, the application navigates back to the payment activity, allowing the user to retry the payment process. Figure 33 represents the user interface associated with the payment/refund module, providing a seamless experience for users to initiate and complete payments. The payment/refund module streamlines the payment process by integrating a secure and reliable payment gateway. Users can make payments for their bookings, ensuring a smooth transaction flow. In the event of payment failures, the system allows users to easily retry the payment process, ensuring a convenient user experience.

```
WebView webView = findViewById(R.id.webview);
webView.loadUrl("https://dev.toyyibpay.com/servicesbarber123");
webView.getSettings().setJavaScriptEnabled(true);

Button transFailed = (Button) findViewById(R.id.transactionfailed);
Button transSuccess = (Button) findViewById(R.id.transactionsuccess);


transFailed.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        startActivity(new Intent( packageContext: PaymentWebView.this,Payment.class));
    }
});

transSuccess.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        startActivity(new Intent( packageContext: PaymentWebView.this,BookingDetails.class));
    }
});
```

Figure 32: Payment Module Code Segment

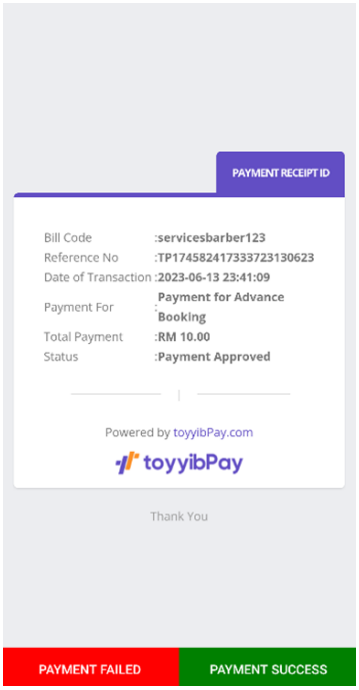

Figure 33: Payment Module Interface

(vii) Logout Module

The logout module encompasses the coding implementation and user interface related to logging out from the application. In Figure 34, the coding implementation is displayed, showcasing the function "`signOut()`" used when the customer or barbershop clicks on the logout button. This function is responsible for logging the user out of the application. Figure 35 represents the user interface component associated with the logout module. The logout button is in the main interface, and upon clicking it, the user is redirected to the "`MainActivity.class`", which is typically the starting point of the application. The logout module provides users with a straightforward way to log out from the application, ensuring the security and privacy of their accounts. By clicking the logout button, users are directed to the designated starting point, allowing for a seamless transition between sessions or user accounts.

```
startActivity(new Intent( packageContext: BarberAppointmentList.this,MainActivity.class));
FirebaseAuth.getInstance().signOut();
isThreadRunning = false;
finish();
```

Figure 34: Logout Code Segment

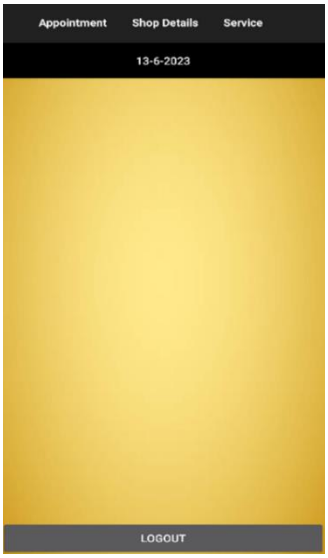

Figure 35: Logout Interface

4.2 User Testing

The Function Testing was conducted with a total of 23 test cases across different modules. The testing included registration, login, slot booking/cancellation, manage customers, manage location, payment/refund, and logout. All test cases were successfully passed, indicating that the application's main features are functioning as intended. The testing results are summarized in **Table 5.**

Table 5: Testing Results for Functional Testing

Summarization (Test Design for Barber Booking Mobile Application)		
Main Features	**Total Number of Test Cases**	**Pass / Fail**
Registration Module	3	Pass
Login Module	3	Pass
Slot Booking/Cancellation Module	4	Pass
Manage Customers Module	5	Pass
Manage Location Module	3	Pass
Payment/Refund Module	4	Pass
Logout	1	Pass

The User Acceptance Testing was conducted with 5 users to test both barbershop and customer module. The testing included registration, login, slot booking/cancellation, manage customers, manage location, payment/refund, and logout. The result of user acceptance testing was generated by the questionnaire given to the 5 users about the system. The results turn out to be strongly satisfactory mostly. The testing results are summarized in Figure 36.

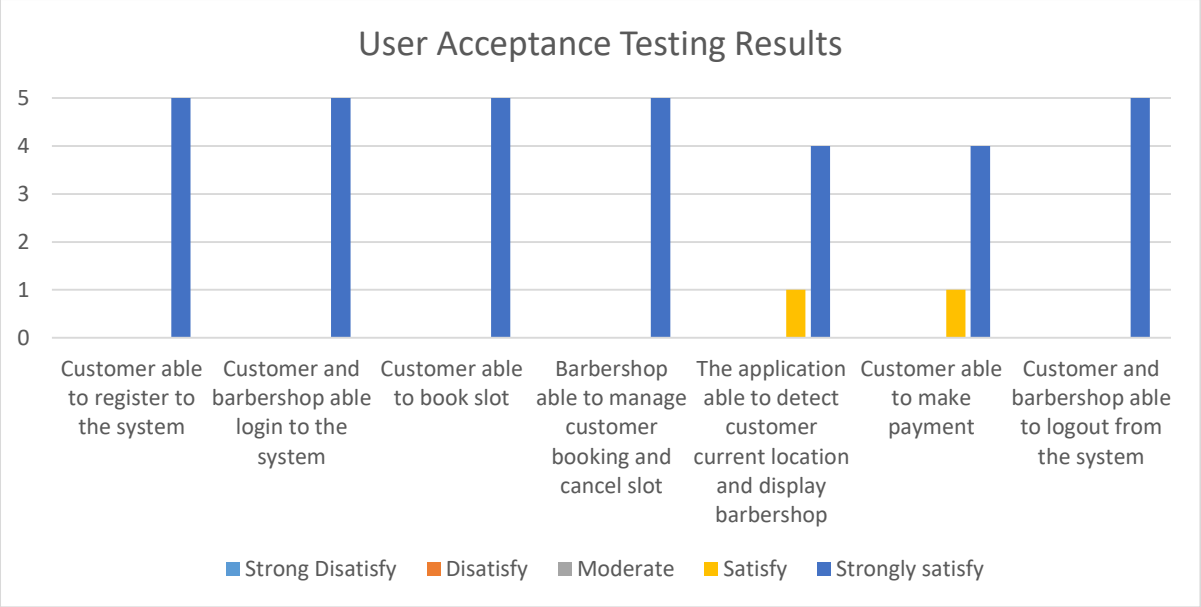

Figure 36: Testing Results for User Acceptance Testing

5. Conclusion

In conclusion, the application's creation and design have made it possible for barbershops and customers to use it to their advantage. Numerous modules are suggested as the solutions after examining management challenges with barbering services and investigating existing system. Two groups of app users—customers and barbers—are distinguished. According to the development plan, the barber booking mobile application has eight changes from the current manual process, including mobile based application, the customer registries will be kept digitally, the customers and barbers can a log in screen using into the application, an application updates slot booking or cancellation periodically, manage customer details and barbershop, a location searching function for customers, and lastly the display price interface that allow a customer to pay in advance. For better and more effective management of the barbering services, a notification and refunding procedures are also incorporated. Eventually, the Barber Booking Mobile Application is expected to have all the capabilities required to administer and run an online booking system effectively.

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] V. G. Childe, *The prehistory of European society*. London: Spokesman Books, 1962, pp. 184.

[2] Monkey Barbershop, "About" *Facebook*. [Online]. Available: https://www.facebook.com/people/Monkey-Barber-Shop/100043646186123/. [Accessed: 10-Jan-2023].

[3] J. K. Makumbiro, "An evaluation Of The Efficiency Of Queuing Management Systems", 2020.

[4] A. Rashima, S. Janani, S. Dhanalakshmi, R. Krishnakumar, and P. P. Santanam, "Visual demand, visual ability and vision standards for hairdressers – an observational study from Chennai, Tamil nadu," Indian Journal of Ophthalmology, vol. 69, no. 6, pp. 1369, 2021.

[5] C. Li, J. Bao, and H. Wang, "Optimizing low memory killers for mobile devices using Reinforcement Learning," *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, missing location, 2017.

[6] V. Ratten and A.J. Thompson, "Digital sport marketing," *Developing Digital Marketing*, pp. 75–86, 2021.

[7] D. J. Langley, J. van Doorn, I. C. L. Ng, S. Stieglitz, A. Lazovik, and A. Boonstra, "The internet of everything: Smart things and their impact on business models," Journal of Business Research, vol. 122, pp. 853–863, 2021.

[8] J. Loonam, S. Eaves, V. Kumar, and G. Parry, "Towards digital transformation: Lessons learned from traditional organizations," Strategic Change, vol. 27, no. 2, pp. 101–109, 2018.

[9] C. Gao, A. Gutierrez, M. Rajan, R. G. Dreslinski, T. Mudge, and C.-J. Wu, "A study of mobile device utilization," *2015 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, Missing location, 2015.

[10] U. Arshad, Farhat-ul-Ain, J. Gauntlett, N. Husain, N. Chaudhry, and P. J. Taylor, "A systematic review of the evidence supporting Mobile- and internet-based psychological interventions for self-harm," Suicide and Life-Threatening Behavior, vol. 50, no. 1, pp. 151–179, 2019.

[11] A. Ademuson, A. O. Rebecca, and O. Akanle, "Men in women industrial space: Male hairdressers of Ibadan, Nigeria," Journal of Asian and African Studies, p. 002190962210996, 2022.

[12] BookedIn "Online appointment scheduling," *Bookedin*, 30-Aug-2022. [Online]. Available: https://bookedin.com/. [Accessed: 1-Jan-2023].

[13] The Barber's Room "Mens premier grooming salon in Old Welwyn - Barber's room." [Online]. Available: https://www.barbersroom.com/. [Accessed: 1-Jan-2023].

[14] JustBookSalon Pte Ltd "Just Book Salon," [Online]. Available: https://www.justbooksalon.com/. [Accessed: 1-Jan-2023].

[15] C. Larman and B. Vodde, *Large-scale scrum: More with less*. Boston: Addison-Wesley, 2017.

[16] J. D. Summers, S. Joshi, and B. Morkos, "Requirements evolution: Relating functional and non-functional requirement change on student project success," *Volume 3: 16th International Conference on Advanced Vehicle Technologies; 11th International Conference on Design Education; 7th Frontiers in Biomedical Devices*, 2014.

[17] S. Alsaleh and H. Haron, "The most important functional and non-functional requirements of knowledge sharing system at public academic institutions: A case study," Lecture Notes on Software Engineering, vol. 4, no. 2, pp. 157–161, 2016.

[18] M. Maguire and N. Bevan, "User Requirements Analysis," IFIP Advances in Information and Communication Technology, pp. 133–148, 2002.

[19] S. Al-Fedaghi, "Diagramming the Class Diagram: Toward a Unified Modeling Methodology," International Journal of Computer Science and Information Security, 2017.