BILLHUB – CONSOLIDATED BILLS MANAGEMENT SYSTEM

CHEAH WEI HENG

UNIVERSITI TUNKU ABDUL RAHMAN

BILLHUB – CONSOLIDATED BILLS MANAGEMENT SYSTEM

Cheah Wei Heng

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Science (Honours) Software Engineering

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

October 2024

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature :

Name : Cheah Wei Heng

ID No. : 2005769

Date : 28/09/2024

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"BillHub – Consolidated Bills Management System"** was prepared by **Cheah Wei Heng** has met the required standard for submission in partial fulfilment of the requirements for the award of BACHELOR OF SCIENCE (HONOURS) SOFTWARE ENGINEERING at Universiti Tunku Abdul Rahman.

Approved by,

Signature : ________________

Supervisor : Beh Hooi Ching

Date : 28 September 2024

Signature : ________________

Co-Supervisor : ________________

Date : ________________

ACKNOWLEDGEMENT

I would like to thank my supervisor, Ms Beh Hooi Ching and my moderator, Dr Chia Kai Lin for their unwavering support and insightful guidance throughout the development of this project. Their valuable advice and encouragement have been meaningful to me.

Additionally, I would like to express my appreciation to my former employer, Digi, for providing me with opportunities to understand the challenges associated with bill payments in the current market. This experience significantly contributed to shaping the concept of my project.

ABSTRACT

With the growth of the internet and technology, consumers are increasingly using subscription services. Many billing companies are promoting paperless and electronic billing to protect the environment. Consequently, online bill payments have become a necessity for bill payers. To encourage e-billing, each company has developed its own application for managing and paying bills. However, as the number of subscriptions rises, users have to use more and more applications to handle their growing number of bills, making the process tedious and repetitive. This increase in bills and the shift to paperless statements also leads to other issues such as missed payments, tracking errors, and financial management problems. To solve these problems, a consolidated billing system, BillHub was developed to help users manage and pay all their bills under a single application. The system includes features such as bill registration, the ability to pay multiple bills in one transaction, bill history, bill analysis, reminders, and auto-billing. The project was implemented using the React Native framework, with a Node.js server connecting to MongoDB. The server facilitates communication between bill payers and billing companies. The development followed an incremental model, focusing on one module at a time, starting with the most crucial features. After development, API testing ensured smooth communication between users, billing companies and server. Usability and user acceptance testing were conducted, with BillHub achieving a SUS score of 92.2, indicating high ease of use. Although the application includes multiple functionalities, it has not yet established business connections with any billing companies. Further promotion and collaboration with billing companies are needed for market deployment. Overall, the project is considered a success as it meets its objectives and scope.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

BMS	Bill Management System
SSBPTs	Self Service Bill Payment Technologies
EBPP	Electronic bill presentment and payment
ABP	Automatic bill payment
TnG	Touch ‘n Go
UI	User Interface

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 General Introduction

People start to pay bills once they start using services or products that require payment on a recurring basis. This could include utility bills such as electricity, water, and gas, as well as subscription services like internet or streaming platforms. Utilities such as electricity and water are basic living requirements for every household, while Wi-Fi has also become necessary as both studies and work require a Wi-Fi connection. This indicates that people have been managing bills since long time ago before the digital era. Traditionally, people paid bills physically at the branch or centre of the billing companies. The difference of bill payment in this era is that bill payers have the extra option of paying bills digitally through various platforms. Despite, the traditional method remains as an option for those who prefer it. In 2020, 62% bills of the 15.5 billion bills were paid digitally last year (Jonathan Berger, 2022).

In today’s fast-paced digital era, managing bills are the non-avoidable tasks for many individuals. Most of the people now are enjoying services provided by utilities companies, telco companies, streaming companies and insurance companies. It is just either the users are paying bill for himself or have someone paying the bill for them. Thus, bill payment has become a necessary to do thing in our life. From paying bills to tracking bills changes, people are trying not to make any error or oversight. While people are paying bills, they tend to manage the bills to have a better control over their expenses. This becomes challenging to users especially when they have to deal with multiple bills from various service providers, each with its own due dates, payment methods and billing cycles.

Today’s bill payers want a frictionless bill payment experience. However, the third-party websites and applications have historically become better at meeting bill payers’ expectations. Investigation has shown direct billers capture 76 percent of all online bill payments, compared to 22 percent from financial institutions and 2 percent for fintech (Jonathan Berger, 2022). Direct billers are still the main choice of online bill payments while 48 percent of the bills are still paid physically in year 2020 (Jonathan Berger, 2022).

The aim of this project is to address and thus, solve the problem and challenges associated with bill management to ease the overall process and experience of bill management through the development of an integrated bill management system. This system aims to streamline the bill payment process, enable consumer to pay multiple bills by only using one platform. Besides, this system also aims to enhance user convenience by provide valuable insights into consumer's billing history and spending patterns.

In this report, research and literature review are carried out to understanding the needs of the system. Besides, requirements elicitation will also be carried out through survey and interview to understand collect the functional and non-functional requirements needed for the system. Throughout the investigation, common issues experienced by users are explored, such as difficulty of remembering multiply due dates, difficulties to check outstanding amount and managing multiple payment methods. All the issues found are described in detail under the problem statement section of this report. Additionally, reviews of similar applications are carried out, the limitations and advantages of the existing bill payment platforms are mentioned under the similar application review section.

Furthermore, the objectives of the project are outlined and provide an overview of the proposed solution, detailing its key features and functionalities. Through the rigorous research, analysis and development, a robust and intuitive bill management system that meets the needs of modern users are developed.

In conclusion, this project seeks to contribute to the ongoing evolution of financial technology on bill management by offering a practical solution to the challenges. By leveraging user-centred design principles, the integrated bill management system developed will not only simplifies the bill payment process but also fosters greater financial understanding and control of the user.

1.2 Significance of the project

The important of this study is to find the potential to improve the overall bill payment experience. This study is important to find the main challenges faced by bill payers in managing their bills using the current bill payment system. By developing a comprehensive BMS, this project aims to provide a better solution of bill payments.

Currently, bill payers encounter difficulties in keeping track of multiple bills from various service providers, causing missed payments, late payments and service disruption. Besides, complexity of payment methods, tediousness of swapping between multiple applications and complexity of managing bills with different billing dates are truly frustrating, especially for bill payers with busy lifestyles or limited digital payment experience. Moreover, the lack of centralized system in billing system exacerbates the problem, causing it a stress to bill payers as they must use multiple applications for bill payment and management.

This project will explore innovative solutions to consolidate bills and ease the bill payment processes. Literature review, application comparison, users survey, and practical implementation are carried out to gained valuable insights to understand user's needs and preferences. These efforts will make the development of solutions that meet user requirements effectively possible.

In short, this project is important as it addresses the current challenges faced in bill payment and bill management processes. Besides, this project also offers technical solutions to improve the overall bill payment and management experience by development of bills consolidated BMS.

1.3 Problem Statements

Throughout the literature review and the survey conducted. There are few problems encountered with the current available bill payment solution. However, this project will only focus on the major problems as listed here:

1.3.1 Difficulty in remembering due dates.

From the analysis result of questionnaire at section 4.2.1.2, 82.4% of the respondents find it difficult to remember due dates for multiple bills. Bill payers face difficulty in tracking multiple bills with different due dates and billing frequency. This will lead to missed payments, late fees and even barred services. The main reason of this problem is different billing dates, irregular billing cycles and lack of reminders from the biller companies. As result, it causes stress and negative impact toward credit scores. This problem is worsened on older individuals, with diminished memory abilities.

1.3.2 Complexity of payment steps.

There are several payment platform and methods available nowadays. For example, online banking application, application provided by the biller companies, E-Wallet payment and third-party application which support bill payment such as Shopee. Different platform required different payment methods and steps, which are difficult for novice user. One of the reasons for low adoption of online payment is due to the complexity of various electronic payment methods (Laukkanen and Lauronen, 2005). Example of the steps are key in account number, inputting sensitive financial information and verifying transaction details. This complexity increases the risk of errors and frustrations of the users. Interfaces which are not user-friendly are also one of the causes to this problem.

1.3.3 Tracking fluctuating billing amounts.

Bill payers struggle with budgeting for bills due to fluctuating amounts of bills. This problem will lead to financial stress and uncertainty. This is because bills such as utilities, credit cards and certain phone plans are different each month based on usage. However, these bills are still manageable by the users. Bill such as insurance is different, it is uncontrollable and subject to factors such as rising medical costs, pandemic and natural disaster (Yao and Li, 2016). This unpredictability of billing amounts will disrupt the financial planning and cash flow of bill payers. Without understanding the fluctuation patterns of the billing amounts, bill payers are unprepared for the changes and unable to manage their finances effectively.

1.3.4 Limited access of billing information.

Bill payers are struggling to access their bills and payment information when they are paying bills with third-party applications such as E-Wallet, online banking applications and e-Commerce platform like Shopee. The nature of the billing systems makes it difficult for bill payers to consolidate all their billing information in one place, causing it difficult to access to bill payment required information such as account numbers and outstanding amounts. Instead, bill payer must navigate to individual company applications to check their outstanding amount and download the bill statements for further review.

1.3.5 Tediousness of paying multiple bills.

Based on the survey analysis on section 4.2.1.2, there are 85.3% of the respondents find it tedious to pay multiple bills. Paying multiple bills through multiple platforms and manually entering the payment details can be tedious and time-consuming for bill payers. Bill payers who wish to pay bills with application provided by the billing companies must switch between multiple applications, as each company has their own application. Furthermore, bill payers have to make a transaction for every single bill they are paying, leading to repetitive transactions. On the other hand, bill payer who wish to make payments through third-party applications must switch between the third-party application and the application provided by the billing companies to check and retrieve the account number and outstanding amount. In short, bill payers must make multiple transactions and switch between multiple applications to check the billing information. The lack of a centralized system will lead to tediousness of paying multiple bills and result in poor user experience.

1.4 Project Goal and Objectives

1.4.1 Project Goal

Throughout the literature review, a few problems have been found with the current available bill payment systems. Thus, this project aims to solve the problems mentioned in section 1.3 by developing a consolidated bill payment and management system by September 2024.

1.4.2 Project Objectives

1. To examine existing bill presentment and payment process by reviewing similar applications in bill payment and management system.
2. To create BillHub as a Consolidated Bill Management System (BMS) that is user-friendly and simplifies the process of managing multiple bills from various service providers.
3. To evaluate the BillHub using a usability test and user acceptance test.

1.5 Project Solution

The BillHub is a BMS (a centre party) that will communicate with two parties, bill payers and billing companies. The system consists of front-end user interface, backend processing of bills and payment and database to store the necessary bills related data. Users can register accounts and register their bills. After registration, the system will store the accounts information of the bills in database. At every database changes, the updated bills' details are retrieved from the billing companies through the backend request sent by the billing companies. Once the bills are retrieved, it is processed and prompted to users' interface and processed for automated services. Subsequently, application's front-end will show user's the necessary information of their registered bills that are needed for bills payment. Users may interact with the application front-end to select the desire functionality or make payments. During payment, system will navigate users to payment gateway, once users completed the transaction, backend of the system will process the payment and pay the bills to different billing companies respectively, a receipt will be generated for the transaction. After that, the system will record the history of payment made in database. The changes of payment records will then trigger the bills analysis module to update the latest bill analysis module. User may view the result of analysis by navigating to the page using the mobile application. In addition, users may also set up reminder of the bills, which the backend server of the BMS will hold and send notification upon the expiry date of the bills. Besides, users may also set up auto bill payment functionality using the mobile application. Once set up, the backend of the system will retrieve the bills and ready to make payments automatically to the bills that users registered on the predefined dates. Figures 1.1 and 1.2 will visualise the overall flows of the system design.

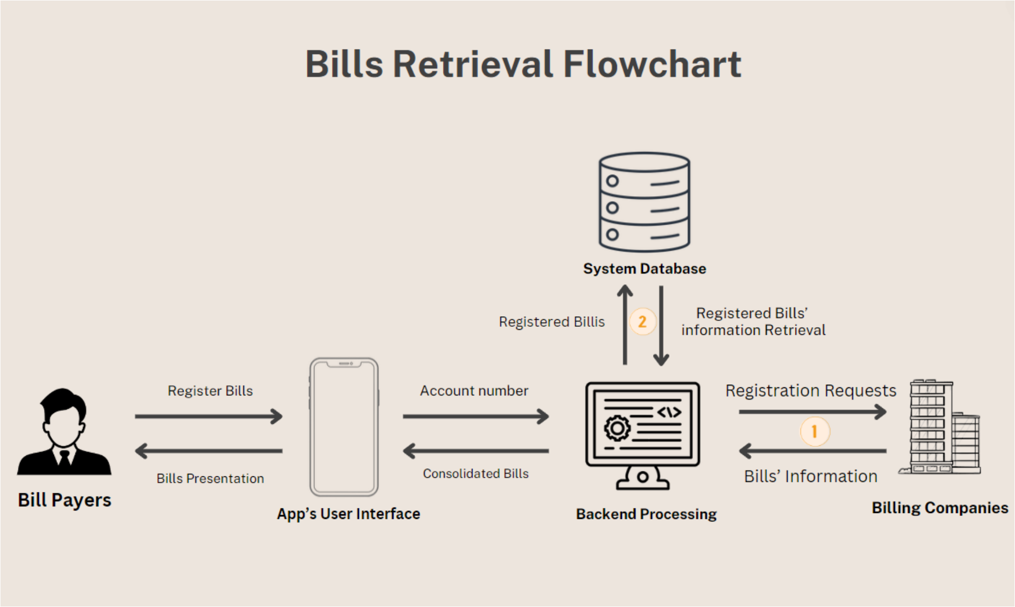


Figure 1.1 Bills Retrieval Flowchart

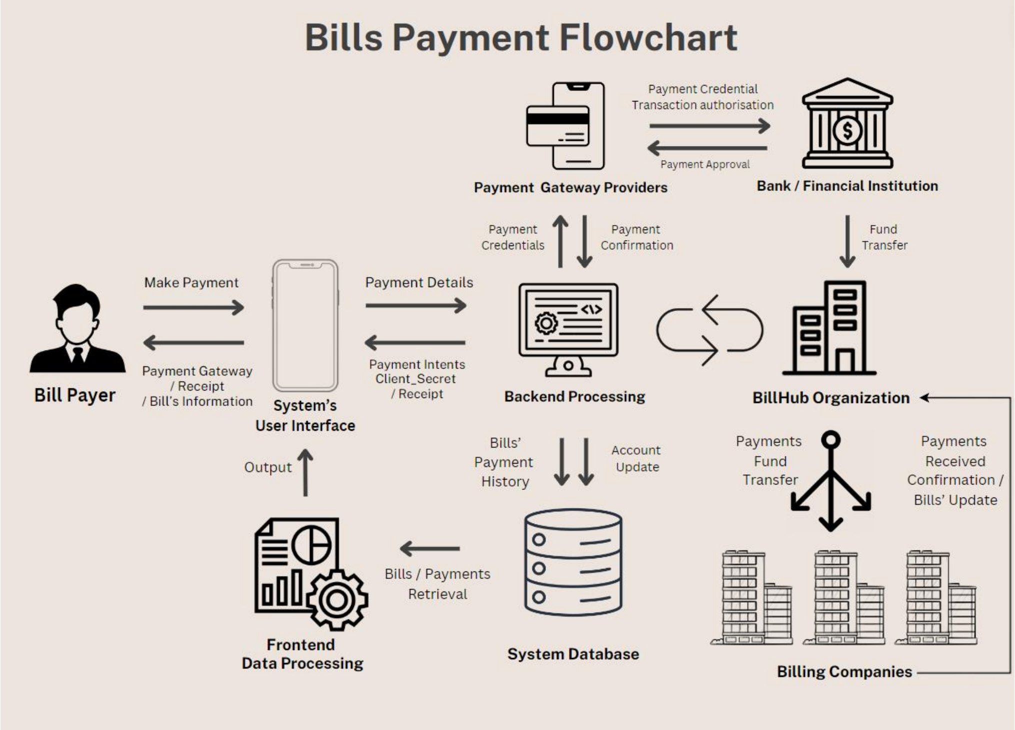


Figure 1.2 Bills Payment Flowchart

1.6 Project Approach

1.6.1 Research Approach

The research approach selected for this project is quantitative approach, with data being collected through google form survey. The data is analysed to generate numerical data and statistics, which can be readily understood through graphical visualisation generated by Google. With such abundant number of samples, we can ensure that the project will fulfil the needs of many bill payers. While quantitative approaches are generally dependable since there are a wide number of users involved, there is a potential risk of bias if the survey is conducted only within 1 category of potential user. Thus, users from different gender and different age groups are approached to collect a meaningful non-bias data. Google form questionnaires are utilized to collect relevant data, opinions and experiences from bill payers and non-bill payer. From the analysis of the questionnaires, the problems faced by bill payers are identified and addressed as problem statements, guiding the drafting of scope of this project. This is to ensure that the BMS developed can fulfil the user's demands and effectively manage the existing challenges encountered by bill payers.

However, the quantitative approach alone may not be sufficient to gather all the necessary information. Among older individuals, there are individuals who may still be responsible for bill payments but are less familiar with and less willing to respond to Google Form surveys. Therefore, to complement the quantitative data collection, a qualitative approach is also employed. Interviews are conducted to gather requirements directly from these individuals, ensuring a more comprehensive understanding of their needs and preferences. Besides, not all the problem faced by bill payers can be addressed through quantitative approach. During the interview, multiple open-ended questions will guide bill payers to share and describe the problem they are facing with the bill payment processes they currently experience. Furthermore, the interviewer will also encourage the interviewees to elaborate more on their pain point regarding bill payment and provide possible suggestions for improving the BMS.

1.6.2 Development Approach

The development approach selected for this project is the incremental model, which is a methodology within the software development lifecycle (SDLC) that divides requirements into multiple modules. One of the key advantages of the incremental model is its ability to deliver value quickly. During the first iteration, the primary module can be developed, with subsequent iterations focusing on the next important module and so on. The incremental model is able to handle the complexity of this project, which is moderate to high, where the requirements are well-understood and subject to improvement. Furthermore, the incremental model also allows for flexibility in handling requirements. Throughout the development of the system, changes can be accommodated to enhance the system. When there is a need to implement new features or improve the system, the incremental model can support these changes more effectively than other approaches. In addition, the incremental model is also able to detect errors in the early stage of the development process. Since software is built incrementally with multiple iterations, errors are easily identified during the testing phase of each iteration. This iterative approach continues until the final system is delivered. (Sommerville et al., 2011).

1.7 Scope

1.7.1 Target Users

In this project, there are two categories of target users, which are bill payers and administrators from billing companies. Bill payers are able to pay and manage their bills using the system while billing companies should provide the bills information and details for payment.

1.7.1.1 Bill Payers

The first target users of this project are mass-market bill payers, from individuals currently paying bills to individual who might be paying bills in future. Individuals who are managing household expenses are the main and major group of users who are in charge of bills payment. This category is usually working adult who have the financial ability to pay various bills such as utilities (electricity, water, gas), telecommunications (internet, mobile phone), insurance premiums, and other recurring expenses such as entertainment subscription. However, it is important to note that working adult might not necessarily the person who in charge of bill payment directly, possibly due to the limited knowledge and familiarity of digital payment technologies. Thus, they may pass the responsibility to their children who are familiar with online/digital payment to pay the bill, while they are still the financial contributors. On the other hand, elderly individuals and retires are also considered as the target users, especially on those who are still continuing to manage bills on their own. The availability of bills payment methods and platforms are various in every country. This project will focus on the bill payers in Malaysia by studying their bill payment behaviour and the available platforms utilized in Malaysia.

1.7.1.2 Administrators from Billing Companies

In the proposed system, the second user is the administrators from billing companies. The administrators will in charge of facilitating the seamless information exchange between the bill payers and the company. These administrators are tasked with several key responsibilities, one if it will be authenticating user bill registrations. When user registers a bill of a company under their account, the administrator of the company has to make sure the verification code is sent to the user for authentication. After registration, the administrator should also make sure the bill information and outstanding amount are always updated to the bill payer through the backend API communication with the BillHub's server. Furthermore, during the received of bill payment, the administrator has to make sure that the outstanding amount of the registered bill is reflected and updated as soon as possible. Administrators play an important role in the system as they serve as the primary contact point between the company and its customers (bill payers) in the system.

1.7.2 Modules

1.7.2.1 Bills Registration Module

Users can log their bills in this module. The user has to select the billing companies and input the primary identity that linked to their account such as account number, phone number, or subscription ID of the services he/she subscripted. Once the bill is registered into the system, the account details are stored in the database until the user removes it manually. Thus, user doesn’t have to key in the same credential information manually every time they pay bills. Additionally, every bill is registered with a unique bill ID as primary key for database management. This allows the system to track and analyse each bill. Besides, users also allowed to rename the bills with a name that are easier to recognised such as “Grandpa’s House Electric Bill”.

1.7.2.2 Bills Retrieval and Consolidation Module

Once users registered the bills under the system, the system’s backend server will automatically retrieve updates of the billing amount. After the retrieval of latest outstanding amount, the system will consolidate the bills into a single view, allowing users to see all their outstanding bills at once. To ease the understanding and visualisation, the outstanding amount, overdue amount, due date and number of days before due are shown. In addition, users can also filter the bills by the categories of the bills such as “Electricity”, “Water Bill”, “Entertainment” and more. Furthermore, user can also sort the arrangement of the registered bills either by sequence of registration or bills closer to its due date. Besides, users can view the detail information of a single bill by clicking the bill.

1.7.2.3 Bills Payment Module

Inside the bill payment page, users can then select the bills that they wanted to pay by simply ticking checkbox beside the bills, the system will automatically calculate the total amount of the selected bills. In addition, user has freedom to modify the amount that they want to pay through the “Edit” button, users can either pay lesser than outstanding amount, or pay extra in advanced. To ease the decision-making, the deadline of the bill and the number of days before due will also be shown. After the confirmation of chosen bills, the system will generate an invoice that consolidates the selected bills, allowing users to pay all the bills in a single transaction. A receipt will be shown after the complete of payment.

1.7.2.4 Bills Tracking and Analysis Module

The system will analyse the amounts of every bill and sum up the total amount billed and paid. Pie charts will be generated to visualise the total cost and cost proportion of the bills issued and payments made. Besides, the system will also track on the changes and fluctuation of the total bills issued and payment made every month. In this case, a line chart will be generated for users to visualise and understand the changes of the bills. These analyses help user to understand the bills issued and the payment made to a particular company.

1.7.2.5 Auto Billing Module

Users are able to set auto billing to selected bills. Auto billing module will help user to pay the bills on the day of bill releases or the bill's deadline according to the user preference. Thus, user doesn't have to manually pay the bills.

1.7.2.6 Bill Reminder Module

When this reminder module is activated, the system will monitor the payment deadlines of registered bills. If any bills remain unpaid and are approaching their expiration, the system will send a reminder notification to the user via the application.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In today's fast-paced society, every process is trying to achieve better convenience and higher efficiency, including bill payments. Bill payment process has evolved from delivering paper bill and making physical cash payment at counter up to digital online payment and auto-payment today. However, these bill payments methods are still not the optimal way to manage the bills. The difficulties faced by bill payers with online bill payment applications are such as difficulty in using bill payment application (Ray *et al.*, 2023), high complexity of payment steps (Staykova and Damsgaard, 2016) and tediousness of switching between multiple applications. The details of these problem are described clearly in section 2.2. On the other hand, auto billing seems like a perfect solution to these issues. However, it also has hidden negative effect. Throughout the study of section 2.2.2, it is shown that the application of auto billing increases the residential electricity consumption by 4.0% and commercial electricity consumption by 8.1% due to the salience effects. As users no longer paying the bill manually, users tend to forget their utilities bills are directly proportional to their usage, and thus lead to a new problem: energy wastage. Since each of the latest solution mentioned above consists of problems, this project was intended to develop a system that possibly remove the problems with a new bills payment solution.

In this chapter, research is conducted on the history of bill payments to understand problem encountered and solution invented. Additionally, this research also identifies new problems that arisen and seek for solution of the problems. Furthermore, this project will focus on the bill payment platforms available in Malaysia. The comparison between the platforms and payment steps will help to figure out the advantages and disadvantages of the applications. In addition, surveys and interviews are carried out to collect user's data and feedback. The collected data will then be analysed to generate useful information such as the requirements document. Thus, by enhancing and applying the advantages on the multiple applications, and eliminate the disadvantages, the bill management system developed are able to satisfy the users. Besides, the development methodologies and testing methods will also be analysed to determine the most suitable methodologies and method to be selected for the development process.

2.2 History and Evolution of Bill Payment

In the past 20 years, billing companies are focusing on developing Self Service Bill Payment Technologies (SSBPTs) such as bill payment websites or mobile applications (Jayasimha and Nargundkar, 2006). Bill payers begin to pay bills electronically due a few factors such as controlling their personal preferences, record keeping, the convenience of digital payment, the incentives provided by the billing companies, the personal involvement of bill payment and protect the privacy of billing information (Mantel 2021). As a result, bill payers can pay bills respectively using the application provided by the billing company itself. For example, AirSelangor, AIA+, myTNB and MyUMobile application. However, research has shown that Generation X bill payers are not familiar with technology trends and often struggle when using bill payment mobile applications (Ray *et al.*, 2023). While smartphones and mobile payment solutions have being implemented, many bill payers still preferring traditional payment methods like cards and cash at the physical store, most mobile payment apps struggle to attract users, mainly due to the complexity of mobile payment (Staykova and Damsgaard, 2016). Through the Reach and Range framework, it shows that it is really important for these companies to make sure they have the right balance of people using the platform and the services they offer. As stated in the report, the bill payment applications should be converted to multi-sided platforms that facilitate interactions and transactions between multiple groups of participants, including the billing companies and bill payers in order to gain mass quantity of users and to simplify the overall online bill payments experiences (Staykova and Damsgaard, 2016).

2.2.1 Traditional Bill Payment vs Digital Bill Payment

2.2.1.1 Traditional paper bill presentment and payment

Traditional bill presentment is a process of using physical paper to show the bill payers with a paper bill for the goods or services they subscribed to, and the bill payers will often pay via cheque. The bill presentment process comprises various tasks in billing operation, such as generating, printing, mailing and delivering the bills to the user. The expenses used for these processes vary between $0.70 to $1.50 (PayAnyBill, 2000). In other words, billing companies spend between 10.5 billion and 25.5 billion each year to present their bills.

Additionally, this process includes the generating of periodic report from its billings systems, followed by new notification of the bills to their customers. However, this type of physical billing is not a cost nor time friendly process, as it requires the printing of account statements, packing them into envelopes, and organising them before mailing and delivery. It may take 1 to 3 days to generate and process the bills, depending on the degree of automation (Doculabs, 1999). In the traditional bill payment world, notification and presentment of a bill occur simultaneously when the bill payer receives the bill in their mailbox. For most bills today, the post office plays a role as a notifier and presenter by delivering each of the bills from billing companies to bill payers. Averagely, the bill takes between 3 to 5 days to reach to the target bill payer.

Before the digital era, most of the billing companies have streamlined and improved their paper bills processes as much as possible. However, this conventional paper billing method, still has obstacles like inconvenience, time-consuming, cost and reliability. Paper bills restrict consumers to access their bills anytime and anywhere. Furthermore, it is less error acceptance, a wrong customer's address or overlook of billing mail may lead to significantly longer time of bill payments. The overall process is still time-consuming and costly to the billing companies, it also lacks reliability because paper bills have no guaranteed delivery. Furthermore, failure to receipt the paper bills will cause more time, cost, late fees payment, and bad bill payment experience for both parties.

2.2.1.2 Modern paper bill presentment and payment

Modern bill presentment does not use the mailing system as a means of delivering bill presentment and payment notifications. Rather, the Internet is utilised as a higher speed and lower cost medium to present the bills electronically through platforms such as billing companies' self-developed application, Gmail and Hotmail. Two of the main electronic bill presentment and payment (EBPP) models are the biller direct model and the consolidation model. There are a few variations of the consolidation model, which includes personal financial management software, screen scraping, and scan and pay methods. Both models involve billing companies notify their customers or bill payer subsequently login to the consolidator application to check the details of their outstanding bills.

2.2.1.2.1Biller-direct model

As for the biller-direct model, the billing company will generate a digital bill for the customer that has enrolled for the EBPP services. Next, the billing company will send an email to the consumer to notify them of an outstanding/pending bill. After receiving the e-mail, the bill payer has to log into respective biller's website or application separately, where the digital billing statement are presented online. Upon viewing the digital bill, the consumer can initiate payment directly from the website or application. The biller (billing company) is responsible for providing a user interface to allow customers from notifying, accessing, and paying the bill. Even though the initial development fee will be higher, it is estimated that the bill statement production cost in term of money and time would be significantly reduced when using digitally generated bills in long run. This offers a more reliable and efficient delivery mechanism for both bill delivery and payment. The advantage of biller-direct model for the billing companies is that it maintains the direct connections with their customers. Since the billing companies can control the visualisation of their website, it can control its brand identity. Furthermore, since the billing companies relies on its own payment system, it eliminates the risk of third-party integration problems that are often found in other EBPP models. Looking from the customer's perspective, the biller-direct model meets most of the expectations from users from traditional billing system. Consequently, customers may find it more comfortable to establish EBPP directly with billing companies that they trust rather than a new and random third-party company. Utilising the Internet as stable delivery medium also results in lesser number of mails lost, delayed payment and misplaced payments.

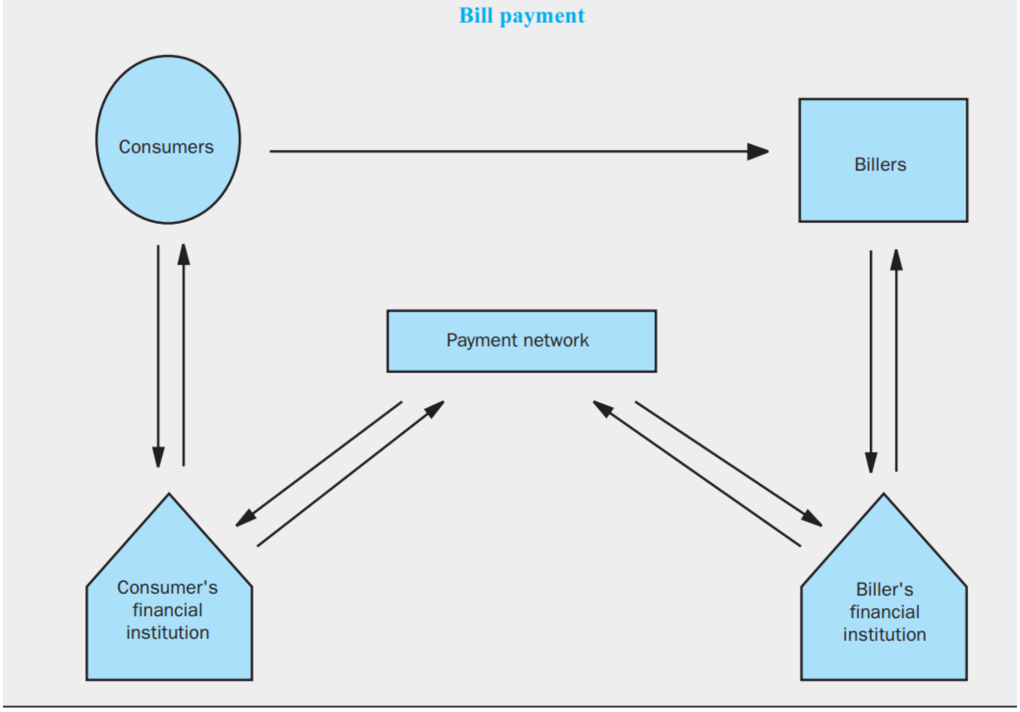

Bill payment
Consumers
Billers
Payment network
Consumer's financial institution
Biller's financial institution

Figure 2.1: Five parties involved in modern digital bill payment process. Alexandria Andreeff et al. (2001)

The modern digital bill payment process involves more than five parties, which are the bill payer, the bill payer's banking institution, billing company, its financial institution, and the payment network. The relationship of the parties is shown in figure 2.1 above. The bill payer will use the funds stored in their banking's institution to make payment. For cheque payments, the bill payer must first deposit the funds into their account before they can make the payment. The bill payer's banking institution will transfer the funds via the network to the financial institution of the billing companies, provided if the funds of the bill payer is sufficient. Besides, the biller-direct model would be beneficial to the consumers for its ability to access latest or real-time information since billing companies can update the customer data frequently.

However, biller-direct model also comes with its disadvantages, the biggest con is the consumers are required to be proactive in visiting multiple billing companies’ websites or mobile application to review and pay their bills. Even if there are billing companies that use the same billing service provider (BSP) will not solve this inconvenience as the billing information cannot be retrieved from one single site. This can again be time-consuming and might cause higher complexity of bill payment. If all the billing companies are not using the similar processes or user interfaces for presenting bills and receiving payments, users may be confused throughout the processes. In addition, digital bill payment can be challenging for some users as they may have a hard time remembering multiple usernames and passwords when paying their bills. To make things worse, not all the billing companies offer a diverse range of payment methods, this indirectly forces the bill payer to learn and use different payment methods for different bills. In short, the biller-direct model was acceptable to the early adopters, but it may be a hassle in today's context, especially to users who were expecting a better system with higher efficiency and convenience.

2.2.1.2.2 Consolidation Model

Alternatively, there is the consolidation/ aggregation model. This model was created to meet the needs of bill payers who wants to only use one single application/platform for accessing and paying their bills while minimizing the potential expenses. The billing companies sends the customer’s billing statement and information to a third-party company who control the consolidation website/application, the consolidator will then present the bills from multiple companies to the respective users. During the presentation, the consolidator will collects and merges the data from multiple billing companies and consolidates the information at a single module of their website/application.

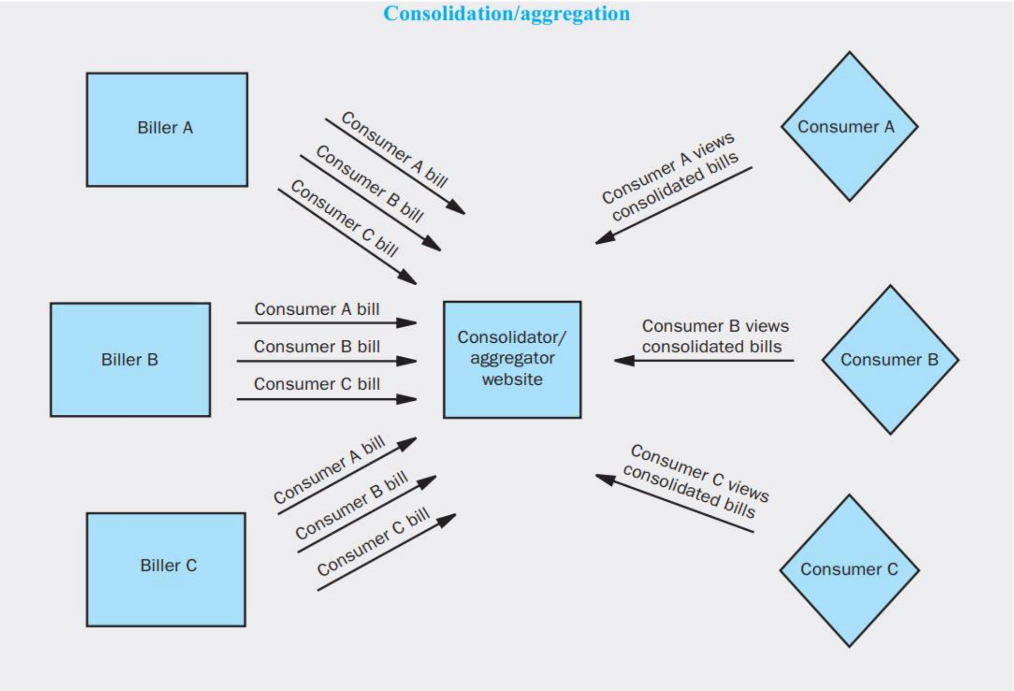


Figure 2.2: Consolidation Model (Adopted from Alexandria Andreeff et al. (2001))

There are two types of consolidation, thick consolidation model and thin consolidation model. The consolidator will present both the details billing statement and summary of the bill payer's billing information. Therefore, users of the thick consolidation model are not required to login into the websites/ application provided by the billing company in order to view the detailed billing information. The advantage of thick consolidation model is the billing company can provide EBPP services to the bill payers with lower cost. Thus, the billing companies with lower financial ability can still enter the EBPP platform. From a bill payer view, having a third-party EBPP services becomes an advantage if they able to provide various digital payment options that are not supported directly by the billing companies themselves. However, thick consolidation model also comes with disadvantage, billing companies are generally losing their branding, marketing and cross-selling opportunities as consolidators eliminate the direct contact between the billing companies and their customers. From a view of bill payer, third-party consolidation/aggregation model provider usually charge a processing fee, this is a main cause that 59% of the bill payers are not willing to pay bill online, and only 6 percent are willing to pay more than 5% for the services (Kerr and Litan, 2000). Besides, quality of customer service will also reduce because customers are dealing with third-party consolidation provider for some of the problems.

On the other hand, thin consolidation model will not hold the details of the billing information. In short, bill payers can only view the summary of their bills while using the consolidator's site or application. If the bill payers wish to view the details of their bills such as bill statement, they have to use the visit the original billing companies' website or application. One of the benefits of this thin consolidation model is that it allows billing companies to provide online customer service, cross-market products and gain better control of their own e-billing processes by holding the control over the consumer data at their own platform (Kerr, 2000). In additional, the thin consolidation model provides higher data security as their personal information and billing details are only stored by the billing companies and are not shared with a third-party company. However, this model also comes with disadvantages, the billing companies still lose the marketing opportunity if the bill payers do not choose to view the details of their bills. Besides, user will also find it not convenient to access the details of their bills as extra steps of login and accessing another platform are required.

2.2.2 Automatic Bill Payment (ABP)

Automatic bill payment is a transfer of money scheduled on a predetermined date or trigger to pay a recurring bill. Auto-billing are payments routine that can be made from a bank, E-Wallet or mutual fund account. Traditionally, ABP are commonly set up with the billing companies which offer subscription services. However, it is also possible to schedule automatic payments through a third-party platform which offer bill payments services. ABP are now widely implemented throughout the world, it can be scheduled for all types of payments transactions. This includes, instalment loans, credit card bills, electric bills, subscription bills and more. To setup automatic bill payment, bill payer has to make arrangements with the bank to make the exact payment each month. This power is given to the billing companies to charge the bill payers for whatever amount is owed in that particular billing cycle.

One of the advantages of using ABP is payments are easy to automate from a checking account, a bill payer can simply register for auto-billing at any branch of the billing companies or register online through the mobile application. Besides, enabling auto-billing also helps bill payers to avoid late payments. With the avoidance of late payments, bill payers do not have to worry about remembering the billing date and service interruption. In this case, the bill payers are able to improve or maintain a good credit score, which are helpful to the bill payers when they would like to apply for a loan. Last but not least, auto-billing can help bill payers to reduce their task, bill payers with auto-billing registered don't have to keep doing the repetitive bill payments task every month.

However, using ABP also come with drawbacks. Firstly, if bill payers are not maintaining sufficient balance in his checking account, automatic payments will lead to bounced payments, which is late fees. Additionally, bill payer relying on automatic payments could overlook mistakes or excessive charges that should not be charged. This is because the payment is process without bill payer's manual review. Moreover, once bill payers have registered with automatic payments, it might be difficult for them to cancel it, which potentially leading to ongoing charges even if the bill payers no longer using the services. Furthermore, ABP also causes a salience effect, which consumer will increase the utilities usage unintentionally when they are using ABP. Results of a research has shown that ABP enrolment increases residential electricity consumption by 4% and commercial electricity consumption by 8.1% (Sexton, 2015). This will lead to new problems such as energy wastage and increase of utilities fee.

2.3 Review of Similar applications

Nowadays, there are a lot of applications that provide bill payment functionality. In this section, the types of applications that would be analysed and compared are e-Commerce application, E-Wallet application, banking application and the application provided by the billing company. The steps of the bill payments in each application are analysed. The goal of this section is to find out the pros and cons of each application type and identify the attributes that should be implemented on this project.

2.3.1 E-Commerce Applications (Shopee)

2.3.1.1 Introduction

Shopee is an e-commerce platform in Southeast Asia that was launched in year 2015, it is a platform providing customers with a convenient to do online shopping. Besides, Shopee also hosting a secure E-Wallet platform which enable users to pay all their bills online. Shopee claims that it is easy to pay the bills online using Shopee application, and it also comes with many perks and benefits. Shopee provide bill payers with vouchers to save more on their payment, as well as Shopee cashback to receive Shopee coins that can be used as rebate for the next bill. Shopee also provide multiple payment methods such as ShopeePay, debit card, credit card and major E-Wallet. In addition, Shopee also enable users to link their Maybank2U with Shopee, to make the bill payment online one click away.

2.3.1.2 Steps to pay bills online on Shopee

Firstly, user has login their Shopee account first. On the home page, select “see more”, in order to see more options. Inside the “all products and services” section, select Prepaid, Bills & Tickets to go into the bill payment module.

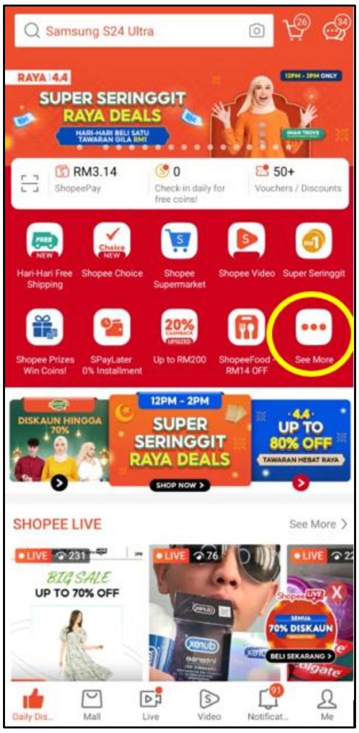


Figure 2.3: Shopee home page

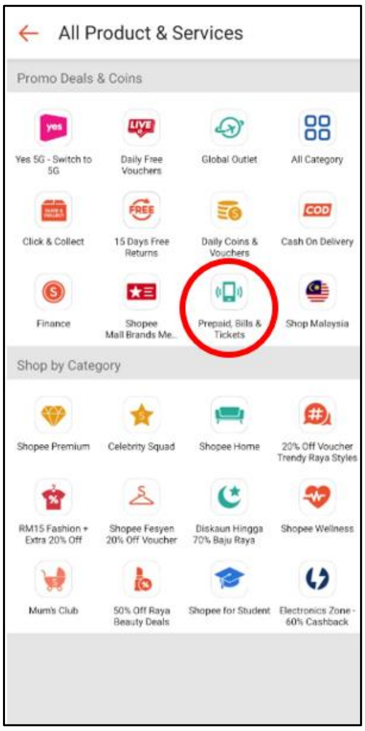


Figure 2.4: Shopee all product and services page

Inside the “Prepaid, Bills & Tickets module, select the type of bill to pay. The available options are postpaid, Astro, internet & voice, electricity, water, local council and PTPTN. A postpaid bill is selected here as an example.

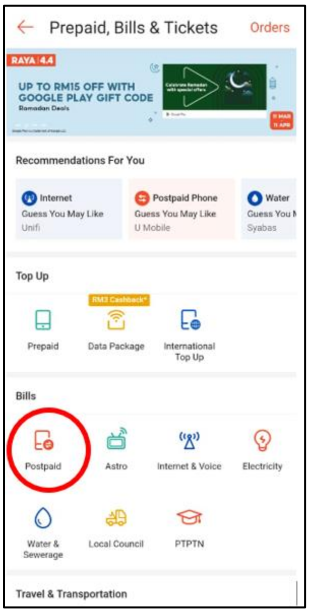


Figure 2.5: Shopee Prepaid, Bills & Tickets page

After selecting the type of payment, the options of available billing companies are shown. Select the billing company to pay the bill.

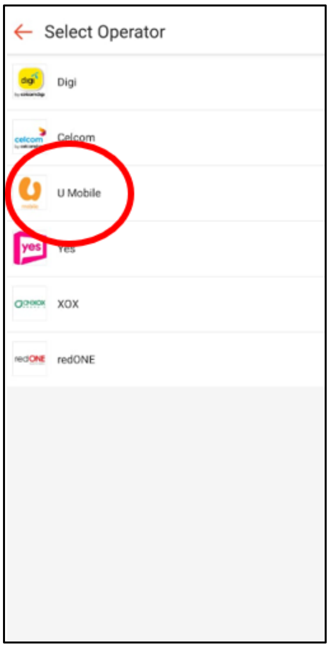
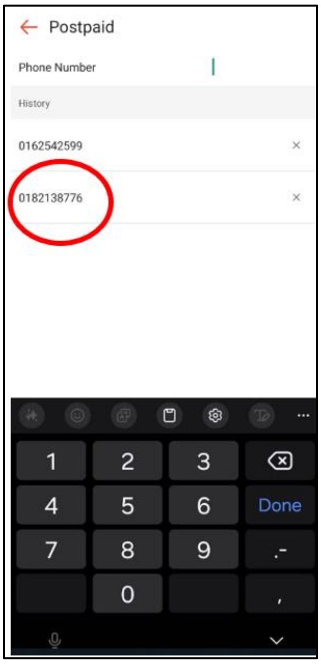


Figure 2.6: Select Operator in Shopee

The next step is to key in the account details for bill payments such as account number or phone number. Shopee will hold the history of payment so that users do not need to memorise or key in the account number or phone number every single time.



Postpaid

Phone Number

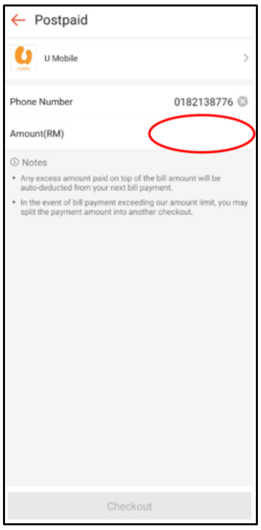
History

0162542599 ×

0182138776 ×

Figure 2.7: Key in payment credentials in Shopee

After pressing the “Done” button, user is asked to key in the amount to pay. The checkout button is lighted up indicating the amount to pay is valid, pressing the checkout button will lead to the “Checkout page”.



Postpaid

U Mobile

Phone Number 0182138776

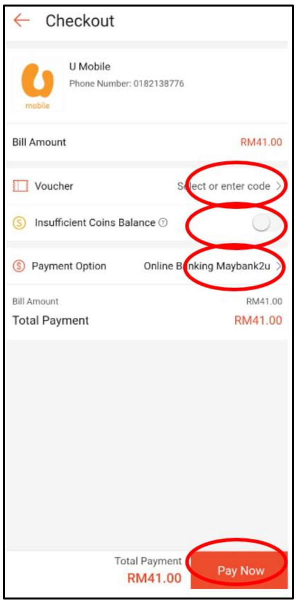
Amount(RM)

ⓘ Notes

- Any excess amount paid on top of the bill amount will be auto-deducted from your next bill payment.
- In the event of bill payment exceeding our amount limit, you may split the payment amount into another checkout.

Figure 2.8: Key in amount to pay in Shopee

In the checkout page, confirmation message are shown, users can select to use voucher, coins, or both together to get discount for their bill. The voucher available areshown after user clicked the “Select or enter code” button. The total payment amount after the discount areshown at the button of the page. Besides, user also can press the “payment option” button to change their payment mehod, or else payment aredone with the recommended payment option. After confirmation, pressing the “Pay Now” button will bring user to the payment gateway of a third-party application or show the modal of ShopeePay, depending on the payment method chosen by the user. For example, if ShopeePay is selected as the payment method, user has to key in the password of their ShopeePay to approve the transaction, the credit inside the ShopeePay wallet arededucted.



Checkout

U Mobile

Phone Number: 0182138776

Bill Amount RM41.00

Voucher Select or enter code

Insufficient Coins Balance

Payment Option Online Banking Maybank2u

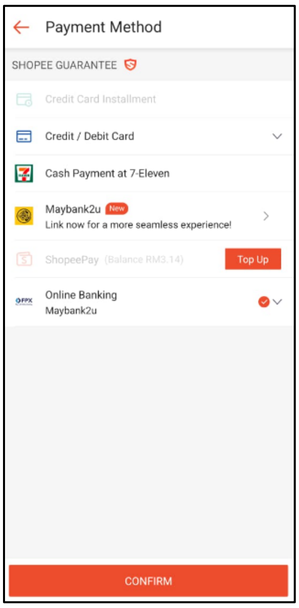
Bill Amount RM41.00

Total Payment RM41.00

Total Payment RM41.00

Pay Now

Figure 2.9: Checkout page in Shopee



Payment Method

SHOPEE GUARANTEE

Credit Card Installment

Credit / Debit Card

Cash Payment at 7-Eleven

Maybank2u New

Link now for a more seamless experience!

ShopeePay (Balance RM3.14) Top Up

Online Banking

Maybank2u

CONFIRM

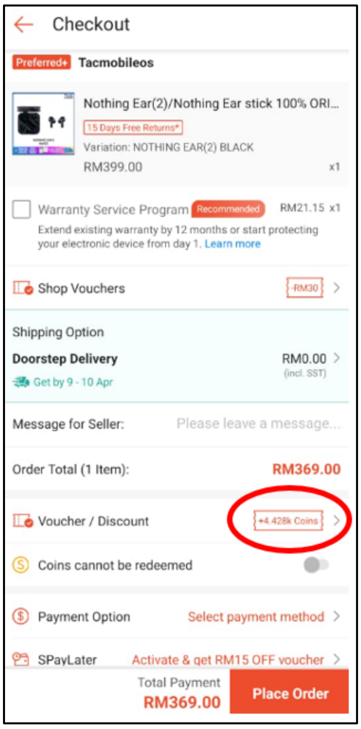
Figure 2.10: Payment method page in Shopee

2.3.1.3 Pro of paying bills using Shopee

One of the benefits of paying bill using Shopee is the ability to remember to billing account details. Shopee will hold the history of the account number/ phone number of the bill's user paid. Thus, user doesn't have to key in the billing account details each time.

Besides, Shopee also provides voucher for bill discount at certain time, such as "eleven-eleven" (11th November), payday (25th of each month) and 15th of each month. Bill payers can claim the vouchers at

Furthermore, Shopee user also able to get coins cashback when they make purchases using the cashback vouchers in Shopee. Shopee usually give coins cashback voucher of 10%, 12%, 15%, 20% and even up to 25%, and every 100 Shopee coins can be used for RM1 discount. For example, user who purchase an item of Rm369 with 12% coins cashback voucher will receive 4.428k of Shopee coins cashback. This 4.428k coins can be treated as Rm44.28 that can be used to deduct the amount of bill payment.



Checkout

Preferred+ Tacmobileos

Nothing Ear(2)/Nothing Ear stick 100% ORI...
15 Days Free Returns*
Variation: NOTHING EAR(2) BLACK
RM399.00 x1

□ Warranty Service Program Recommended RM21.15 x1
Extend existing warranty by 12 months or start protecting your electronic device from day 1. Learn more

Shop Vouchers -RM30 >

Shipping Option
Doorstep Delivery RM0.00 >
(incl. SST)
Get by 9 - 10 Apr

Message for Seller: Please leave a message...

Order Total (1 Item): RM369.00

Voucher / Discount +4.428k Coins >

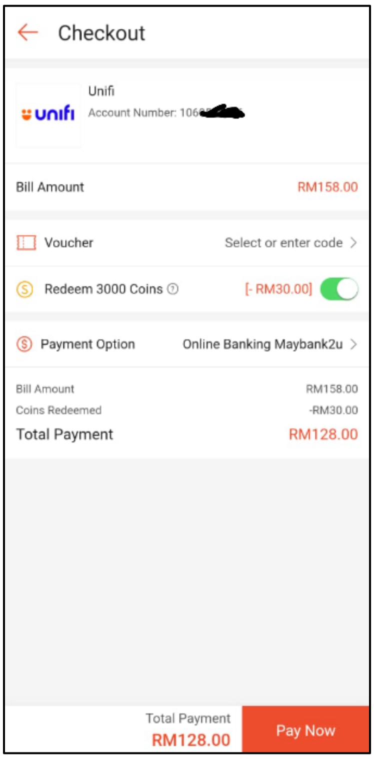
Coins cannot be redeemed

Payment Option Select payment method >

SPayLater Activate & get RM15 OFF voucher >

Total Payment RM369.00
Place Order

Figure 2.11: Applying coins cashback voucher during a purchase



Checkout

Unifi
Account Number: 106[redacted]

Bill Amount RM158.00

Voucher Select or enter code

Redeem 3000 Coins [- RM30.00]

Payment Option Online Banking Maybank2u

Bill Amount	RM158.00
Coins Redeemed	-RM30.00
Total Payment	RM128.00

Total Payment RM128.00

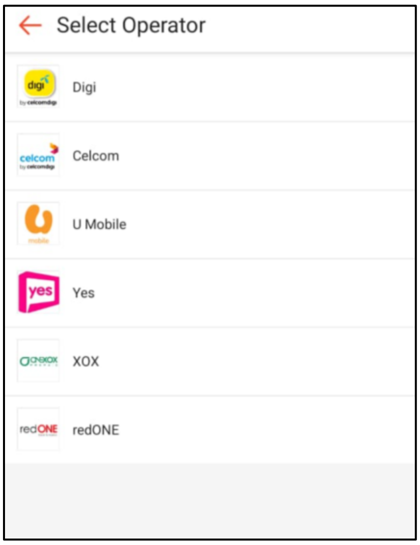
Figure 2.12: Coins rebate during bill payment

2.3.1.4 Cons of paying bills using Shopee

There are also some disadvantages of making payment using Shopee. Firstly, the bill paid is not reflected immediately, it takes slightly longer time for Shopee to process the bills. According to the Shopee official website, a successful payment for bills will be updated between 1 to 3 working days. As the impact, user who are paying bill last minute will have issues such as late payment or barred account. Besides, account that is barred due to late payment will have to wait for a longer period to unbar as there is a waiting period until the payment is processed.

Besides, Shopee also doesn’t come with a function to remind user for bill payments. Thus, bill payers who like to pay bills using Shopee have pay extra effort to set their own reminder to avoid late payments. In order to avoid the issues of late payments, some applications provide users with the option of automatic bill payments, but Shopee doesn’t provide this functionality. In short, there is no supportive function for a bill payer to avoid late bill payments in Shopee, bill payers who choose to pay bill using Shopee has to be always alert to their billing dates and deadline by checking their email or reminder from the billing companies.

Furthermore, another disadvantage is low availabity. Shopee doesn’t support bill payment for all the billing companies available. For example, Maxis and Hotlink bills users are not able to pay their bills at Shopee.



Select Operator

Digi

Celcom

U Mobile

Yes

XOX

redONE

Figure 2.13: Telco operators available in Shopee bill payment

In addition, bill payers who choose to pay their bills using Shopee are most likely drawn to the benefits provided by Shopee, especially the coins rebate and voucher discounts. However, Shopee vouchers are not always available. Thus, bill payers might have to wait until a certain time when the voucher is claimable on Shopee. To exacerbate the situation, some vouchers operate on a first-come-first-served basis, requiring users to snatch the voucher promptly and use it as soon as possible before it is depleted. Consequently, the difficulty and effort needed to claim, "snatch", and use the vouchers become a burden for bill payers. Bill payers who intend to use the vouchers for discounts will often wait and keep their bills unpaid until the dates when Shopee makes the vouchers available. However, failure to claim the voucher after a long waiting period and effort expended will lead to user disappointment. Besides, Shopee coins obtained from Shopee purchases will be received by buyer when the order is completed. Thus, bill payers may spend time waiting to receive the item and complete the order. The waiting time spend may cause late payment and result in poor credit.

Next, Shopee also doesn’t include the functionality to pay multiple bills at once, even within the same billing company. In short, only a single bill can be pay through a single transaction. For example, a bill payer who responsible to pay Umobile telecommunication bills for his family members of 4, has to repeat the same steps of bill payment for 4 times and make 4 transactions for 4 different numbers.

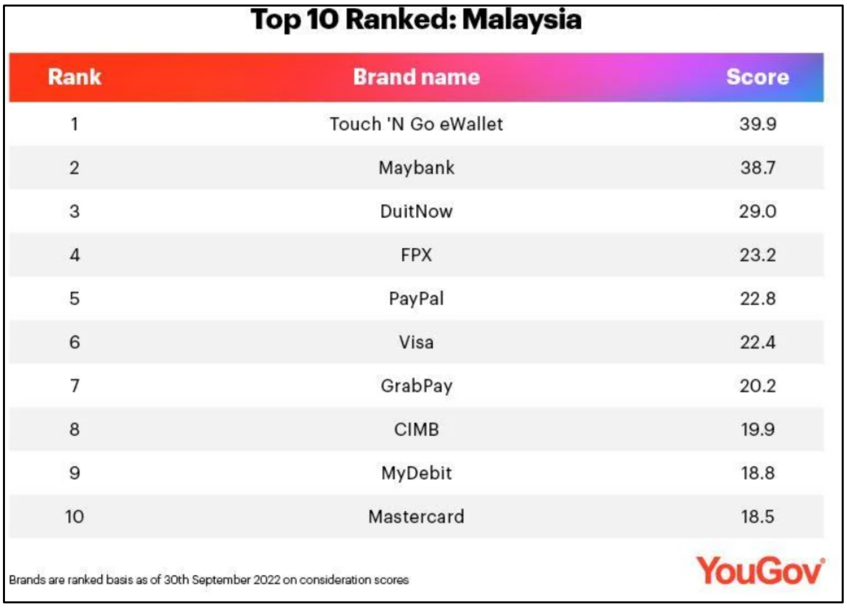
Another disadvantage of using Shopee for bill payment is users unable to verify their account identity. After entering the account number and the amount to pay, users are directed to the payment. Shopee doesn’t retrieve any messages that show users the name, address or identities of the account owner. Thus, users have to be extra careful when entering the account numbers to avoid paying bills to a wrong account.

2.3.2 E-Wallet applications (Touch ‘n Go eWallet)

2.3.2.1 Introduction

Touch ‘n Go (TnG) eWallet is a Malaysian digital wallet and online payment platform, established in Kuala Lumpur, Malaysia in July 2017 as a joint venture between Touch ‘n Go and Ant Financial (Fintech affiliate of Alibaba). It allows users to make payment at merchant through QR code, pay for tolls, street parking, payment on e-hailing, pay bills and top-up mobile prepaid, make credit transfer between multiple users.

During the COVID-19 pandemic and the enforcement of the movement control order, TnG eWallet services become popular as many bill payers rely on it to make merchant payments and other online payments including bill payments. According to the official TnG website, there is over 20 million of registered TnG eWallet users in July 2023. TnG eWallet has ranked No1 in YouGov’s Bank and Payment Brand Rankings of 2022, which is 1.2 points higher than Maybank who placed second (Dudekula, 2022).



Top 10 Ranked: Malaysia

Rank	Brand name	Score
1	Touch 'N Go eWallet	39.9
2	Maybank	38.7
3	DuitNow	29.0
4	FPX	23.2
5	PayPal	22.8
6	Visa	22.4
7	GrabPay	20.2
8	CIMB	19.9
9	MyDebit	18.8
10	Mastercard	18.5

Brands are ranked basis as of 30th September 2022 on consideration scores

YouGov

Figure 2.14: Banks and Payment brand ranking 2022. (Adopted from Reema Dudekula. (2022))

2.3.2.2 Steps of bill payment using TnG eWallet

There are two methods to pay bills using TnG eWallet, which are manual bill payment and automatic bill payment. Both bill payment methods will use the credit topped up in TnG eWallet, the amount paid are deducted from the credit.

To manually pay a bill, users have to firstly login the TnG eWallet. At the home page, the users have to make sure his credit available are sufficient to pay their bills. If the credit is not sufficient, users have to reload their E-Wallet balance by either DuitNow Transfer, Reload PIN or setup auto-reload.

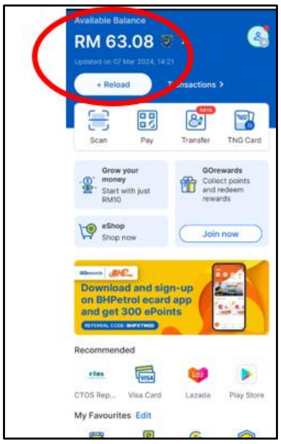
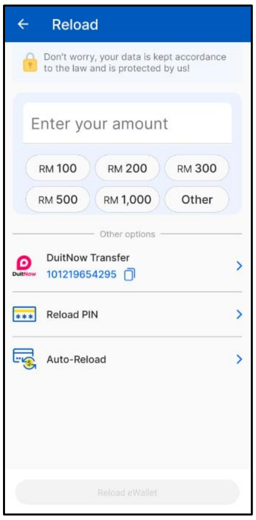


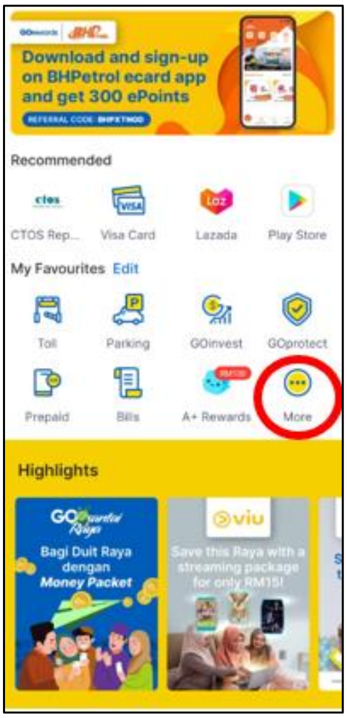
Figure 2.15: TnG eWallet home page



Don't worry, your data is kept accordance to the law and is protected by us!

Figure 2.16: TnG eWallet Reload Page

To navigate to bill payments module, users have to scroll down from the home page the find "Bills" button under "My Favourite" section. If the button cannot be found, click the "More" button. Scroll down to the "Bills & Prepaid" section. At this section, some billing companies are shown, such as TNB, Maxis and Sohoj. Users may click the logo of the billing company if they wish to pay the respective bill, or else click the "Bills" button.



Download and sign-up on BHPetrol ecard app and get 300 ePoints
Recommended
CTOS Rep...
Visa Card
Lazada
Play Store
My Favourites Edit
Toll
Parking
GOinvest
GOprotect
Prepaid
Bills
A+ Rewards
More
Highlights
Bagi Duit Raya dengan Money Packet
viu
Save this Raya with a streaming package for only RM15!

Figure 2.17: "More" button on TnG eWallet's Home Page

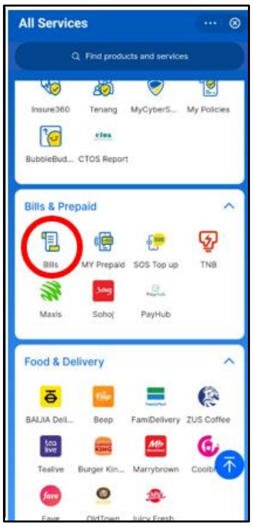


Figure 2.18: TnG eWallet “All Services” page

At the “Bills” module, there is a section which store the recent/saved Bills, which ease users to access the billing companies they would like to make payment. By pressing the “+” button of “Save your favourite bills” or “View More” button, both will redirect users to the menu of categorised billing companies.

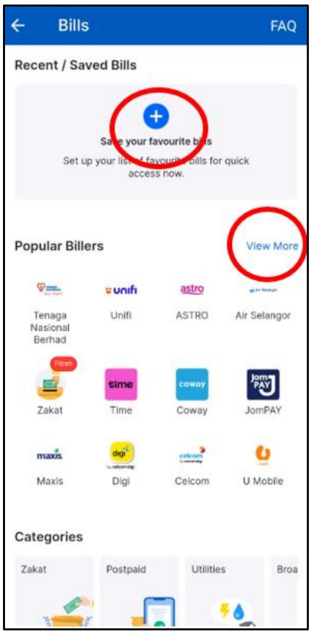


Figure 2.19: TnG eWallet Bills Page

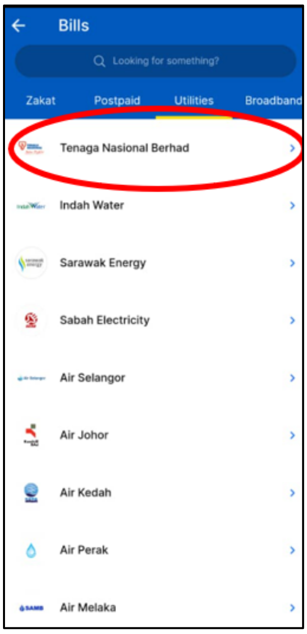


Figure 2.20: TnG eWallet Select Bills

By selecting a billing company, users are redirected to a page to key in the account number of the bill that users wish to pay. After entering the account number, users are redirected to next page where they have to input the pay amount. However, the outstanding amount of the bills will not be shown, users have to login the application of their respective billing company to check the amount.

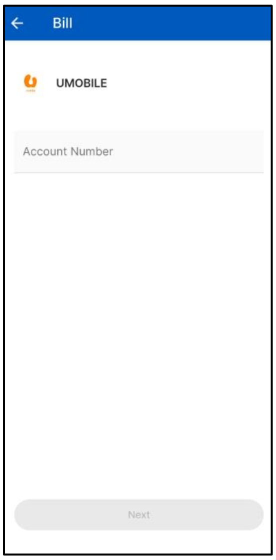
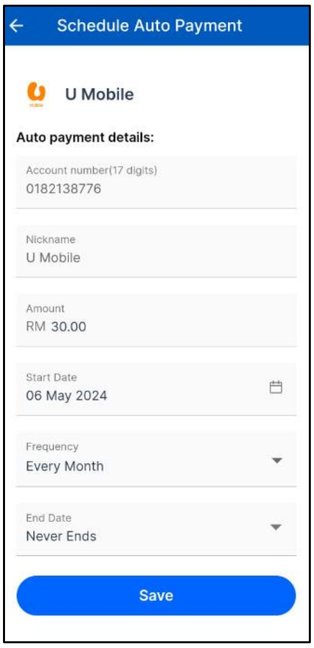


Figure 2.21: TnG eWallet Key in account number

After entering the valid payment amount, users can choose either “Pay” or “Schedule Payment”. If users press Schedule Payment, users have to setup the starting date and the frequency of auto payment. In the end of confirming one-time payment or schedule payment, transaction authorization and approval through 6-digit pin or biometric verification are required.



Auto payment details:

Account number(17 digits): 0182138776

Nickname: U Mobile

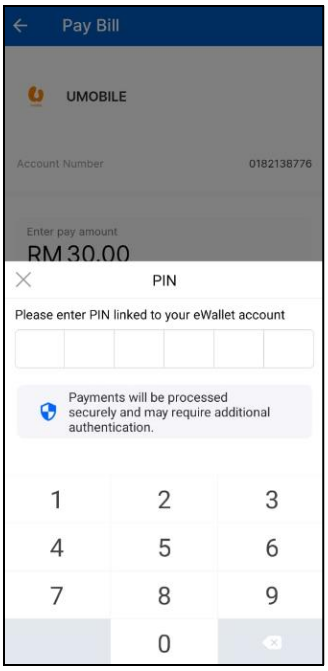
Amount: RM 30.00

Start Date: 06 May 2024

Frequency: Every Month

End Date: Never Ends

Figure 2.22: TnG eWallet Schedule Auto Payment Page



Account Number: 0182138776

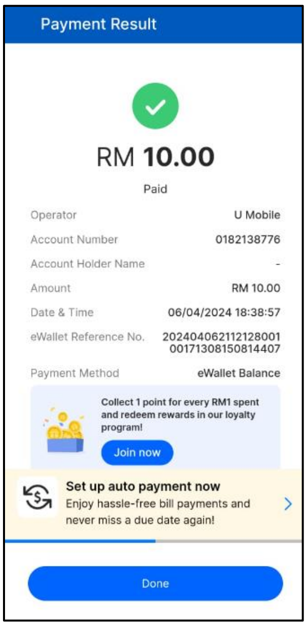
Enter pay amount: RM 30.00

Please enter PIN linked to your eWallet account

Payments will be processed securely and may require additional authentication.

Figure 2.23: TnG eWallet Input Pin

A successful transaction will generate a digital receipt to users immediately. However, the transaction may take up to 24 hours to be processed and be completed.



Payment Result

RM 10.00

Paid

Operator	U Mobile
Account Number	0182138776
Account Holder Name	-
Amount	RM 10.00
Date & Time	06/04/2024 18:38:57
eWallet Reference No.	20240406211212800100171308150814407
Payment Method	eWallet Balance

Collect 1 point for every RM1 spent and redeem rewards in our loyalty program!

Set up auto payment now

Enjoy hassle-free bill payments and never miss a due date again!

Figure 2.24: TnG eWallet Payment Receipt

2.3.2.3 Pros of bill payment using TnG eWallet

One of the advantages of paying bill with TnG eWallet is the quicker processing time compared to other third-party bill payment platform. A complete transaction of bill payment will be updated and reflected in 24 hours, which is comparatively quicker. Thus, users who are facing issues like late payments or barred account can solve their problems quicker.

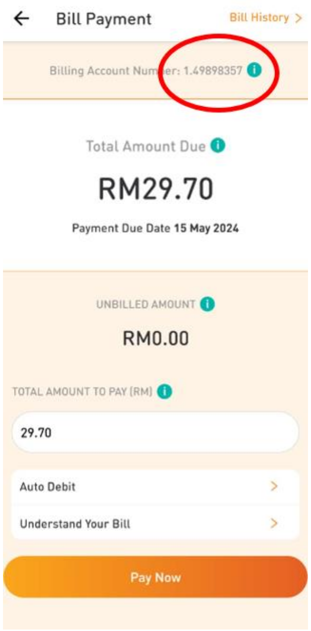
TnG eWallet also store the account number of the bills that users recently paid. Thus, users do not need to key in the credential information each time when he is paying bill.

Next, TnG eWallet also provides auto billing functionality, by schedule the payments of a bill at a fixed amount and a fixed frequency of payment time. This schedule billing is suitable for bills recurring on a fixed billing date, especially the bills with a fixed charge. In addition, TnG eWallet also provides function to schedule credit top-up, this function linked a bank account to TnG eWallet. Schedule top up will ensures TnG eWallet always have sufficient credit for the automatic bill payment in schedule.

Besides, TnG eWallet also categorised bills according to their types. In addition, TnG eWallet also more enable bill payments to more billing companies, including the less-popular companies. Thus, a bill payer can easily find the billing company he wants to make payment through the category of the bill. A higher availability of the companies also ensures that the companies can be easily found in TnG eWallet.

2.3.2.4 Cons of bill payment using TnG eWallet

TnG eWallet also comes with several cons, which bring inconvenience and bad user experience. Firstly, the incomplete and unclear meaning of “account number”. When users choose to make a payment of telco bills, such as Digi, Maxis and Umobile, users are asked to enter the account number, as shown in figure 2.21. Unlike other applications that ask for phone number, TnG eWallet required users to key in account number. The example of account number for a telco account is shown in the figure 2.25, which is very hard for users to remember it. Thus, before users can key in the account number, they have to note down or memorise the account number from the application of the billing company, or from the bill statement. According to the TnG eWallet official website, entering phone number will also lead the users to their billing account. However, this is not clearly stated in the textbox, the placeholder “Account Number” shown in figure 2.21 is giving incomplete information.



Bill Payment

Billing Account Number: 1.49898357

Total Amount Due

RM29.70

Payment Due Date 15 May 2024

UNBILLED AMOUNT

RM0.00

TOTAL AMOUNT TO PAY (RM)

29.70

Figure 2.25: Account number shown in MyUMobile

In addition, another inconvenience of using TnG eWallet is the unsure outstanding amount of the bills. After entering the account number and linked to the user's billing account, the outstanding amount will not be retrieved and shown. Thus, users have to refer back to the bill reminder sent by the billing companies through the email or refer back to the application provided by the billing company.

Another disadvantage of using TnG eWallet is users unable to verify their account identity. After entering the account number and the amount to pay, users are directed to the payment transaction by entering the 6-digit pin, there is no confirmation message that show users the name of the account owner. Thus, users have to be extra careful by double checking the account number entered is correct before approving the transaction.

While schedule billing help to save up user's time and effort through ABP, it also comes with an issue. Schedule billing can only be set up with a fixed amount of payment. Thus, users must make sure that their bills have no increase or decrease in charge to avoid excessive or insufficient payments. Besides, fluctuating bills are also not suitable for schedule billing as the bills will have different charges each time. For example, utilities bills like water and electricity are charged based on usage of the billing cycle.

As similar to Shopee, TnG eWallet also doesn’t include the functionality to pay multiple bills at once, even within the same billing company. In short, only a single bill can be pay through a single transaction. If users wish to pay multiple bills, multiple transactions are needed to pay all the bills. At the same time, Tng eWallet also doesn’t provide bill reminder functionality.

2.3.3 Banking Application (MAE)

2.3.3.1 Introduction

Maybank Anytime Everyone (MAE) is a banking application, that allow Maybank customers to open Maybank E-Wallet account through the Interoperable Credit Transfer Framework (ICTF) that issued by Bank Negara Malaysia in 2018. MAE is created for users to manage their expenses for cashless transactions and bill payments.

2.3.3.2 Steps of bill payment using MAE

To make a bill payment using MAE, users must login and authorised themselves using biometric authentication or account password. At the home page, click the “Pay Bills” button, users are required to key in their M2U password before they can make payments.

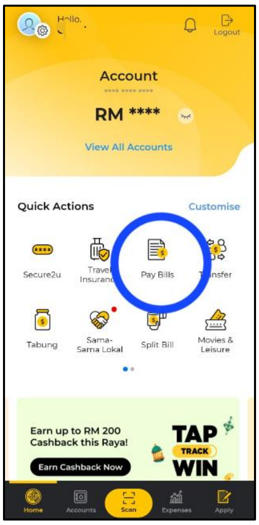
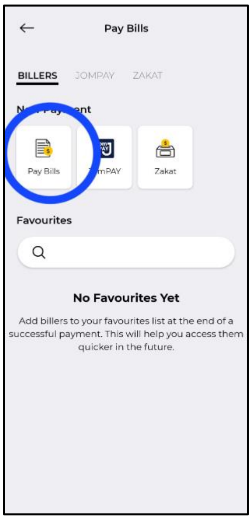


Figure 2.26: MAE Home Page

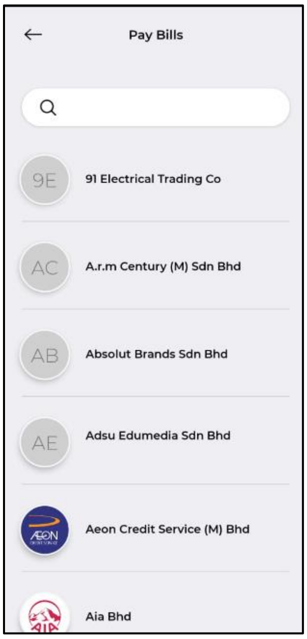


No Favourites Yet

Add billers to your favourites list at the end of a successful payment. This will help you access them quicker in the future.

Figure 2.27: MAE Pay Bills Page

After entering the “Pay Bills” module, users have to find their billing companies through scrolling or searching the company name.



91 Electrical Trading Co

A.r.m Century (M) Sdn Bhd

Absolut Brands Sdn Bhd

Adsu Edumedia Sdn Bhd

Aeon Credit Service (M) Bhd

Aia Bhd

Figure 2.28: MAE Select Billing Companies

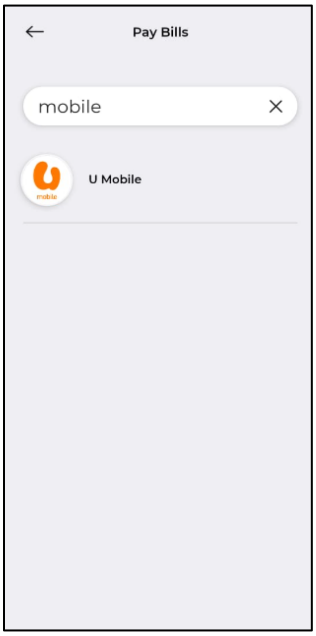


Figure 2.29: MAE Search Billing Company

After choosing the billing company, users are asked to enter the account number or other credential information.

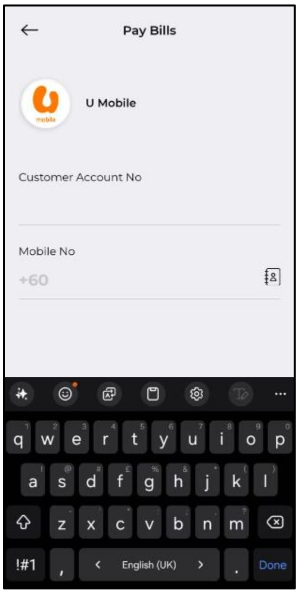
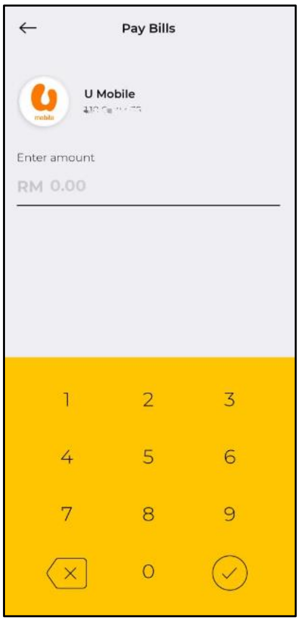


Figure 2.30: MAE enter account details

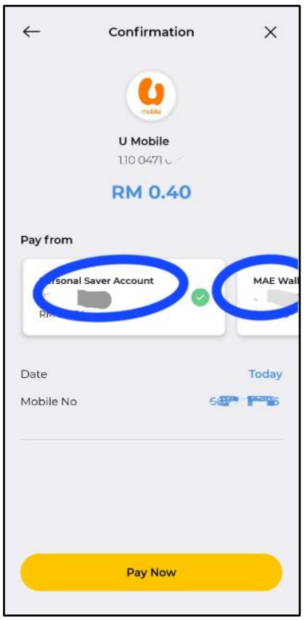
The next step is to enter the amount to pay. At confirmation page, users are able to select to pay from which account, either their Maybank personal saver account or their MAE Wallet.



Enter amount

RM 0.00

Figure 2.31: MAE enter payment amount



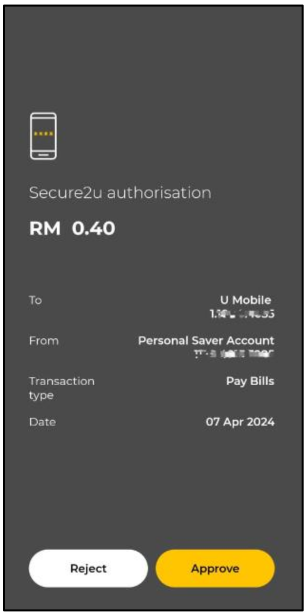
RM 0.40

Date Today

Mobile No [redacted]

Figure 2.32: MAE Payment Confirmation Page

After confirming the account and clicked “pay”, users are required to approve the Secure2u authorisation by simply selecting “approve” or “reject”.



Secure2u authorisation

RM 0.40

To U Mobile
[redacted]

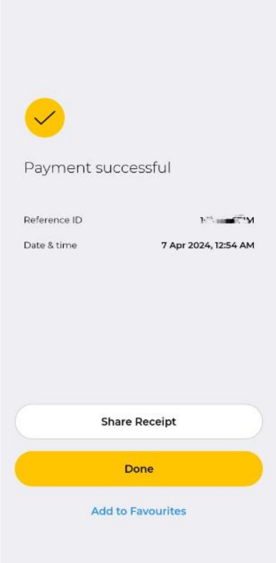
From Personal Saver Account
[redacted]

Transaction type Pay Bills

Date 07 Apr 2024

Figure 2.33: MAE payment authorisation page

After the successful payment, receipt is shown. Users can save the bill as favourite, choose to share the receipt or simply complete the transaction by clicking “Done” button.



Payment successful

Reference ID [redacted]

Date & time 7 Apr 2024, 12:54 AM

Figure 2.34: MAE payment receipt

2.3.3.3 Pros of bill payment using MAE

One of the benefits of using MAE is it provides functionality to add billers as favourites list at the end of a successful bill payment, MAE will store the billing company and the account number of the bills. Thus, users do not need to key in the credential information each time when he is paying bill.

Another advantage of using MAE is high availability of choices, up to 3000 billing companies. Thus, user can pay most of their bills using a single platform.

In addition, in term of number of clicks and time needed to make a bill payment, MAE are relatively user friendly and quicker, less steps are required to make a bill payment when compared to other platforms. In short, one of the benefits of MAE bill payment is high efficiency.

2.3.3.4 Cons of bill payment using MAE

One of the disadvantages of using MAE to pay bill is extra effort required to apply for auto billings. If users want to make auto bill payments through Maybank, user has to go through a tedious applying process. The steps are as such:

1. Print out enrolment Form.
2. Fill in the enrolment Form with personal details, Maybank card details and bill details.
3. Email the enrolment form to respective email of Maybank.
4. Wait for 1-2 months for the ABP service to be activated.

Besides, when using the MAE application to pay bill, there is no categorisation of billing companies depending on the services they provide. Thus, it would be difficult for user to scroll through the long list to search for the billing company they want to pay for. Instead, user has to key in the company name to search for it.

There are several disadvantages like other applications, which are mentioned in section 2.3.1.4 and 2.3.2.4. Thus, it will only be mentioned but not be re-explain again under this section. These disadvantages are unshown outstanding amount, inability to verify account identity, difficult steps to change auto billing amount, only able to pay fixed amount with auto billing, low availability of billing companies with auto billing, unable to pay multiple bills at once and no reminder of bill payments.

2.3.4 Application provided by billing companies (myTNB)

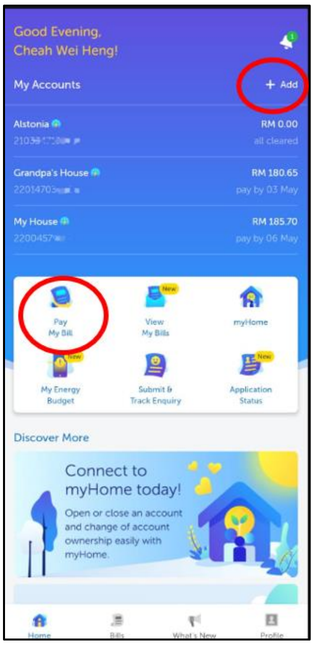
2.3.4.1 Introduction

Most of the billing companies nowadays provide an application for their users to access the details of the service they offer and also allow users to make payments online.

myTNB app is a free mobile application for users to manage their TNB electricity accounts. Users can view and pay their electricity bills anytime, anywhere as long as they have access to the internet. myTNB application allow users to find out electricity consumption, view the total electricity bill for all the TNB accounts and submit feedback for any bill related matters.

2.3.4.2 Steps of bill payment using myTNB

After login into a registered account, user aredirected to the home page which contains the name of the billing accounts, the account numbers and the outstanding amount. If there is no registered account, users can add an account using the “+Add” button. When adding an account, users have to key in an account number and users are allowed to setup a nickname for better recognition.



Good Evening,
Cheah Wei Heng!

My Accounts

Alstonia — RM 0.00
2103[redacted] — all cleared

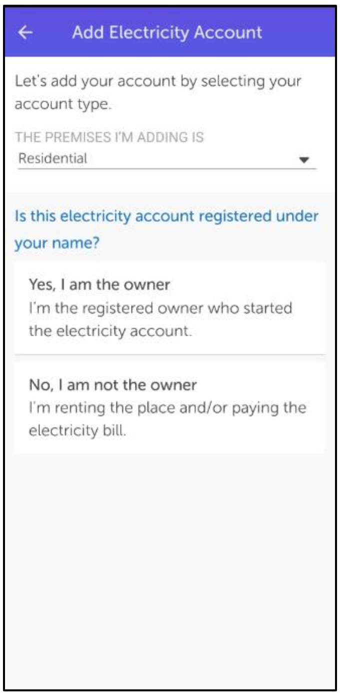
Grandpa's House — RM 180.65
22014703[redacted] — pay by 03 May

My House — RM 185.70
2200457[redacted] — pay by 06 May

Discover More

Connect to myHome today!
Open or close an account and change of account ownership easily with myHome.

Figure 2.35: MyTNB Home Page



Add Electricity Account

Let's add your account by selecting your account type.

THE PREMISES I'M ADDING IS
Residential ▾

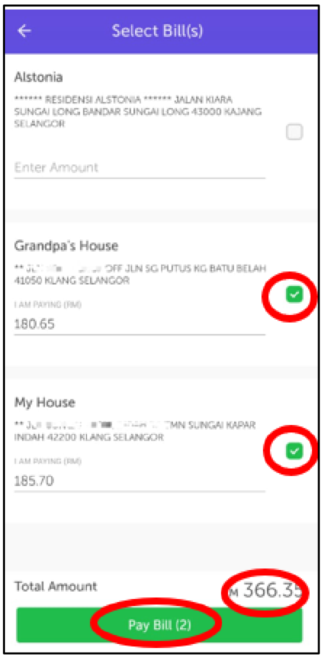
Is this electricity account registered under your name?

Yes, I am the owner
I'm the registered owner who started the electricity account.

No, I am not the owner
I'm renting the place and/or paying the electricity bill.

Figure 2.36: MyTNB Add Electricity Account Module

To pay the bills, user can simply press the "Pay my bill" button at the home page. Users are asked to select bill(s) when there are multiple bills available. After checking the accounts with outstanding bills, the total number of the bills and total amounts of the bills are calculated by the system and shown to the users.



Select Bill(s)

Alstonia
****** RESIDENSI ALSTONIA ****** JALAN KIARA SUNGAI LONG BANDAR SUNGAI LONG 43000 KAJANG SELANGOR ☐
Enter Amount

Grandpa's House
** [redacted] OFF JLN SG PUTUS KG BATU BELAH 41050 KLANG SELANGOR ☑
I AM PAYING (RM)
180.65

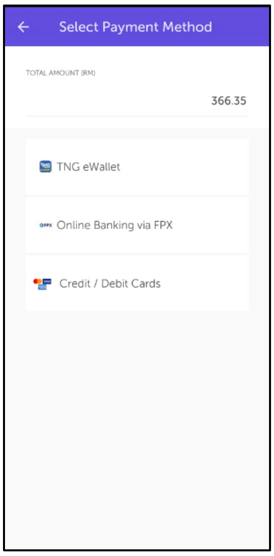
My House
** [redacted] TMN SUNGAI KAPAR INDAH 42200 KLANG SELANGOR ☑
I AM PAYING (RM)
185.70

Total Amount M 366.35

Pay Bill (2)

Figure 2.37: Select bills for payment

To continue, users can select one of the payment methods available, which are TNG eWallet, Online Banking FPX or using credit/debit card. Users areredirected to the payment gateway after choosing the desire payment method.



TOTAL AMOUNT (RM)
366.35

Figure 2.38: MyTNB select payment method

2.3.4.2 Pros of bill payment using myTNB

One of the advantages of paying bill using myTNB is the ability to pay multiple bills, as shown in the section 2.3.4.1. Thus, user only has to make 1 transaction to TNB if they want to pay three different electricity bills.

Besides, another benefit of using myTNB is users able to know the exact outstanding amount they have to pay. Thus, users do not need to check their bill statements through the emails or printed bill.

In additional to that, users are also able to understand the details of their bill. By pressing the button “View My Bills” at home page, bill statement is shown.

Even though these advantages of myTNB application are not available on all the applications provided by other billing companies, but most of the applications are having the similar advantages.

2.3.4.3 Cons of bill payment using myTNB

One of the disadvantages of using myTNB is users can only pay the bills of a single company using 1 application. Thus, users have to use multiple applications so that they are able to pay all their different bills of different billing companies.

Secondly, auto-billing services are mostly not available within the application provided by the billing company. Thus, users who pay bills using this method must manually pay their bills every month.

2.3.5 Summary of Comparison

After comparing the 4 types of bill payment applications, the advantages and disadvantages of each application are summarised and stated in Table 2.1.

Table 2.1: Advantages and disadvantages of applications compared

Compared Application	Advantages	Disadvantages
Shopee	-Remember account details of bills paid previously. -Offers vouchers for bill discounts. -Provides coins cashback voucher for purchases, which can be used to deduct bill payments amount.	-Payment may not be reflected immiediately. -Lacks bill payments reminder functionality. -Lacks auto bill payment functionality. -Low availability of supported billing companies. -Unable to check outstanding amount of bills. -Lack of identity verification during the payment process.
TnG eWallet	-Stores account numbers of recent bill payment. -Offers auto-billing and automatic credits top-up. -Categorise billing companies according to	-Unclear instruction of credentials information needs for payment -Unable to check outstanding amount of bills.

	types of services provided. -Faster payment processing speed.	-Lack of identity verification during the payment process.
MAE(Maybank)	-Allow users to save account by adding billers to favourite. -High availability of billing companies. -Relatively lesser click and steps for bill payments.	-Very tedious process to apply auto bill payment. -No categorisation of billing companies, making it difficult to find. -Unable to check outstanding amount of bills. -Lack of identity verification during the payment process.
MyTNB	-Allow users to pay multiple bills in a single transaction. -Provide users with exact outstanding bill amounts and detailed bill statements.	-User can only pay bills for a single company using the application. -Auto billing services are mostly unavailable.

After learning the advantages and disadvantages of the proposed system. Most of the advantages mentioned above will be applied on the proposed system while the disadvantages will be avoided.

2.4 Comparison of Frameworks

In this section, development frameworks are researched. There are multiple ways to develop a mobile application. However, there is no single solution to suit all the problems. Therefore, pros and cons of each framework will be studied to identify the most suitable development framework for this project.

The top three mobile app development frameworks in year 2024 are React Native, Flutter and Xamarin (Most popular mobile app development Frameworks for app developers, 2024). In this section, these 3 development frameworks are studied and compared.

2.4.1 React Native

React Native is a framework developed by Facebook. React Native is a highly accessible, cross-platform application development framework which enable single development to work on both Android and IOS devices. React Native allows software engineers to build platform-specific versions of multiple function, by using a single codebase across all platforms. React Native design primarily uses JavaScript along with the JSX, which is similar to React for web development. However, React Native does not use HTML and CSS, but it provides its own set of components and styles which are similar. These components work about the same but are specific to mobile app development only. One of the advantages of using React Native is because it enables quicker development with higher efficiency. Reusable components and pre-developed components for front-end development are key issues that increase the productivity (Kamienski, 2023). React Native also allow applications to use functionality such as the user’s location or the phone camera by using platform APIs. The limitations of React Native are not efficient for complex interfaces, it is not suitable to achieve the top performance in certain scenarios, especially when handling complex animations. Besides, React Native also has a complicated process of updating version.

2.4.2 Flutter

Flutter is a free framework created by Google that allows developer to create native Android and IOS applications with a simple codebase, just like React Native. Flutter is unique because it allows developer to create applications that make users feel like they are made for dedicated platform. An advantage of using Flutter is the rapid development of good-looking user interfaces as it comes with multiple functionality that speed up the development process. For example, tools for designing the interfaces, fixing bugs and connecting to other services. Flutter has been used by a few well-known companies such as Google and Abbey Road Studios. The advantages of using Flutter include rapid development, cross-platform, visual-appealing interfaces and high performance. Besides, Flutter also provides native-like performance (Montano, 2024). In addition, Flutter also has a powerful community, as more than 40 percent of software developers have chosen Flutter over the last three year (Montano, 2024). Thus, this will lead to a reliable community support. However, Flutter come with disadvantages as well, which are large and weighty applications, low popularity of Dart programming language, integration issues with iOS and the limited number of third-party libraries (Montano, 2024).

2.4.3 Xamarin

Xamarin is also a cross-platform application development framework for developing Android and iOS application. Xamarin uses c+ programming language as the main development language, unlike React Native and Flutter. The development of application through Xamarin required lesser lines of code, which result in quicker coding process. At the same time, Xamarin also allows developer to rapidly transfer the scripts between multiple systems, such as Windows and macOS. Even though Xamarin enable a higher speed of application development, it does not sacrifice the quality the native application developed. The benefits of Xamarin framework include centralized code sharing which allows writing and testing code in a single project, offer native performance, access to native APIs and support for UI development. On the other hand, Xamarin also contains disadvantages. One of it is the steep learning curve, Xamarin required more time and effort to learn, especially developers who are not familiar with C#. With the included Mono runtime and framework libraries, Xamarin apps tend to be larger file sizes. Besides, the community of Xamarin is not matured as another framework like React Native or Flutter, causing lesser community support and lower availability of resources (Almeida, 2022).

2.4.4 Summary

Table 2. 2: Table of differences between frameworks compared

Feature	React Native	Flutter	Xamarin
Development Language	JavaScript	Dart	C#
User Interface	-	-	Xamarin.Forms
Community Support	Large	Growing	Moderate
Learning Curve	Moderate	Moderate	Steeper
App Size	Small	Larger	Larger
Suitable For	Simple to moderately complex apps	Simple to complex apps	Performance-critical apps with deep integration

React Native, Flutter and Xamarin use three different programming languages. These three frameworks provide good native-like performance. React Native relies on native components, pre-developed components and highlight reusability. Flutter offers a rich set of customizable UI components which are called widgets. Xamarin use platform-specific UI components, enabling developers to create native-looking apps. In term of development, React Native provides a faster development cycle with real-time refreshing, allowing the developers to see changes immediately. Flutter provides feature for quick iteration and experimentation, making the testing earlier and easier. Xamarin offers a robust development, which is high speed and rich in debugging tools. While comparing the community and support, React Native has the largest active community with tons of resources and third-party libraries available. In short, each framework has its strengths and weakness, and the choice depends on many factors. React Native is the chosen development framework due to the large community and ecosystem, cross-platform development using a single codebase, pre-developed components and high reusability that enhance the productivity. Lastly, the integration with React.js website development is also one of the reasons of choice, this property allows easy code sharing from mobile application to website, and ease developers to integrate this application with web application in future.

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

In this chapter, choice of methodology for software development, description of development tools, programming languages, database and the work plan for this project are discussed. Details of each phase are explained in this chapter. In addition, a work breakdown structure and Gantt Chart are also included at the last section of this chapter.

3.2 Software Development Methodology

The incremental development methodology was selected for this project. Incremental methodology is an iterative approach, where the project was broken down into multiple small increments. Each increment was represented by one functional requirement of the project. During the incremental development, continuous feedback and testing are carried out. After each iteration, the feedback and testing were used to improve and refine the project based on its requirements and design. Next phase was started from the design phase again once the previous phase was completed. Furthermore, functionalities/modules of the system are analysed and arranged according to the priority. In other words, the incremental development started with the most important features first. This allowed the delivery of value in the early phase of the project. The increments and iterations were repeated with the next important module until the complete system was developed. In addition, incremental development has a high flexibility to changing requirement, changes on the requirement and adding of module can be accommodated easily by adding a new iteration. At the end of the development, the closing phase was carried out to test the overall system functionality and integration, the documentation and reports were prepared for the maintainability. Figure 3.1 below shows the general idea of increments in this project.

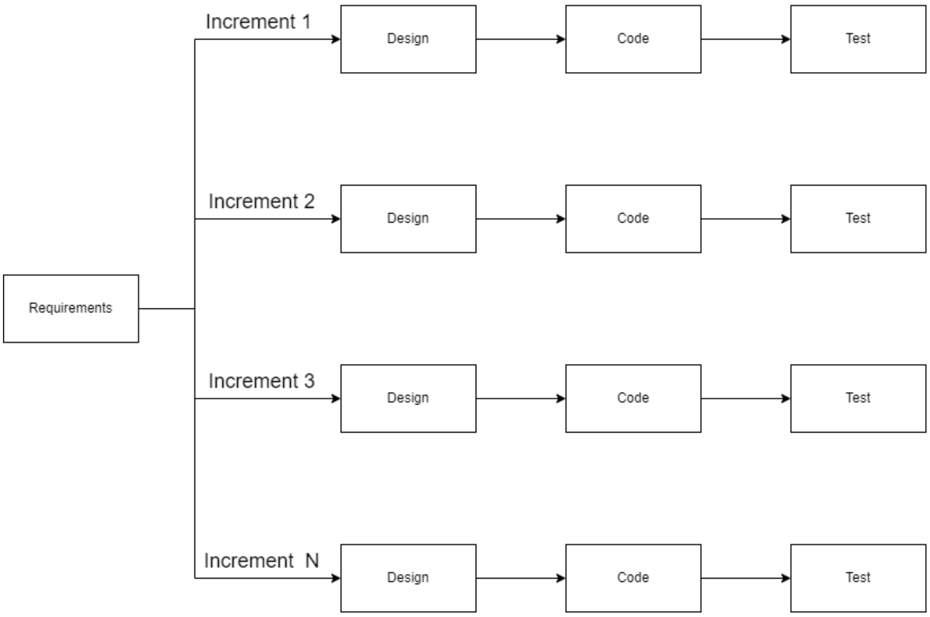


Figure 3.1: Incremental Model applied in this project.

3.2.1 Planning

This is the initial phase of the SDLC development process. A good planning on project’s scope and timeline was carried out to ensure the project’s success. During this planning phase, a major problem with potential solution was identified. Surrounding the problem, a project was initiated by listing the objectives, scope, problem statements and proposed solution. Besides, to achieve the objectives, the development methodology was also defined in this phase. Furthermore, the requirement gathering, and elicitation were conducted to find the needs and preferences of bill payers. At the end of this phase, the project schedule was created to estimate the project duration.

3.2.1.1 Project proposal

The project proposal acts as a foundation for this project. Firstly, the project began with identification of major problems regarding current bill payment system. The target user was also identified, which are common bill payers in Malaysia. Similar bill payment applications were analysed to find out the challenges encountered by most of the bill payers. From the analysis, the problem statements were defined. Thus, the project objectives were listed as the milestones of developing the solution successfully. Furthermore, research and analysis have been carried out to find the suitable platform and development tools. Besides, the suitable methodology was also analysed to manage the overall development processes. Overall, the project proposal served as a blueprint for the project's execution by outlining the key milestones and deliverables and providing clear view of project's goals and objectives.

3.2.1.2 Requirement Gathering and Elicitation

After the approval of the project proposal, requirement gathering were carried out through a questionnaire survey. The questionnaire was created using google form and sent to target users, bill payers and potential bill payers. The distribution of the survey took around 2 weeks. After collecting 50 responses, the survey was finalized and anlysed. The data collected were analysed by the google system to generate useful information such as statistics result. Examples of the information collected are percentage of respondents facing certain issues, average numbers of bills a respondent pay per month and bill analysis preferences of respondents. To further understand the causes/roots of the problems, review of other bill payment applications was also conducted. In addition, literature review was also conducted to understand the pros and cons of each bill payment methods in the history of BMS evolvement. After completing the requirements gathering activities, the requirements were elicited to determine the features required to solve the current problems faced by the bill payers. As the result, use case diagrams was drawn and functional/non-functional requirements were listed for the proposed system.

3.2.1.2.1 Questionnaires

This online questionnaire survey approach was chosen due to its ability to collect data from multiple respondents in a short time and at a lower cost. The questionnaire was designed for two types of respondents, current bill payer who are in charge of household bill payment and potential bill payers who are not currently paying bills. The questions were separated into three different sections, first section collected the respondent's demographic data to identify the category of respondents. The second section collected the processes, feedback, experiences and problems facing by current bill payers only. This section did not collect responses from respondents who was not paying bill at the time. The third section collected the opinion and preferences of the proposed bill payment system from all respondents. This questionnaire only collected the responses from respondents above 18 years sold, the responses are deleted if it is accidentally submitted by respondents below the age. This is because respondents below 18 years old might not have enough understanding regarding the bill payments processes. Additionally, this questionnaire was also controlled to collect information only from people living in Malaysia only. At the end of this survey, 50 responses were successfully collected with 34 bill payers and 16 potential bill payers.

3.2.1.2.2Review of similar applications

As mentioned in section 3.2.1.2, there are 4 types of similar applications reviewed and compared, which are eCommerce application, E-Wallet application, banking application and bill management application provided by the billing companies. For each of the application types, an application was chosen to represent it. The chosen applications were Shopee (eCommerce), TnG eWallet (eWallet), MAE (banking) and myTNB (billing company). After the user walkthrough, the pros and cons identified in each of the applications were listed and compared in section 2.3.5.

3.2.1.3 Project scheduling

During the planning phase, the final objective is to generate a project schedule. The tasks were broken down into multiple small tasks by determine the details and milestones of each stage of the project, a detailed work breakdown structure (WBS) was developed. By referring the WBS, it helped to ensure the essential tasks were not overlooked during the development process.

In addition, the tasks were be ensured that it can be completed within the given time. Thus, a Gantt chart consists of tasks of each incremental phase was developed. Each of the task was listed with its expected starting date and expected finishing date. By summing up the time assigned, the project duration was being estimated. Gantt chart served as project progress tracker to ensure the development process are going on smoothly with the given time scope. In simple words, the project’s WBS and Gantt chart are the scope and time guidelines that should be followed throughout the development process.

3.2.2 Analysis and Design

During the planning phase, the project objectives were determined, and data were collected through the methods mentioned in section 3.2.1.2. The data were analysed in this analysis and design phase. Through the analysis, information was formed together with the development of system requirements list and project scopes. Furthermore, the requirements were arranged according to the importance of the feature towards the system functionality. This helps to identify the features that should be implemented first in each increment. In addition, use case diagram was also produced to understand and visualise the interactions between end users and the system. Use case descriptions were also listed to understand each of the use case better, especially on the interaction between users & system and the steps of performing certain actions. In addition, the project scopes were also broken down into smaller tasks and related modules, each of the increment delivered a specific feature of the system.

Next, before begins with the incremental of development and testing, the incremental iterations were designed. For each increment, a specific module was assigned according to the list of prioritised requirements found in the analysis phase. The flows, user interfaces, algorithms and API of each increment were analysed. In term of user interfaces, Figma tool were used to develop the prototype designs to show the interfaces of the modules. In addition, interfaces flow chart was also designed to achieve a better understanding on system navigation. Furthermore, an entity relationship diagram was also developed to gain a basic understanding on attributes needed for each entity. Throughout the increments, the attributes required were reanalysed and refine. Thus, overall theme, navigation and layout design were prepared.

3.2.3 Design, Implementation and Testing

Before beginning with the development, environment required for the incremental development were prepared. Software chosen for this project was downloaded into the development computer, such as Visual Studio Code, Reat Native, Android Studio, Git and Node JS. In addition, extensions for JavaScript and React Native were installed during the project configuration. Besides, the necessary project dependencies were also installed at the same time, such as React Native dependencies, navigation library and React State Management.

The project continued with the incremental development, which each increment repeats the phases of design, implementation and testing of a single module of the system. The module chosen to be developed in the first increment cycle is the most important functionality of the system extracted from the system's requirements. The next increment was continued with next prioritised module, until all the modules are developed.

3.2.3.1 First Increment

In the first increment, the user authentication and authorisation module of the system were implemented. To develop this module, features such as account registration and user were implemented first. The structure and components of the module were designed. To store the user credentials, data model and session were designed using the database model. Next, the prototype of user interface that collects user's input for registration data and authorisation data were revisited and analysed for improvement. Furthermore, the backend logic of the account registration and login were designed or drafted.

Once the designing of the module was completed, the iteration continued with the development phase. Firstly, the database that stored user's information was created before the system was able to register user into the system. Next, the server-side logic processing for user authentication checking was developed using the backend development system. The API was developed for handling of the user registration, login, and password-forgotten operations. Once backend APIs were ready. The front-end user interfaces were developed using React-native according to the prototype. During the front-end development, the backend APIs were called, a seamless interaction between the front-end and back-end of the system successfully generated a functionable module.

After completion of the development process, the testing activities were conducted to ensure the functionalities of account registration and user login modules were working perfectly.

3.2.3.2 Second Increment

Most of the modules in the proposed BMS were working based on the registered bills. Thus, the module selected for the second increment was the bill registration module. During the design phase, the requirements for the bill registration module were analysed to design the attributes and parameters of the module. The billing companies were identified and grouped according to the types of service they provided. Besides, the information required by the billing companies to register the bill was also identified. Furthermore, the user interface design was revisited, including the input fields, validation rules, and user interaction flows. In addition, the data model used to store the registered bill, designed earlier, was revisited to finalise the attributes.

After the design phase, the implementation phase began with the backend development. The APIs that handled the logic of bill registration requests from the users to the billing companies were prepared. Besides, the registration information was validated to ensure the correctness of the format before being sent to the billing companies. Furthermore, the system backend also handled the responses from the billing companies. Next, the user interface was implemented for the bill registration module based on the prototype design. The necessary screens, forms, and validation for users' inputs were designed using React Native. To integrate and store the billing information provided by users and billing companies, a database table for bills was also created.

Unit testing was conducted to verify the functionality of the bill registration module. Besides, integration testing was performed to ensure that only authorised users were able to access their bills. In addition, user acceptance testing was conducted to gather feedback from bill payers and billing companies.

3.2.3.3 Third Increment

The module chosen to be developed during the third increment was the payment module, which was the main functionality of the system. The payment module was chosen to be developed during the third increment because it could only be tested when the user authentication and bill registration modules were ready. The payment methods and their requirements were analysed.

During the design phase, the types of payment methods were analysed to ensure a clear understanding of the payment processes, payment gateway integration, and the security requirements. Next, the overall logic flow of payment processing, including user interaction, payment gateway integration, and receipt showing, was also designed. The prototype for selecting bills for payment, designed using Figma, was revisited for refinement.

During the implementation, the backend development began with payment gateway integration. The backend APIs that called the payment gateway were developed to handle the parameter passing. The backend logic was implemented to handle payment requests, process transactions, authorize payments, handle redirects, and show receipts. Lastly, the frontend user interface was developed for users to select bills to pay, enter payment details, select a payment method, and view the receipt.

During the testing phase, end-to-end testing was carried out to ensure that payment transactions could be completed. Various payment scenarios were tested, including single bill payment, multiple bills payment, and pending payment. Besides, integration testing was also carried out to ensure the payment gateway could work seamlessly with the system.

3.2.3.4 Forth Increment

During the fourth increment, the bill reminder module was focused on. Throughout the review of similar applications, various bill reminder functionalities were analysed. Subsequently, the most suitable method was chosen for implementation.

The bill payment reminder module was implemented using React Native, where users are free to customize their preferred reminder timing and reminder medium. For example, the reminder timings included options such as the bill release date, a day before the due date, and when the total amount was about to exceed the credit limit provided. Besides, users could also select their preferred reminder medium, such as notification, email, or message.

After completing the development process, functionality testing was conducted to ensure that the reminder module was working as intended for all the payment timings and mediums.

3.2.3.5 Fifth Iteration

The module that was selected in this iteration was the bill analysis module. During the design phase, the algorithm for bill analysis was analysed and studied. The algorithm that was most effective and aligned with the objectives of the system was chosen. Additionally, the prototypes of the user interface that visualised the analysed results were developed.

During the development phase, the bill analysis algorithms were applied and integrated into the backend of the system. Conditions that triggered the bill analysis automatically initiated the bill analysis process, ensuring real-time updates of the bill analysis results. The billing data required for analysis was retrieved from the database. After the analysis, the information was generated based on the data. The user interfaces were developed to present the analysis results through graphical visualisations such as graphs and charts.

Following the implementation phase, the results and the accuracy of the bill analysis module were verified. The actual results and expected results were compared. Through comprehensive testing, the reliability and accuracy of the module were verified.

3.2.4 Closing

After finishing the iterations, all of the designed modules were developed and tested. During the closing phase, final testing was conducted to test the entire system and identify any remaining defects. Besides, quality assurance was performed to ensure the system met the specified requirements and expected quality. Additionally, usability testing was carried out to test the easiness to use of the application. Furthermore, API testing was conducted to ensure the system and server worked seamlessly together without causing bugs. After the internal checking, user acceptance testing (UAT) was carried out by having bill payers use the system to pay and manage their bills.

The feedback and concerns provided by the users were collected for further analysis. The system documentation was prepared to increase the readability and maintainability of the system. Furthermore, to implement the system with more billing companies, the deployment documentation was prepared to guide the cooperation and integration of the billing companies, especially their IT teams.

3.3 Development Tools

3.3.1 Figma

Figma is the design tool selected to create user interface prototypes for this project. Figma could be accessed online or through the downloaded desktop application. It is a cloud-based tool that allows real-time collaboration between multiple users on the same project. The projects were also saved in the cloud, enabling developer to access the projects across different devices, platforms, and accounts without the need for manual file sharing. In this project, Figma was used to create the user interface screens to guide the React Native view development. Besides, linking the interface screens together also eased the understanding of the interface flows by demonstrating the user mobile application interaction and navigation.

3.3.2 Visual Studio Code

Visual studio code is an integrated development environment (IDE) that allows the developer to write codes in various programming languages. In this project, it was used as the main IDE to implement the coding of React Native framework. Visual Studio Code provides a high efficiency and user-friendly environment for building mobile applications, it supports different programming languages and frameworks such as C#, JavaScript, HTML, CSS and more. Thus, if an integration of a second system's backend is needed, there is no need to switch to other IDE. Additionally, Visual Studio Code also provide developers with extensions for multiple frameworks and programming languages. Downloading the suitable extensions for React Native and JavaScript can assist the developer in code alignment and syntax error checking.

3.3.3 ReactJS

ReactJS is a JavaScript framework developed by Meta (previously called Facebook) used for building user interfaces of web applications. ReactJS is well known for its virtual DOM feature, which increase the efficiency of development process. Besides, ReactJS enables developers to create reusable UI components, this promotes standardisation, code reusability and maintainability of the system. In addition, ReactJS uses client-side rendering, by generating the user interface at the user's devices and update it on the user's web browser.

3.3.3.1 React Native

React Native is a framework built on top of ReactJS for native mobile applications development, React's syntax and behaviours mentioned in section 3.3.3 are also inherited by React Native. Mobile applications created by React Native are compatible for both iOS and Android devices. React Native allowed developers to use React and native UI components, which act similar to HTML and CSS in the web development. In addition, React Native provides built-in components, supports component reused and real-time rendering, these characteristics will speed up the overall development process.

3.3.4 Expo Go

Expo Go is an application that allows developers to build, deploy and test React Native application. With Expo Go, the developed application can be used in a physical mobile phone instead of a virtual device run by the computer. Thus, providing an environment to test the developed application. Expo Go support user interfaces, hardware access and database integration. Besides, Expo Go can be used to run the application on both actual Android and iOS devices.

3.3.5 NodeJS

NodeJS is a server-site JavaScript runtime environment to build backend services and APIs of a web application system. However, it the backend services and APIs can also interact seamlessly with mobile applications. NodeJS is chosen as the backend development tool to achieve single language stack as JavaScript is also used for the frontend development. The objective is to simplify the knowledge transfer between the frontend development and backend development. Besides, NodeJS is also suitable for handling input/output operations of a real-time applications due to its asynchronous and event-driven properties. Furthermore, NodeJS also comes with a rich open-source libraries that can be installed easily using node package manager, this helps to accelerate the development and avoid building components from scratch.

3.3.6 MongoDB and MongoDB Compass

MongoDB is a NoSQL database known for its flexibility, which the main reason of it becoming the choice for modern application development. MongoDB stores data in JSON-like documents (BSON format), this allows the server to build a more complex and hierarchical data structures in the database. Besides, the schema-less nature of MongoDB supports dynamic and evolving data models, which make it more flexible to future changes. The updates of database schema such as adding few more attributes in a collection will not affect its original functionality of the old data format. In addition, ability to handle large volumes of data transaction due to its horizontally scalability is also advantageous for BillHub application, which has growing data as more transactions be made in future.

MongoDB Compass is a software tool that works seamlessly with MongoDB, allowing developers to visualize and manage the data structure of collections and databases through a user interface. Developers can visualize the structure of collections and databases, which enhances understanding and improves the efficiency of development and debugging. Additionally, data can be managed easily by editing or deleting it manually using the functions in the user interface, which saves significant development time compared to managing data through command or APIs. Furthermore, it simplifies the process of creating test data.

3.3.7 Git and GitHub

GitHub is a web-based version control system for Git repositories that works similarly to Git. Due to its nature as a web-based cloud storage platform, it provides additional features such as issue tracking, project management tools, code review, and collaboration modes for teams that have more than 1 developers. In this project, GitHub is used to back up the source code and files of the system in the case of hardware damage or loss.

3.3.8 Postman

Postman is a API testing tool that enables developers to build, test and debug API's efficiently. Postman provides a user interface to send HTTP requests, and view the responses return by the API. Postman is used to simplifies the process of server development, by testing and maintaining APIs. On the other hand, Postman is also used to mimic the billing companies' http request as there are no actual billing companies come into collaboration in this project.

3.4 Work Breakdown Structure (WBS)

1.0 Project Initiation

- 1.1 Analysis problem background
- 1.2 Identify significance of project
- 1.3 List out the problems statement
- 1.4 Identify project goals and objectives
- 1.5 Propose project solution
  - 1.5.1 Research similar solutions
  - 1.5.2 Compare similar solutions
  - 1.5.3 Sketch flow of solutions
- 1.6 Propose project approach
  - 1.6.1 Research approaches
  - 1.6.2 Select development approach
- 1.7 Define project scope and its limitation
  - 1.7.1 Identify target users
  - 1.7.2 Identify required modules
    - 1.7.2.1 Analysis bills registration module
    - 1.7.2.2 Analysis bill retrieval module
    - 1.7.2.3 Analysis bill analysis module
    - 1.7.2.4 Analysis auto billing module
    - 1.7.2.5 Analysis bill reminder module
- 1.8 Gathering requirements
  - 1.8.1 Questionnaire Survey
    - 1.8.1.1 Select platform
    - 1.8.1.2 Design questions
    - 1.8.1.3 Review questions
    - 1.8.1.4 Distribute questionnaire
    - 1.8.1.5 Analyse collected data
  - 1.8.2 Review Similar Applications
    - 1.8.2.1 Research category of similar applications

1.8.2.2 Select application representing each category

1.8.2.3 Research and walkthrough of payment processes

1.8.2.4 Identify Pros and Cons of each application

1.8.2.5 Compare and tabulate result

1.9 Literature Review

1.9.1 Analyse evolution of bill payment solutions

1.9.1.1 Identify problems in each solution

1.9.1.2 Summarize the problems and solutions

1.9.2 Analysis of automatic bill payment

1.9.3 Compare frameworks

1.10 Requirements elicitation

1.10.1 Summarise selected features

1.10.2 Construct functional requirements and non-functional requirements

1.10.3 Review functional requirements and non-functional requirements

1.10.4 Refine functional requirements and non-functional requirements

1.10.5 Finalise the functional requirements and non-functional requirements

1.11 Project scheduling

1.11.1 Prepare work break down structure (WBS)

1.11.1.1 Identify the main activities

1.11.1.2 Break down the activities into smaller tasks

1.11.2 Prepare Gantt Chart

1.11.2.1 Determine task dependency

1.11.2.2 Estimate task duration

1.11.2.3 Estimate start date and end date

1.11.2.4 Draft gantt chart

1.11.2.5 Refine gantt chart

1.11.2.6 Finalise gantt chart

2.0 Analysis and design

2.1 Design use case diagram

2.2 Create use case descriptions

2.3 Design entity relationship diagram

2.4 Design interface flow diagram

2.5 Design prototypes

3.0 Set up project environment

3.1 Install necessary softwares

3.1.1 Install Visual Studio Code

3.1.2 Install Git

3.1.3 Install React Native CLI

3.1.4 Install Node.js and npm

3.1.5 Install Android Studio

3.2 Set up project configuration

3.2.1 Install extensions for JavaScript

3.2.2 Install extensions for React Native

3.2.3 Configure package settings

3.3 Install project dependencies

3.3.1 Install React Native dependencies

3.3.2 Install navigation library

3.3.3 Install state management

3.4 Verify project setup

3.Test project initialisation

3.4.2 Test version control functionality

4.0 Planning and Design phase

4.1 Design system architecture

4.1.1 Analyse proposed modules

4.1.2 Rearrange module due to priority

4.1.4 Determine module for each increment.

5.0 Incremental development

5.1 First Increment

5.1.1 Plan and design login module

5.1.1.1 Define data model for users

5.1.1.2 Plan backend logic

5.1.2 Create authorisation module

5.1.2.1 Create registration page user interface

        - 5.1.2.2 Create backend services for registration
        - 5.1.2.3 Create login page user interface
        - 5.1.2.4 Create backend logic for login
        - 5.1.2.3 Integrate frontend and backend
    - 5.1.3 Test authorisation module
        - 5.1.3.1 Test registration functionality
        - 5.1.3.2 Test login functionality
        - 5.1.3.3 Address defects and refine module
- 5.2 Second Increment
    - 5.2.1 Plan and design bill registration module
        - 5.2.1.1 Define requirements for bill registration
        - 5.2.1.2 Categorise billing companies types
        - 5.2.1.3 Specify the information required for registration
    - 5.2.2 Create bill registration module
        - 5.2.2.1 Create frontend bill registration forms
        - 5.2.2.2 Create backend services for bill registration
        - 5.2.2.3 Integrate frontend and backend
    - 5.2.3 Test bill registration module
        - 5.2.3.1 Test different bill types of registration
        - 5.2.3.2 Test data retrieval after registration
        - 5.2.3.3 Address defects and refine module
- 5.3 Third Increment
    - 5.3.1 Plan and design bill payment module
        - 5.3.1 Analyse payment methods and requirements
        - 5.3.2 Analyse payment gateways compatibility
        - 5.3.3 Design payment backend logic
    - 5.3.2 Create bill payment module
        - 5.3.2.1 Create APIs calling payment gateways
        - 5.3.2.2 Create user interface for bill payment steps
        - 5.3.2.3 Integrate frontend and backend
    - 5.3.3 Test bill payment module
        - 5.3.3.1 Test single bill payment
        - 5.3.3.2 Testing multiple bills payment

5.3.3.3 Address defects and refine module

5.4 Forth Increment

5.4.1 Plan and design bills reminder module

5.4.1.1 Determine suitable reminder methods and timing options

5.4.1.2 Design backend logic for reminder

5.4.2 Create bills reminder module

5.4.2.1 Create frontend interfaces to set up reminder

5.4.2.2 Create backend logic to handle reminders

5.4.2.3 Integrate frontend and backend

5.4.3 Test bill reminder module

5.4.3.1 Test notification reminder

5.4.3.2 Test email reminder

5.4.3.3 Test alarm reminder

5.5 Fifth Increment

5.5.1 Plan and design auto bill payment module

5.5.1.1 Analysis methods and timing for auto bill payments

5.5.1.2 Design the auto bill payment logic flow

5.5.2 Create auto bill payment module

5.5.2.1 Create frontend for set up auto-billing

5.5.2.2 Create backend APIs for scheduling auto payment

5.5.2.3 Integrate auto payment with bill payment module

5.5.3 Test auto bill payment module

5.5.3.1 Test credits auto bill payment

5.5.3.2 Test auto bill payment amount limiter

6.0 Closing

6.1 Final testing and quality assurance

6.1.1 Conduct final testing

6.1.2 Integration Testing

6.1.3 User Acceptance Testing

6.1.2 Conduct quality assurance

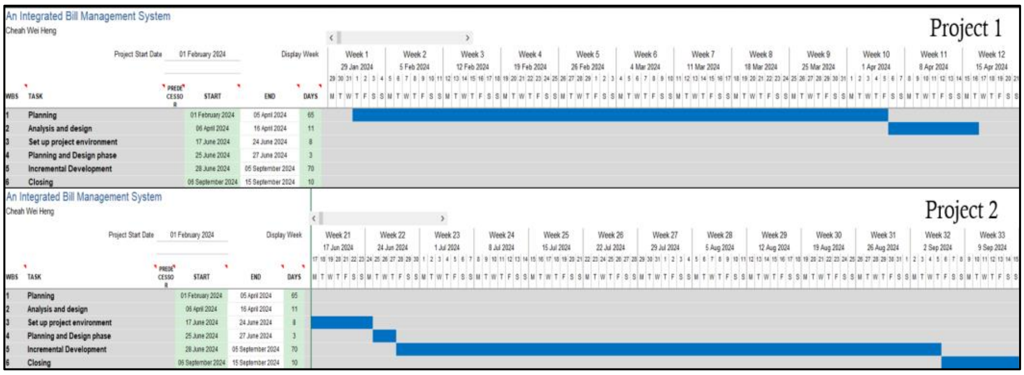
6.2 Finalise deliverables

6.3 Document lessons learnt

6.4 Project closure

3.5 Gantt Chart

3.5.1 Overview of the Project Timeline



An Integrated Bill Management System

Cheah Wei Heng

Project Start Date 01 February 2024

Display Week

WBS	TASK	PREDECESSOR	START	END	DAYS
1	Planning		01 February 2024	05 April 2024	65
2	Analysis and design		06 April 2024	16 April 2024	11
3	Set up project environment		17 June 2024	24 June 2024	8
4	Planning and Design phase		25 June 2024	27 June 2024	3
5	Incremental Development		28 June 2024	05 September 2024	70
6	Closing		06 September 2024	15 September 2024	10

An Integrated Bill Management System

Cheah Wei Heng

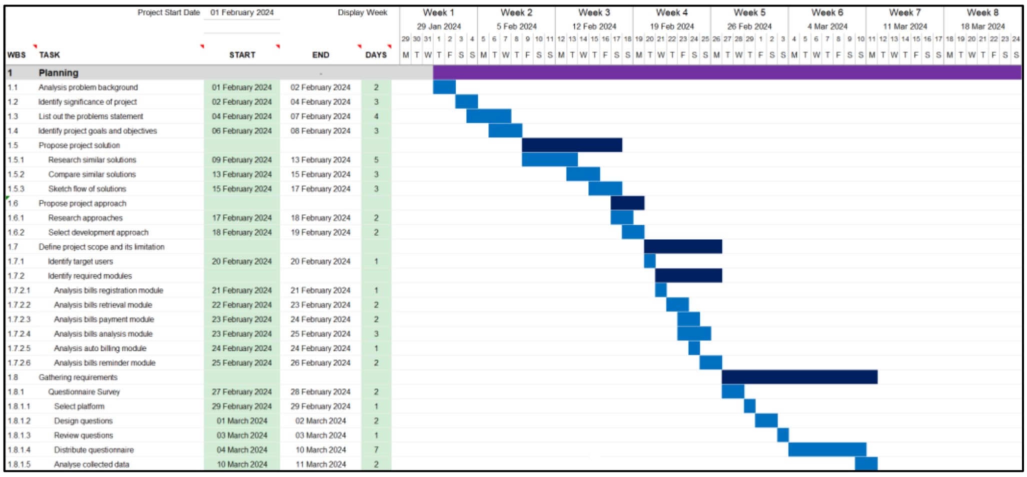
Project Start Date 01 February 2024

Display Week

WBS	TASK	PREDECESSOR	START	END	DAYS
1	Planning		01 February 2024	05 April 2024	65
2	Analysis and design		06 April 2024	16 April 2024	11
3	Set up project environment		17 June 2024	24 June 2024	8
4	Planning and Design phase		25 June 2024	27 June 2024	3
5	Incremental Development		28 June 2024	05 September 2024	70
6	Closing		06 September 2024	15 September 2024	10

Figure 3.2: Overview of project timeline

3.5.2 Planning Phase

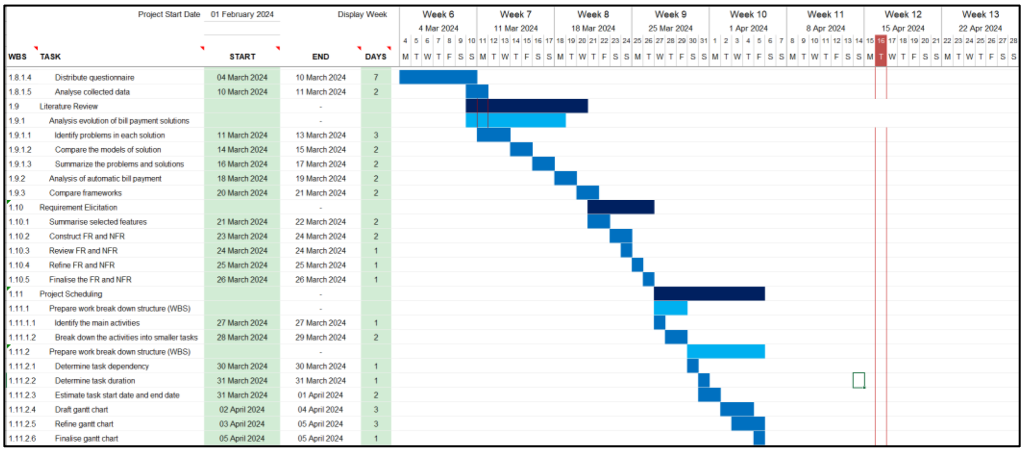


Project Start Date 01 February 2024

Display Week

WBS	TASK	START	END	DAYS
1	Planning		-	
1.1	Analysis problem background	01 February 2024	02 February 2024	2
1.2	Identify significance of project	02 February 2024	04 February 2024	3
1.3	List out the problems statement	04 February 2024	07 February 2024	4
1.4	Identify project goals and objectives	06 February 2024	08 February 2024	3
1.5	Propose project solution			
1.5.1	Research similar solutions	09 February 2024	13 February 2024	5
1.5.2	Compare similar solutions	13 February 2024	15 February 2024	3
1.5.3	Sketch flow of solutions	15 February 2024	17 February 2024	3
1.6	Propose project approach			
1.6.1	Research approaches	17 February 2024	18 February 2024	2
1.6.2	Select development approach	18 February 2024	19 February 2024	2
1.7	Define project scope and its limitation			
1.7.1	Identify target users	20 February 2024	20 February 2024	1
1.7.2	Identify required modules			
1.7.2.1	Analysis bills registration module	21 February 2024	21 February 2024	1
1.7.2.2	Analysis bills retrieval module	22 February 2024	23 February 2024	2
1.7.2.3	Analysis bills payment module	23 February 2024	24 February 2024	2
1.7.2.4	Analysis bills analysis module	23 February 2024	25 February 2024	3
1.7.2.5	Analysis auto billing module	24 February 2024	24 February 2024	1
1.7.2.6	Analysis bills reminder module	25 February 2024	26 February 2024	2
1.8	Gathering requirements			
1.8.1	Questionnaire Survey	27 February 2024	28 February 2024	2
1.8.1.1	Select platform	29 February 2024	29 February 2024	1
1.8.1.2	Design questions	01 March 2024	02 March 2024	2
1.8.1.3	Review questions	03 March 2024	03 March 2024	1
1.8.1.4	Distribute questionnaire	04 March 2024	10 March 2024	7
1.8.1.5	Analyse collected data	10 March 2024	11 March 2024	2

Figure 3.3: Planning phase timeline

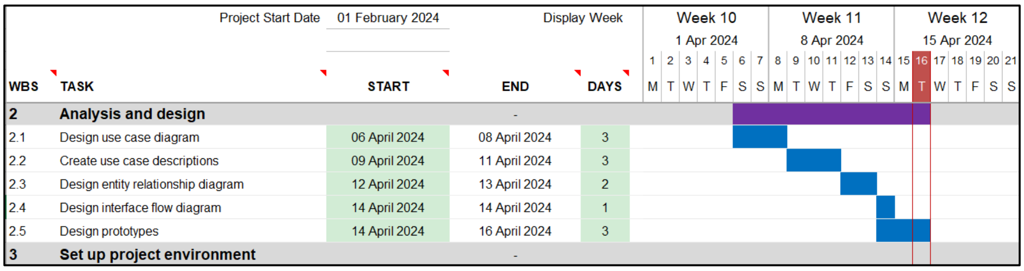


Project Start Date 01 February 2024 Display Week

WBS	TASK	START	END	DAYS
1.8.1.4	Distribute questionnaire	04 March 2024	10 March 2024	7
1.8.1.5	Analyse collected data	10 March 2024	11 March 2024	2
1.9	Literature Review		-	
1.9.1	Analysis evolution of bill payment solutions		-	
1.9.1.1	Identify problems in each solution	11 March 2024	13 March 2024	3
1.9.1.2	Compare the models of solution	14 March 2024	15 March 2024	2
1.9.1.3	Summarize the problems and solutions	16 March 2024	17 March 2024	2
1.9.2	Analysis of automatic bill payment	18 March 2024	19 March 2024	2
1.9.3	Compare frameworks	20 March 2024	21 March 2024	2
1.10	Requirement Elicitation		-	
1.10.1	Summarise selected features	21 March 2024	22 March 2024	2
1.10.2	Construct FR and NFR	23 March 2024	24 March 2024	2
1.10.3	Review FR and NFR	24 March 2024	24 March 2024	1
1.10.4	Refine FR and NFR	25 March 2024	25 March 2024	1
1.10.5	Finalise the FR and NFR	26 March 2024	26 March 2024	1
1.11	Project Scheduling		-	
1.11.1	Prepare work break down structure (WBS)		-	
1.11.1.1	Identify the main activities	27 March 2024	27 March 2024	1
1.11.1.2	Break down the activities into smaller tasks	28 March 2024	29 March 2024	2
1.11.2	Prepare work break down structure (WBS)		-	
1.11.2.1	Determine task dependency	30 March 2024	30 March 2024	1
1.11.2.2	Determine task duration	31 March 2024	31 March 2024	1
1.11.2.3	Estimate task start date and end date	31 March 2024	01 April 2024	2
1.11.2.4	Draft gantt chart	02 April 2024	04 April 2024	3
1.11.2.5	Refine gantt chart	03 April 2024	05 April 2024	3
1.11.2.6	Finalise gantt chart	05 April 2024	05 April 2024	1

Figure 3.4: Planning phase timeline (cont.)

3.5.3 Analysis and Design Phase

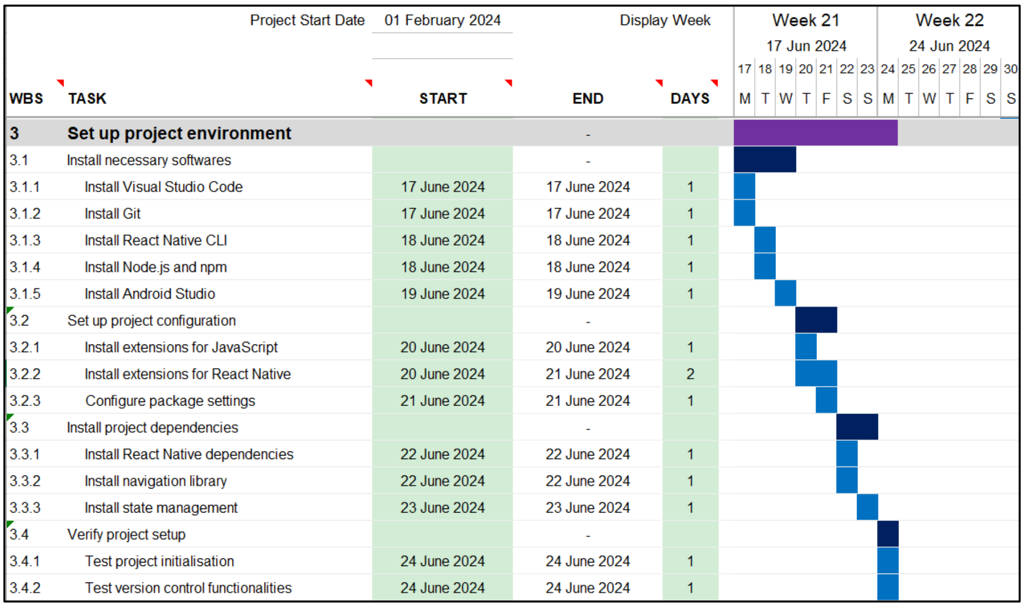


Project Start Date 01 February 2024 Display Week

WBS	TASK	START	END	DAYS
2	**Analysis and design**		-	
2.1	Design use case diagram	06 April 2024	08 April 2024	3
2.2	Create use case descriptions	09 April 2024	11 April 2024	3
2.3	Design entity relationship diagram	12 April 2024	13 April 2024	2
2.4	Design interface flow diagram	14 April 2024	14 April 2024	1
2.5	Design prototypes	14 April 2024	16 April 2024	3
3	**Set up project environment**		-	

Figure 3.5: Planning phase schedule

3.5.4 Set up project environment

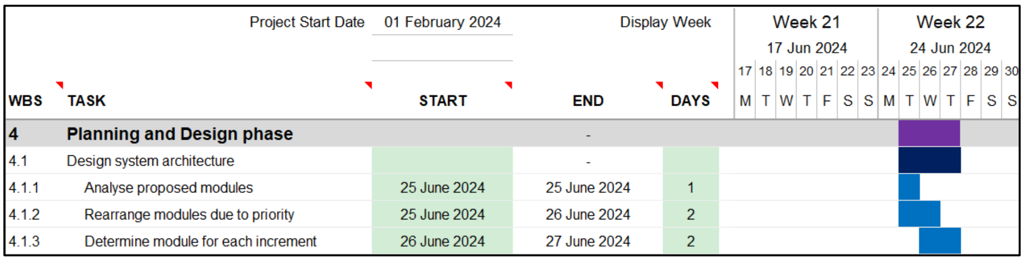


Project Start Date 01 February 2024 Display Week

WBS	TASK	START	END	DAYS
3	**Set up project environment**		-	
3.1	Install necessary softwares		-	
3.1.1	Install Visual Studio Code	17 June 2024	17 June 2024	1
3.1.2	Install Git	17 June 2024	17 June 2024	1
3.1.3	Install React Native CLI	18 June 2024	18 June 2024	1
3.1.4	Install Node.js and npm	18 June 2024	18 June 2024	1
3.1.5	Install Android Studio	19 June 2024	19 June 2024	1
3.2	Set up project configuration		-	
3.2.1	Install extensions for JavaScript	20 June 2024	20 June 2024	1
3.2.2	Install extensions for React Native	20 June 2024	21 June 2024	2
3.2.3	Configure package settings	21 June 2024	21 June 2024	1
3.3	Install project dependencies		-	
3.3.1	Install React Native dependencies	22 June 2024	22 June 2024	1
3.3.2	Install navigation library	22 June 2024	22 June 2024	1
3.3.3	Install state management	23 June 2024	23 June 2024	1
3.4	Verify project setup		-	
3.4.1	Test project initialisation	24 June 2024	24 June 2024	1
3.4.2	Test version control functionalities	24 June 2024	24 June 2024	1

Figure 3.6: Project environment set up schedule

3.5.5 Planning and design phase



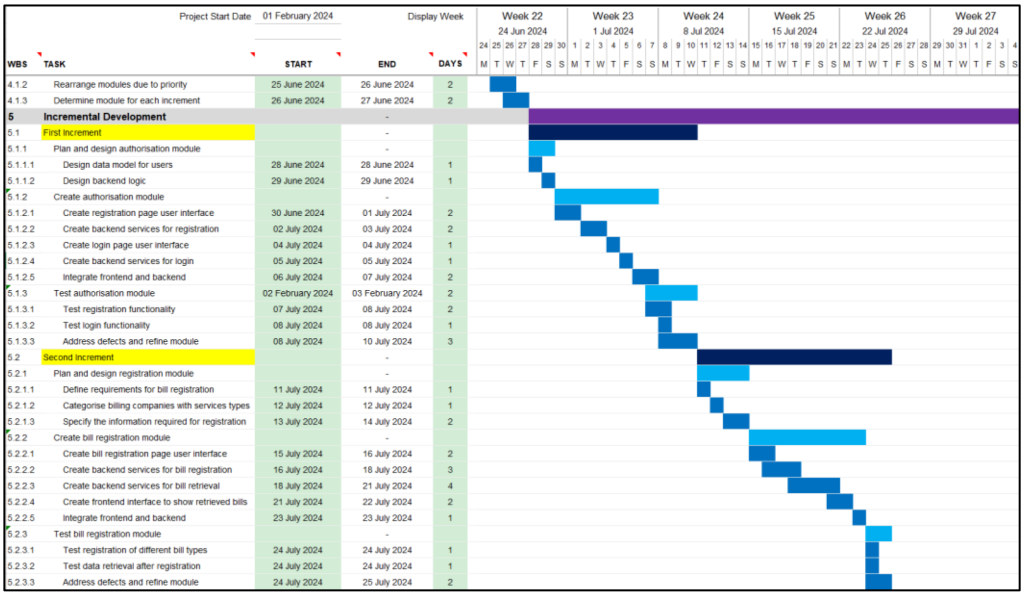
Project Start Date 01 February 2024 Display Week

					Week 21 17 Jun 2024							Week 22 24 Jun 2024						
					17	18	19	20	21	22	23	24	25	26	27	28	29	30
WBS	TASK	START	END	DAYS	M	T	W	T	F	S	S	M	T	W	T	F	S	S
4	**Planning and Design phase**		-										■	■	■			
4.1	Design system architecture		-										■	■	■			
4.1.1	Analyse proposed modules	25 June 2024	25 June 2024	1									■					
4.1.2	Rearrange modules due to priority	25 June 2024	26 June 2024	2									■	■				
4.1.3	Determine module for each increment	26 June 2024	27 June 2024	2										■	■			

Figure 3.7: Planning and design phase schedule

3.5.6 Incremental Development

3.5.6.1 First Increment and Second Increment

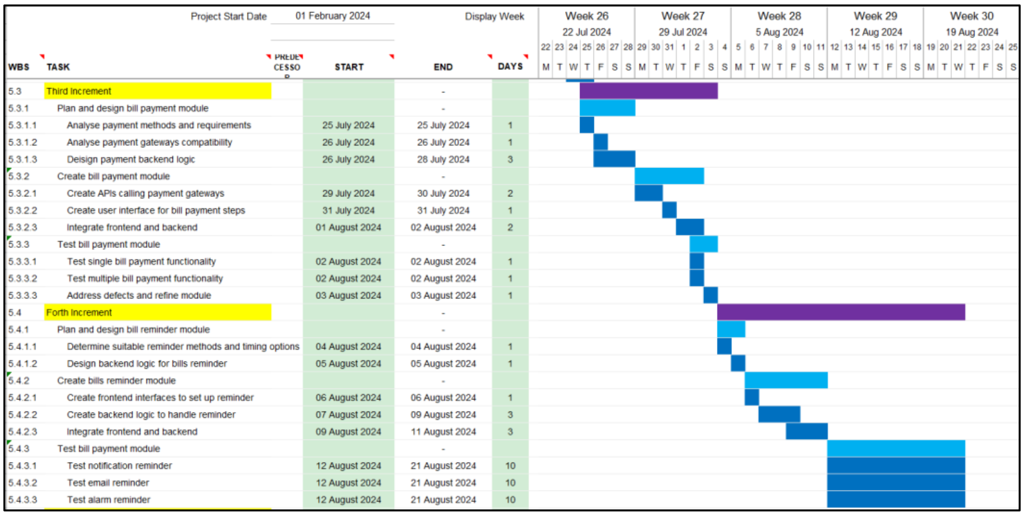


Project Start Date 01 February 2024 Display Week

WBS	TASK	START	END	DAYS
4.1.2	Rearrange modules due to priority	25 June 2024	26 June 2024	2
4.1.3	Determine module for each increment	26 June 2024	27 June 2024	2
5	**Incremental Development**		-	
5.1	First Increment		-	
5.1.1	Plan and design authorisation module		-	
5.1.1.1	Design data model for users	28 June 2024	28 June 2024	1
5.1.1.2	Design backend logic	29 June 2024	29 June 2024	1
5.1.2	Create authorisation module		-	
5.1.2.1	Create registration page user interface	30 June 2024	01 July 2024	2
5.1.2.2	Create backend services for registration	02 July 2024	03 July 2024	2
5.1.2.3	Create login page user interface	04 July 2024	04 July 2024	1
5.1.2.4	Create backend services for login	05 July 2024	05 July 2024	1
5.1.2.5	Integrate frontend and backend	06 July 2024	07 July 2024	2
5.1.3	Test authorisation module	02 February 2024	03 February 2024	2
5.1.3.1	Test registration functionality	07 July 2024	08 July 2024	2
5.1.3.2	Test login functionality	08 July 2024	08 July 2024	1
5.1.3.3	Address defects and refine module	08 July 2024	10 July 2024	3
5.2	Second Increment		-	
5.2.1	Plan and design registration module		-	
5.2.1.1	Define requirements for bill registration	11 July 2024	11 July 2024	1
5.2.1.2	Categorise billing companies with services types	12 July 2024	12 July 2024	1
5.2.1.3	Specify the information required for registration	13 July 2024	14 July 2024	2
5.2.2	Create bill registration module		-	
5.2.2.1	Create bill registration page user interface	15 July 2024	16 July 2024	2
5.2.2.2	Create backend services for bill registration	16 July 2024	18 July 2024	3
5.2.2.3	Create backend services for bill retrieval	18 July 2024	21 July 2024	4
5.2.2.4	Create frontend interface to show retrieved bills	21 July 2024	22 July 2024	2
5.2.2.5	Integrate frontend and backend	23 July 2024	23 July 2024	1
5.2.3	Test bill registration module		-	
5.2.3.1	Test registration of different bill types	24 July 2024	24 July 2024	1
5.2.3.2	Test data retrieval after registration	24 July 2024	24 July 2024	1
5.2.3.3	Address defects and refine module	24 July 2024	25 July 2024	2

Figure 3.8: First and second incremental schedule

3.5.6.2 Third Increment and Fourth Increment

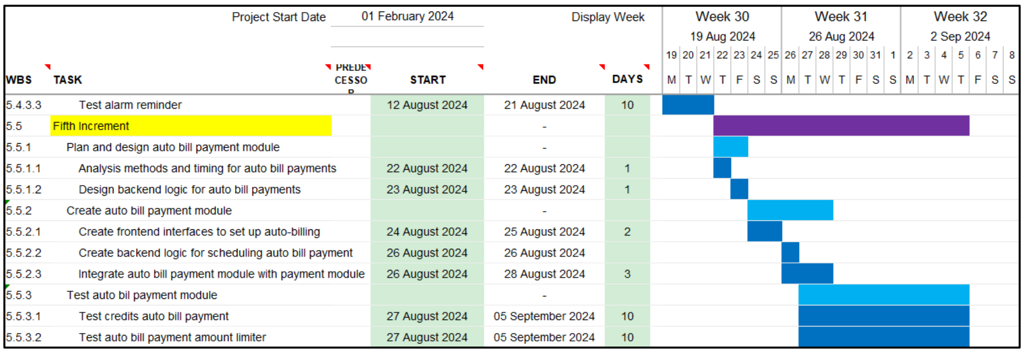


Project Start Date 01 February 2024 Display Week

WBS	TASK	PREDECESSOR	START	END	DAYS
5.3	Third Increment			-	
5.3.1	Plan and design bill payment module			-	
5.3.1.1	Analyse payment methods and requirements		25 July 2024	25 July 2024	1
5.3.1.2	Analyse payment gateways compatibility		26 July 2024	26 July 2024	1
5.3.1.3	Deisign payment backend logic		26 July 2024	28 July 2024	3
5.3.2	Create bill payment module			-	
5.3.2.1	Create APIs calling payment gateways		29 July 2024	30 July 2024	2
5.3.2.2	Create user interface for bill payment steps		31 July 2024	31 July 2024	1
5.3.2.3	Integrate frontend and backend		01 August 2024	02 August 2024	2
5.3.3	Test bill payment module			-	
5.3.3.1	Test single bill payment functionality		02 August 2024	02 August 2024	1
5.3.3.2	Test multiple bill payment functionality		02 August 2024	02 August 2024	1
5.3.3.3	Address defects and refine module		03 August 2024	03 August 2024	1
5.4	Forth Increment			-	
5.4.1	Plan and design bill reminder module			-	
5.4.1.1	Determine suitable reminder methods and timing options		04 August 2024	04 August 2024	1
5.4.1.2	Design backend logic for bills reminder		05 August 2024	05 August 2024	1
5.4.2	Create bills reminder module			-	
5.4.2.1	Create frontend interfaces to set up reminder		06 August 2024	06 August 2024	1
5.4.2.2	Create backend logic to handle reminder		07 August 2024	09 August 2024	3
5.4.2.3	Integrate frontend and backend		09 August 2024	11 August 2024	3
5.4.3	Test bill payment module			-	
5.4.3.1	Test notification reminder		12 August 2024	21 August 2024	10
5.4.3.2	Test email reminder		12 August 2024	21 August 2024	10
5.4.3.3	Test alarm reminder		12 August 2024	21 August 2024	10

Figure 3.9: Third and fourth incremental schedule

3.5.6.3 Fifth Increment



Project Start Date 01 February 2024 Display Week

Week 30 19 Aug 2024 | Week 31 26 Aug 2024 | Week 32 2 Sep 2024

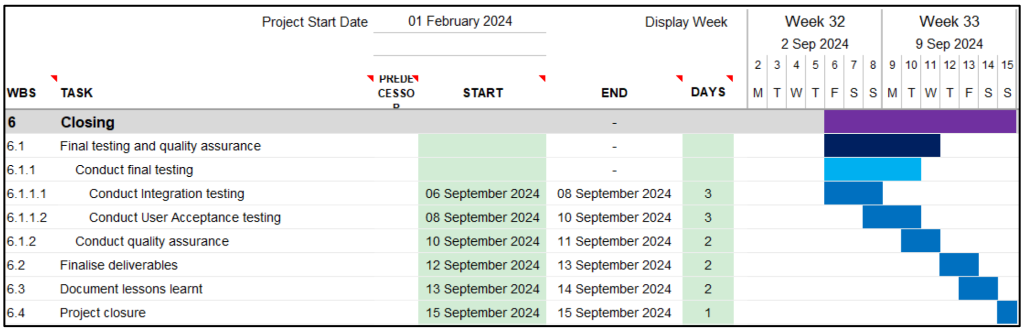
19 20 21 22 23 24 25 26 27 28 29 30 31 1 2 3 4 5 6 7 8

M T W T F S S M T W T F S S M T W T F S S

WBS	TASK	PREDECESSOR	START	END	DAYS
5.4.3.3	Test alarm reminder		12 August 2024	21 August 2024	10
5.5	Fifth Increment			-	
5.5.1	Plan and design auto bill payment module			-	
5.5.1.1	Analysis methods and timing for auto bill payments		22 August 2024	22 August 2024	1
5.5.1.2	Design backend logic for auto bill payments		23 August 2024	23 August 2024	1
5.5.2	Create auto bill payment module			-	
5.5.2.1	Create frontend interfaces to set up auto-billing		24 August 2024	25 August 2024	2
5.5.2.2	Create backend logic for scheduling auto bill payment		26 August 2024	26 August 2024	
5.5.2.3	Integrate auto bill payment module with payment module		26 August 2024	28 August 2024	3
5.5.3	Test auto bil payment module			-	
5.5.3.1	Test credits auto bill payment		27 August 2024	05 September 2024	10
5.5.3.2	Test auto bill payment amount limiter		27 August 2024	05 September 2024	10

Figure 3.10: Fifth increment schedule

3.5.7 Closing



Project Start Date 01 February 2024 Display Week

Week 32 2 Sep 2024 | Week 33 9 Sep 2024

2 3 4 5 6 7 8 9 10 11 12 13 14 15

M T W T F S S M T W T F S S

WBS	TASK	PREDECESSOR	START	END	DAYS
6	**Closing**			-	
6.1	Final testing and quality assurance			-	
6.1.1	Conduct final testing			-	
6.1.1.1	Conduct Integration testing		06 September 2024	08 September 2024	3
6.1.1.2	Conduct User Acceptance testing		08 September 2024	10 September 2024	3
6.1.2	Conduct quality assurance		10 September 2024	11 September 2024	2
6.2	Finalise deliverables		12 September 2024	13 September 2024	2
6.3	Document lessons learnt		13 September 2024	14 September 2024	2
6.4	Project closure		15 September 2024	15 September 2024	1

Figure 3.11: Closing schedule

CHAPTER 4

PROJECT SPECIFICATIONS

4.1 Introduction

In chapter 4, analysis of data collected through questionnaires are carried out. Through the analysis, requirements specification, use case diagram, interface flow diagram and user interfaces are built and shown under this chapter. Thus, an overall view of the system's functionalities and designs are shown.

4.2 Fact Finding

Online questionnaires are created through Google Forms. The google forms were distributed to collect information in order to understand the problems and requirements of the system. It is important to note that this project are focusing on solving problems regarding on bill payments in Malaysia, thus the questionnaires were sent to Malaysia citizen only. The questionnaires collected opinion from bill payers and also potential bill payers, under 2 different categories. This approach was chosen due to the properties of cost-effective and ability to obtain numerous public responses.

4.2.1 Responses on Google Form Questionnaires Survey

4.2.1.1 Personal Information

In this section, demographic information such as gender, age, marital status and education levels were collected. Besides, this section also identifies the respondent is a current bill payer or potential bill payer.

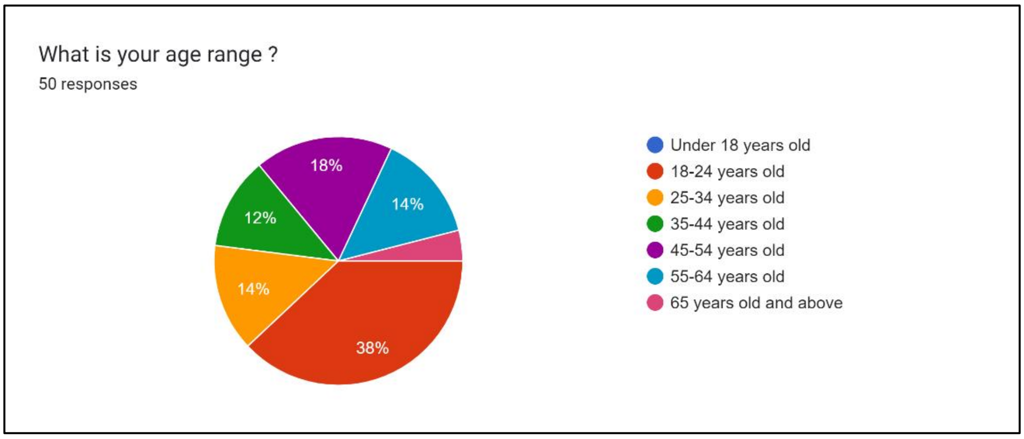


Figure 4.1: Age of respondents

The first question of the questionnaire collects the age of the respondents. This data is collected to investigate what is the relationship between the age of the respondents and their bill payment behaviour. As shown in figure 4.1, the age category with most respondents is 18-24 years old, comprising 38% of 50 respondents. Following this, age category 45-54 years old is contributing 18% of the responds with a total of 9 respondents. Next, age category of 25-34 years old and 55-64 years old are having same number of respondents, 7. Each of this age category is contributing 14% respectively. The category group with age 35-44 years old is represented by 6 respondents. Last category with respondent is age group 65 and above which consists only 1 respondent. However, there is no collected respond from respondents under 18-year-old. In short, this questionnaire is answered by respondents from different age groups above 18 years old.

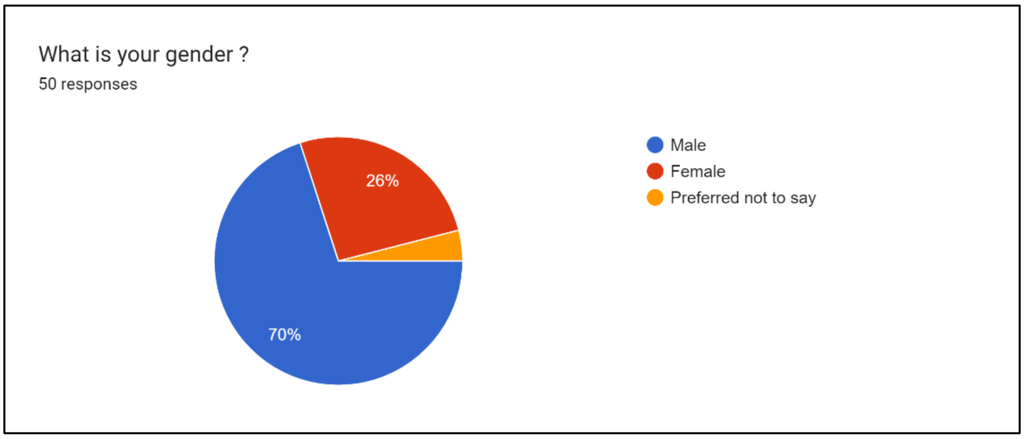


Figure 4.2: Gender of respondents

The second question collects the gender of the respondents. Based on the data analysis, there are 70% of male respondents and 26% of female respondents. However, there are 4% of the respondents preferred not to say.

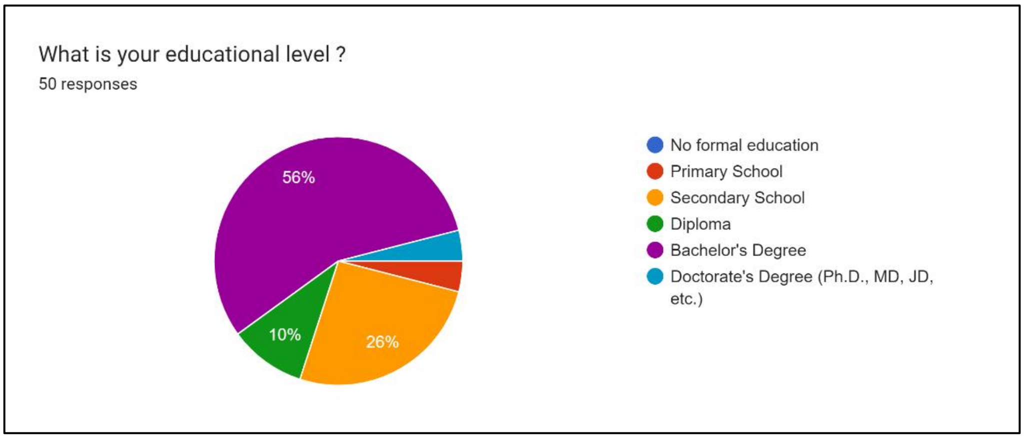


Figure 4.3: Educational Level of Respondents

The third question of the survey collects the educational level of the respondents. This section helps to identify potential problems faced by respondents with different educational level. The majority of the respondents have a bachelor’s degree, comprising 56% of the total 50 respondents. On the other hand, the minority group of the respondents are respondents who hold a Doctorate’s Degree and primary school education, which each only consists of 2 out of 50 respondents. Another two categories are 13 respondents with secondary school graduate and 5 Diploma holder respectively.

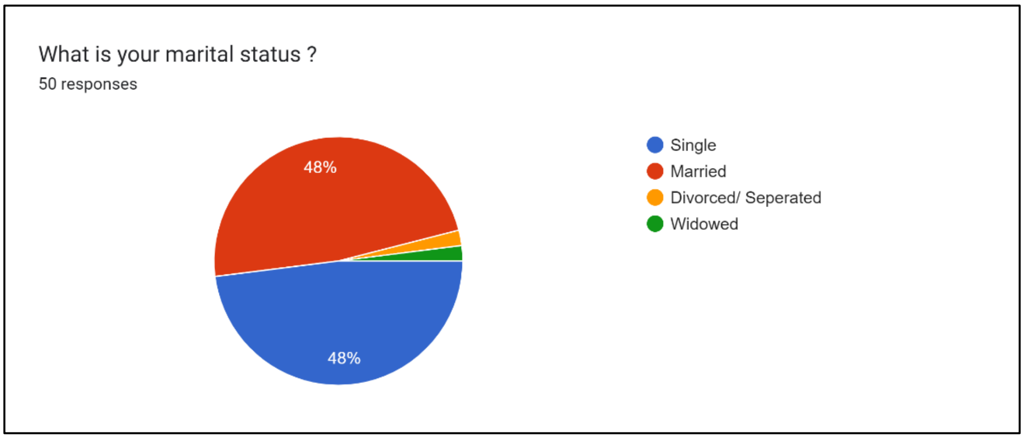


Figure 4.4: Marital Status of respondents

This question collects respondent's marital status to compare the bill paying behaviours and problems faced by single respondent and respondent with a family. The largest portion of respondent married and single, which are 48% of respondents respectively. Divorced and widowed categories encompass a smaller portion, which are 2 % of respondents respectively.

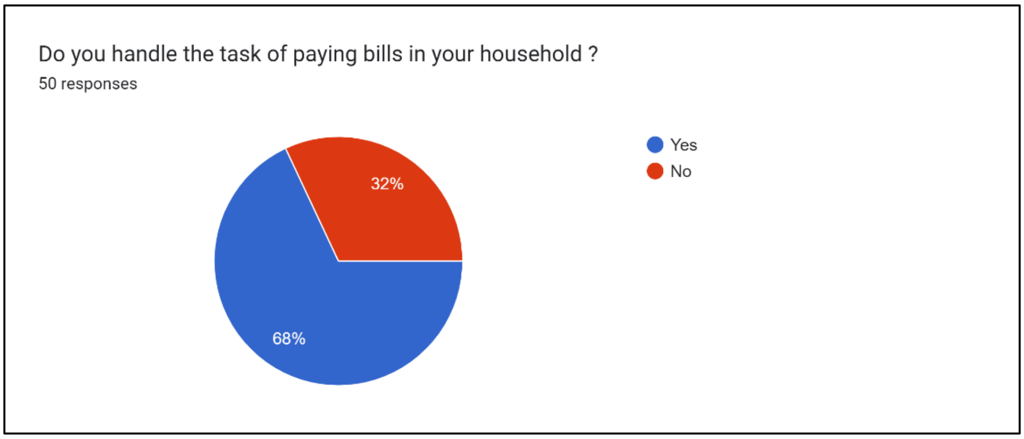


Figure 4.5: Proportion of respondents that handle the task of bill payment.

This question identifies if the respondent is currently a bill payer or a potential bill payer. There are 38 respondents (68% of the total respondents) are handling the task of paying bills in their household, while the rest 22 respondents (32% of the total respondents) having others to pay their bills. The respondents who are handling the task of paying bills are categorised as bill payers, while the rest are categorised as potential bill payers. This questionnaire has collected the experiences, problems that respondents are facing in bill payment and opinion from the bill payers. However, this questionnaire will only collect opinion-based question from the potential bill payers.

4.2.1.2 Collecting information regarding bill payment of respondents.

This section consists of questions that collect detail information about bill payment experiences of respondents, which are only answered by respondents that selects "Yes" in the last question of previous section.

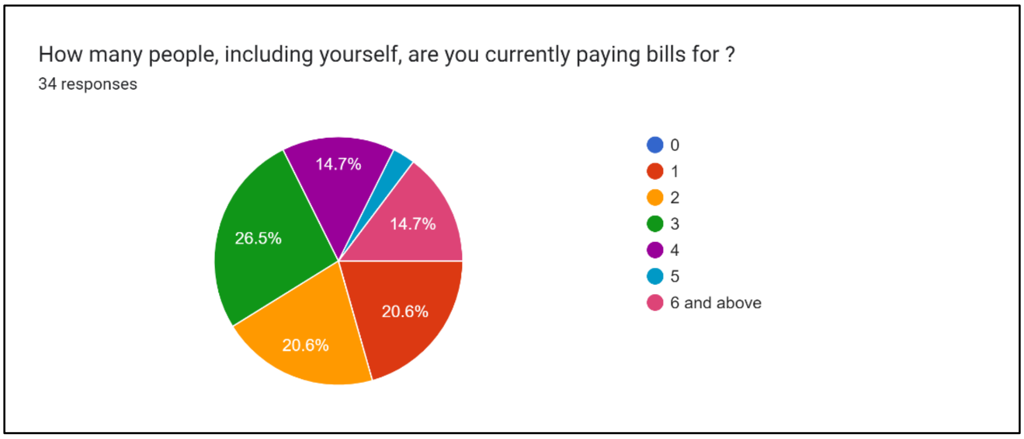


Figure 4.6: Number of people that respondents paying bills for

This question asks for the number of people that the respondent is paying bills for, including himself. The least number of people is one, as the respondent will at least pay for his own bill. There are 7 respondents (20.6%) who are fall under this category. This group of respondents are more likely a single working adult who is paying their own bill. The number of respondents that are paying for 2 people is also 7 (20.6%). The group with the most respondents is group of pay bill for three people, this group consists of 9 respondents (26.5). The smallest group in the result is respondent who pays bills for 5 people, this group only consist of 1 respondent (2%). The remaining respondents are 5 respondents (14.7%) in the group of paying bills for 4 people and group of paying bills for more than 6 people respectively.

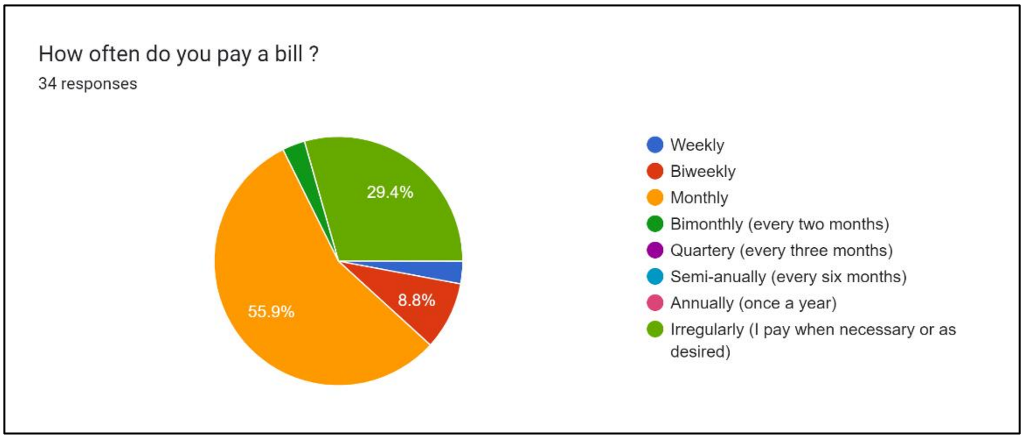


Figure 4.7: Frequency of bill payment of respondents

This question asks respondents how frequent they pay pill. The result shows majority of respondents, 19 respondents (55.9%) pay their bills monthly. Follows by 10 respondents (29.4%) that pay their bills irregularly, this group of respondents usually pay their bills when they see notifications or reminder. The next group of respondents are 3 respondents (8.8%) who pay bills biweekly, mean they will pay bills twice a month. The remaining respondents are 1 respondent (2%) who pay bills once per week and 1 respondent (2%) who pay bills bimonthly. There is no respondent who pay bills with a frequency longer than 2 months.

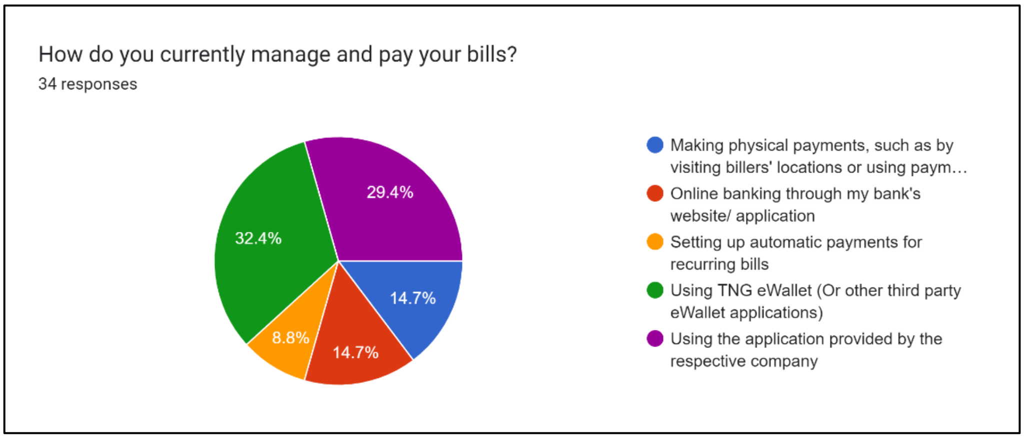


Figure 4.8: Current bill payment methods of respondents

This question asks respondents how they currently pay their bills. The result shows a majority group of 11 respondents (32.4%) are paying bills using TnG eWallet or other third party E-Wallet application such as Boost and GrabPay. The second large group consists of 10 respondents (29.4%) who are paying bills using the application provided by the billing companies respectively. Besides, there are same amount of 5 respondents (14.7%) under group of respondents paying bill physically at the store and group of respondents paying bills with online banking. From the collected data, respondents who use automatic bill payment make the smallest group of 3 (8.8%).

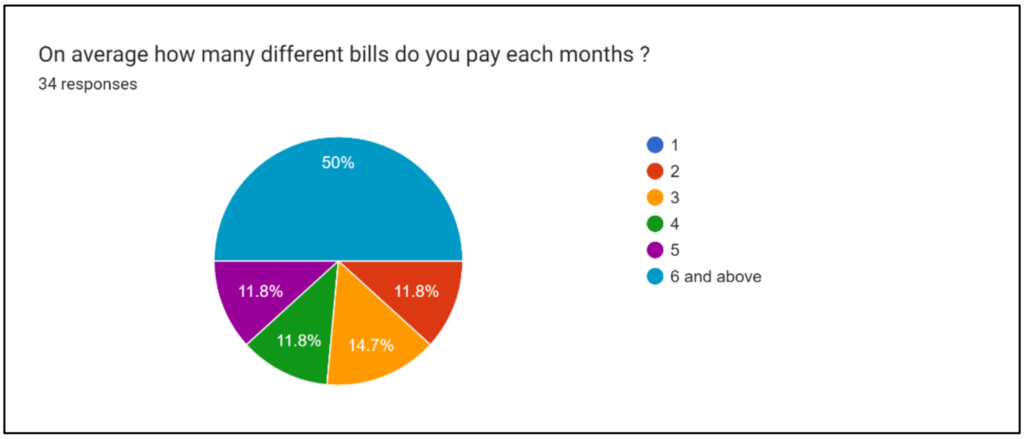


Figure 4.9: Average number of bills paid by respondents per month.

This question asks respondents about the average number of bills they paid monthly. The result shows that half of the respondents is paying more than 6 bills in a month. Follows by groups with 4 respondents (11.8%) paying 2,4 and 5 bills respectively. Besides, there is also a small group of 5 respondents (14.7%) who pay 3 bills in a month.

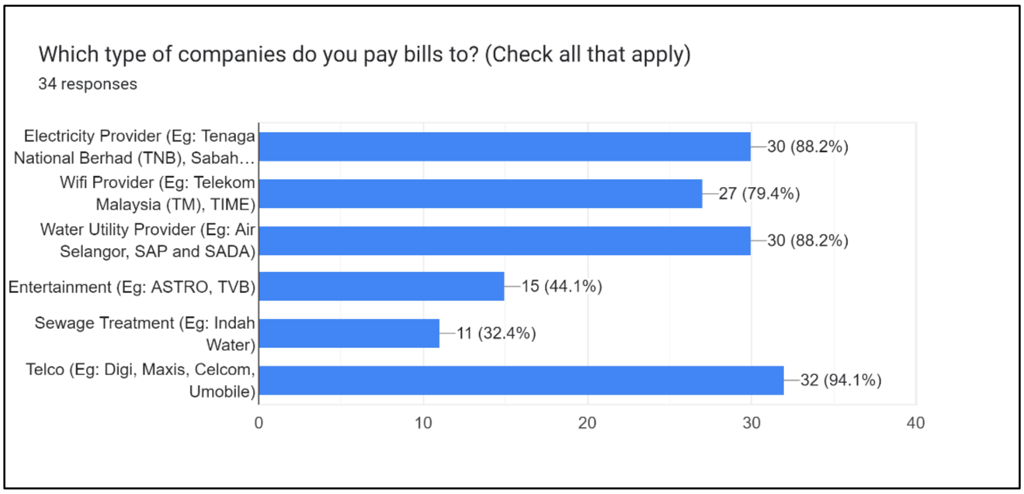


Figure 4.10: Types of company that respondents are paying bills to.

This question asks respondents type of services that they are paying bills to. The result shows there are 88.2% of respondents pay electrical bills and water bills, 79.4% of respondents pay internet bills. The majority, 94.1% of the respondents pay telco bills. On the other hand, the group of respondents who pay bills for entertainment and sewage treatment are relatively lesser. There are 44.1% of respondent who pay bills for entertainment and 32.4% of respondent who pay bill for sewage treatment.

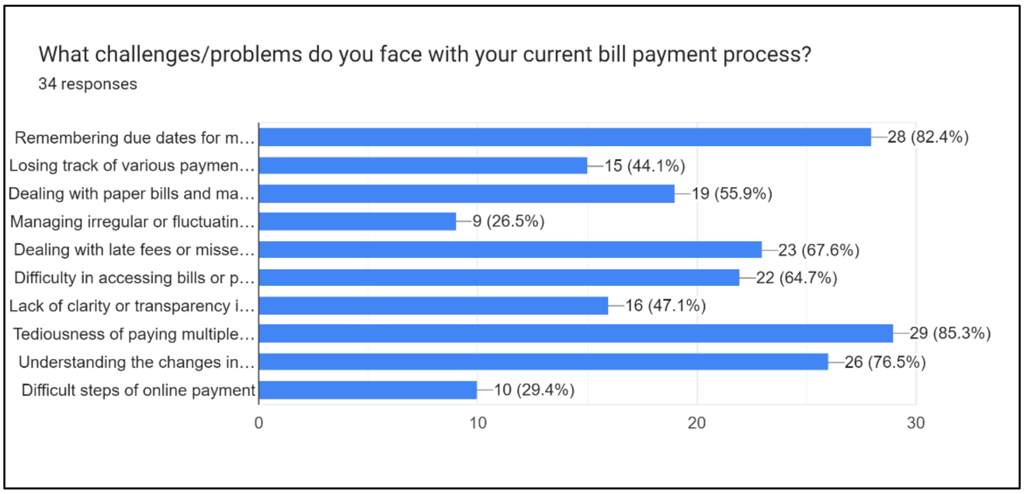


Figure 4.11: Problems faced by respondents in bill payments

This question asks respondents to tick the problems they are facing in bill payment. The result indicates the most significant problem is tediousness of paying multiple bills as 85.3% of the respondents are facing this problem. The rest of the problems arranged according to the severity are as follows:

-Remembering due dates for multiple bills (82.4%)

-Understanding the changes in the billing fees (76.5%)

-Dealing with late fees or missed payments (67.6%)

-Difficulty in accessing bills or payment information (64.7%)

-Dealing with paper bills and manual payment (55.9%)

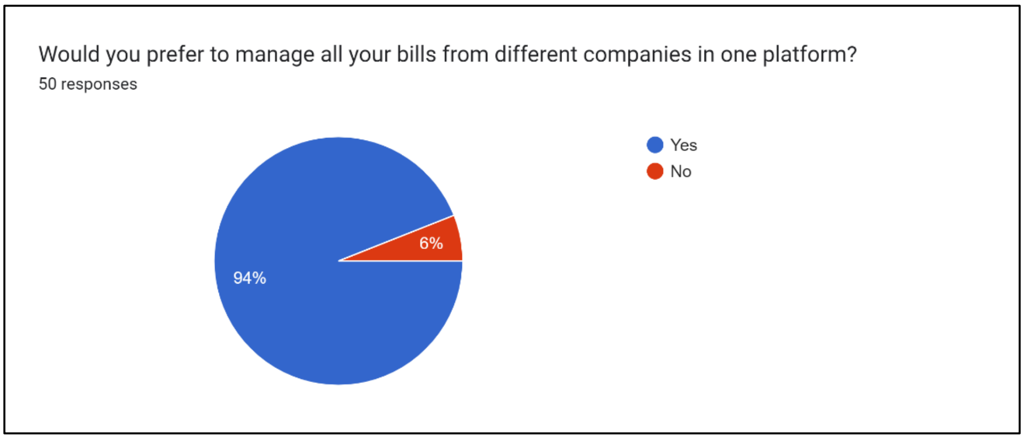
-Lack of clarity or transparency in billing statements (47.1%)

-Losing track of various payment methods and accounts (44.1%)

-Difficult steps of online payment (29.4%)

-Managing irregular or fluctuating bills (26.5%)

4.2.1.3 System Functionality Survey

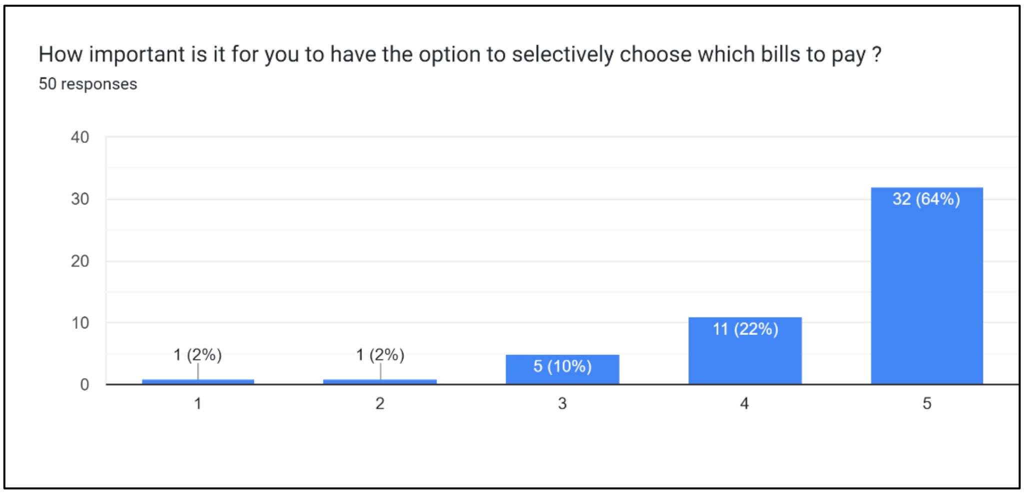


Would you prefer to manage all your bills from different companies in one platform?

50 responses

Figure 4.12: Preference of using consolidation model of BMS

This question asks respondents about their preference of using a single platform to manage and pay all their bills from various billing companies. Result shows that vast majority of 47 respondents (94%) prefer to use a BMS instead of using multiple applications to manage multiple bills. However, a minority of respondents (6%) do not prefer.



How important is it for you to have the option to selectively choose which bills to pay ?

50 responses

Figure 4. 13: Importance of selectively paying bills.

This question asks respondents how important it is for them to selectively pay their bills. This system offers a consolidation model, which allow bill payers to pay all their bills under a single transaction. However, the result shows that a majority group of 32 respondents (64%) found that pay bills selectively is very important. Follow by 11 respondents (22%) think it is quite important to selectively choose the bills to pay. At the neutral side, there is a smaller group of 5 respondents (10%) forming a middle group. The remaining respondents held a minority group, which are 1 respondent (2%) that think it is not so important and another 1 respondent (2%) that think it is not important at all.

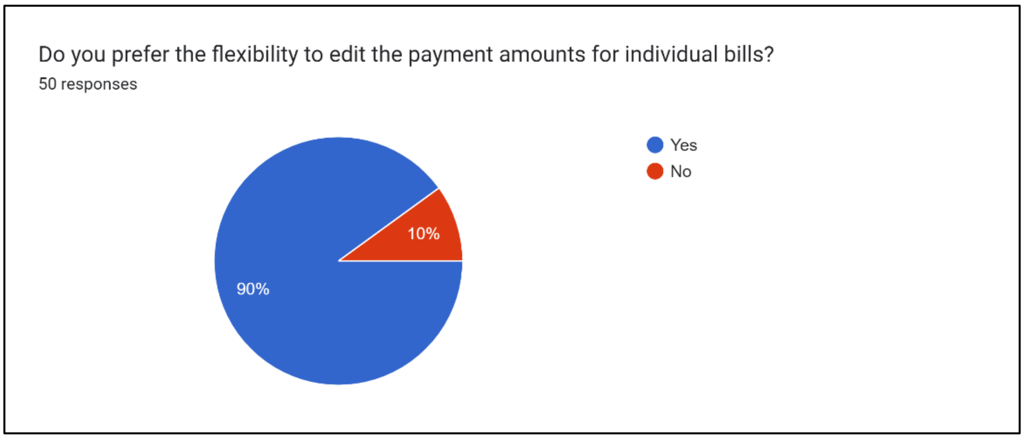


Figure 4.14: Preference to edit the payment amounts

This question asks respondents whether the total outstanding amount should be paid, or bill payer should be provided with flexibility to edit the payment amounts for each of the bill. The result clearly shows that a majority of 45 respondents (90%) prefer to have the flexibility to edit their payment amounts. However, there is a minor group of 5 respondents (10%) who think bill payers should just pay the entire outstanding amount and there is no need to edit the payment amounts.

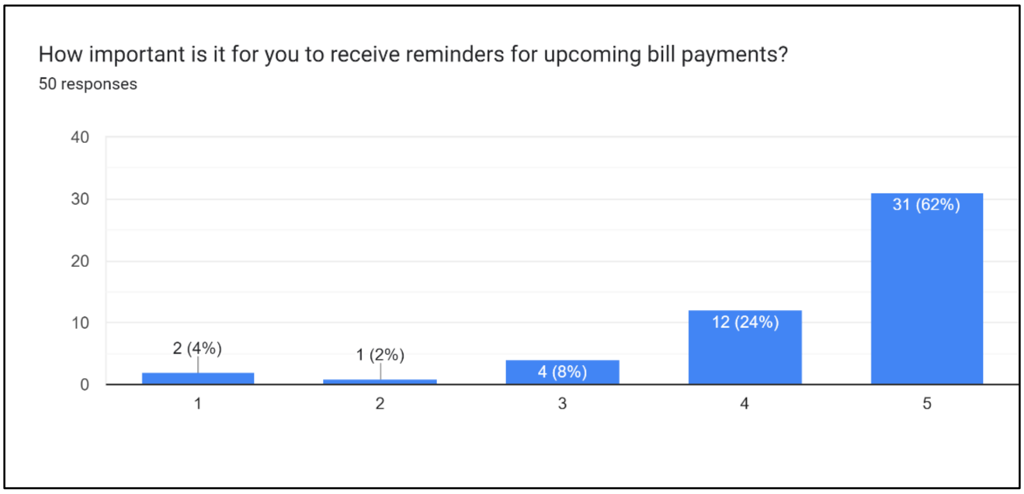


Figure 4.15: Importance of receive reminders

This question asks respondents about the importance of a bill reminder module for them. From the result, there are 31 respondents (62%) think it is very important, forming the majority group. Next, the second large group that consists of 12 respondents (24%) think it is quite important. At the neutral side, there are 4 respondents (8%) who are okay with or without the bill reminder. The remaining respondents are 1 respondent (2%) think bill reminder is not important and 2 respondents (4%) think it is not important at all.

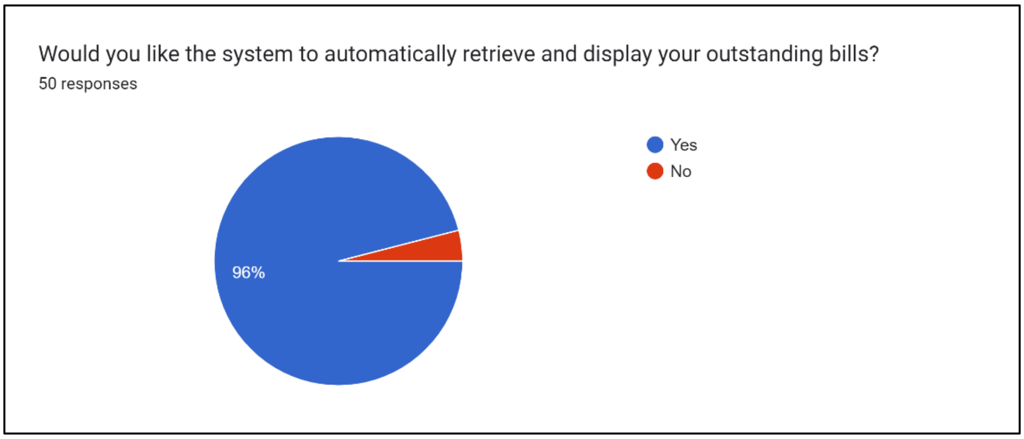


Figure 4.16: Preference of automatic bills retrieval

This question asks if the respondents if the system should automatically retrieve and show them the updates of the registered bills. From the result collected, 96% of the respondents are interested in automatic bills retrieval, thus they do not need to search for the bill manually. However, there are 2 respondents do not want the system to automatically retrieve and display the bills.

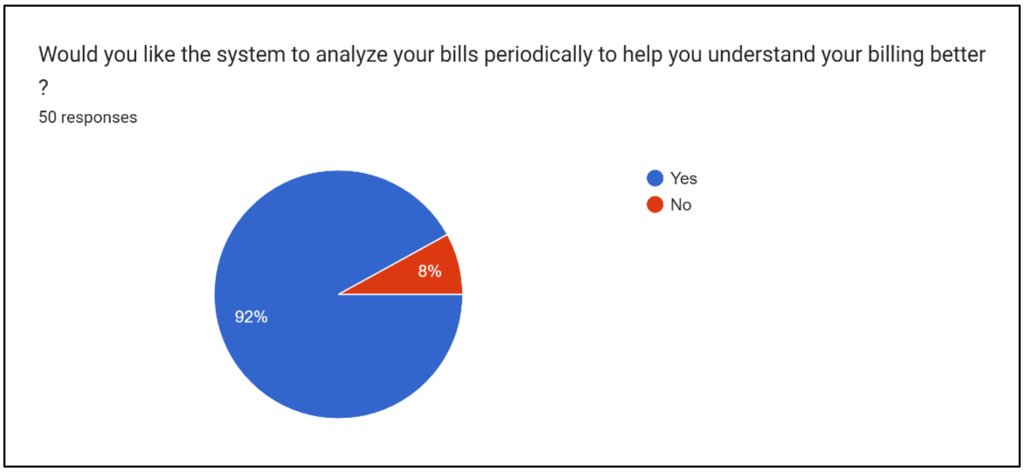


Figure 4.17: Preference of bill analysis

This question asks if respondents want the system to analyse the bills and provide help to understand their billing better. As the result, majority of the respondents are interested with the functionality of bill analysis. A vast majority of 92%, 46 respondents choose yes, while the remaining 8%, 4 respondents are not interested with the bill analysis functionalities.

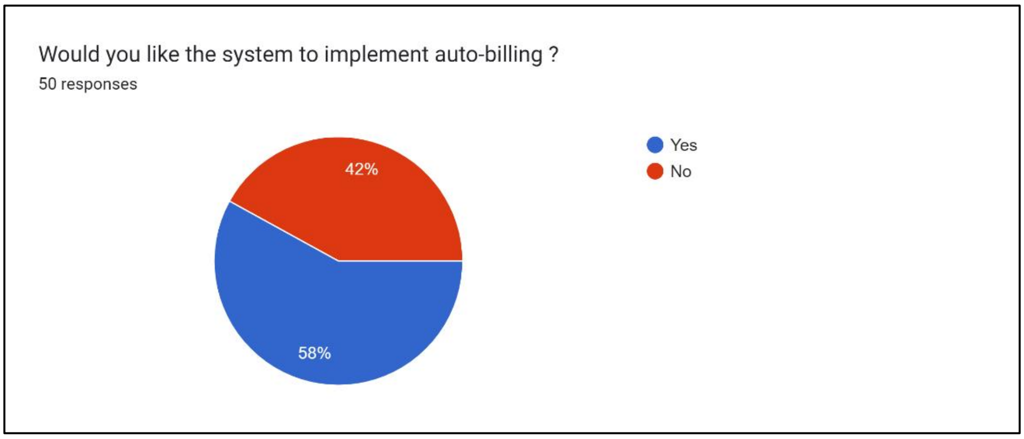


Figure 4.18: Preference of auto bill payment

This survey question asks if respondents would like the system to provide auto bill payment functionality. The result indicates that 42% of the respondents want the system to implement auto bill payment functionality, while 58% are not. In short, based on the result, a slight majority of the respondents are not willing to use auto-billing. Thus, the system should provide optional auto-billing for bill payers who want to utilize the feature.

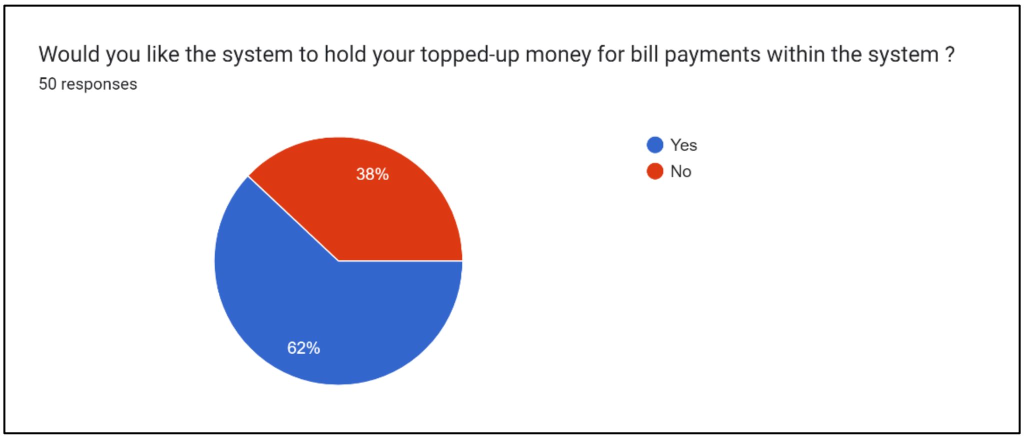


Figure 4.19: Preference of having credits and reloadability in system

This question collects respondents' opinions on their willingness to top up credits to the system for the purpose of bill payments in future. The survey results indicate that 62% of respondents are willing to top-up credits into the system for future bill payments, while the remaining 38% do not prefer to do so.

4.2 Requirement Specification

4.3.1 Functional Requirements

FR1: The system should allow bill payers to register accounts by collecting necessary information such as name, email and password.

FR2: The system should allow bill payers to login registered account by authenticating users based on their credentials.

FR3: The system should allow bill payers to register bills with the account numbers or phone numbers.

FR4: The system should allow bill payers to view a list of registered bills with bills information such as outstanding amount and payment deadline.

FR5: The system should allow bill payers to pay one bill or multiple bills under one single transaction.

FR6: The system should allow bill payers to apply for auto payment on their selected bills.

FR7: The system should remind bill payers of upcoming bills due dates.

FR8: The system should allow bill payers to view the analysis of bills issued and payments made.

FR9: The system should allow bill payers to view the details of their bills.

FR10: The system should allow bill payers to view billing and payment histories of bills.

FR11: The system should allow bill payers to top up credits for bill payment.

FR12: The system should allow bill payers to check notifications received.

FR13: The system should allow billing companies to send and update billing details.

4.3.2 Non-Functional Requirements

NFR1: Performance Requirements.

- The system should respond to user's interaction less than 1 second.

- The system should load the user’s registered bills with less than 3 seconds.
- The system should take at most 3 hours to complete the user’s transaction,

NFR2: Security Requirements

- The system shall authenticate users through validating their login credentials before login.
- The system shall only allow the authorised users to access the billing details of the user.

NFR3: Reliability

- The system should alert users and asked for confirmation when users perform destructive actions like removing registered bills or delete billing history.

NFR4: Availability

- The system down time should be no more than 1 hours in a day.

NFR5: Usability

- The user interface should be user-friendly and easy to use.
- When error occurs, the system should provide clear and understandable error messages.

NFR6: Maintainability

- The code of the system should be well-organized, easy to maintain by following the coding standards and best practices.

NFR7: Compatibility

- The system should be compatible with both Android and iOS mobile operating systems.

4.4 Use Case Modelling

4.4.1 Use Case Diagram

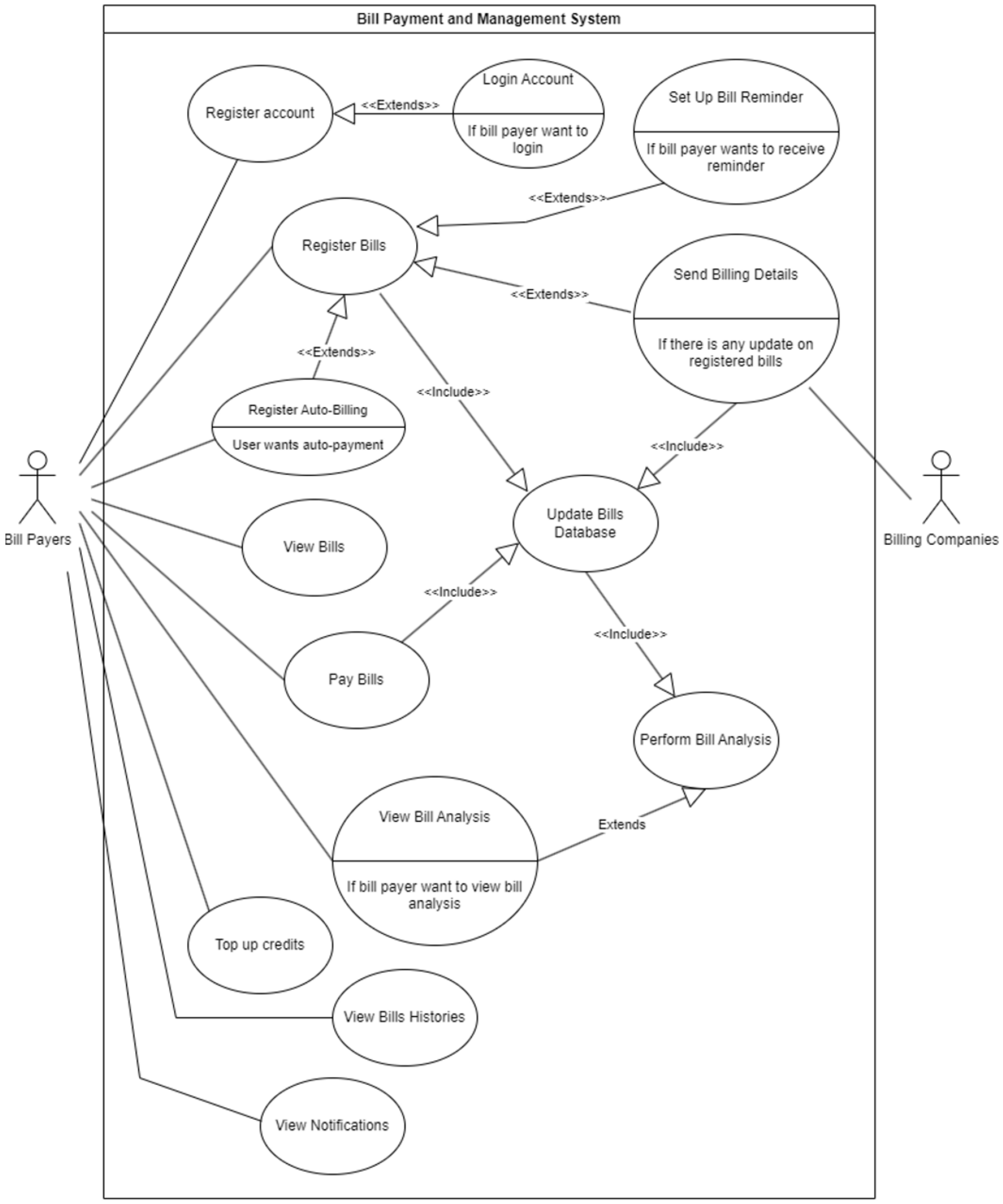


Figure 4.20: Use Case Diagram

4.4.2 Use Case Description

Remark: The “User” under this section is stand for “bill payer”.

Table 4.1: Use Case of Register Account

Use Case Name**:** Register Account	ID**: UC**1	Importance Level**: High**
Primary Actor**:** Bill Payer	Use Case Type**:** Detail, Real	
Stakeholders and Interests**:** Bill Payer– Create new account to access the bill management system.		
Brief Description： This use case allows users (bill payers) to register account. The details of the users are hold within the account. During registration, user has to register account by providing credentials information such as username, email, phone number and password.		
Trigger**:** A new bill payer wants to get started with the system for the first time.		
Relationships: Association **:** Bill Payer Include **:** - Extend **:** - Generalization**:**		

Normal Flow of Events**:**

1. The system prompt user to login or register account.
2. User chooses to register a new account.
3. The system will show user an account registration page for users to fill in the registration credentials.
4. User fills in the required information.
5. System will validate the format of the information to ensure its correct.
6. User submits the registration form.
7. System creates a new account for the user using the submitted information if the account not yet exist. Or else, perform S-1.
8. System redirects user to login page with a message of a successful registration.

Sub-flows**:**

S-1:

1. The system will show a message notifying the existent of the account.
2. The system will ask user to create account using another email/phone number or login to his existing account.

If user choose to create another account, perform UC1 again. If user choose to login, perform UC2.

Alternate/Exceptional Flows:

Table 4.2: Use Case of Login Account

Use Case Name**:** Login Account	ID**: UC**2	Importance Level**: High**

Primary Actor**:** Bill Payer	Use Case Type**:** Detail, Real
Stakeholders and Interests**:** Bill payer – Bill payers with registered account can login their account to pay and manage their bills.	
Brief Description**:** This use case describes how bill payers with registered account can login their account for accessing, managing and pay their bills.	
Trigger**:** When a bill payer wants to access his account for view, managing and pay his bill.	
Relationships: Association **:** Bill Payer Include **: -** Extend **: -** Generalization **: -**	
Normal Flow of Events**:** 1. The system shows bill payers with login page. 2. Bill payers choose to login by entering login credentials. 3. System validates the entered login credentials. 4. If login credential is valid, the system will login the user and redirect to home page and allowing the access. 5. If login credential is invalid, perform S-1.	

Sub-flows**:**

S-1:

1. If login credentials are wrong consecutively for 5 times, lock the account.
2. If user click the “forgot password”, perform S-1.1.
3. If user click the “register account”, redirect to use case ID1.

S-1.1:

1. Redirect user to “forgot password” module.
2. Prompt user to key in the credential information, without needling the password.
3. Select the method to verify and recover the account.
4. Prompt user to key in the one-time-password.
5. If the one-time-password is verified, redirect user to S-1.2, else if incorrect, prompt user to E-1.

Alternate/Exceptional Flows:

E-1:

1. Prompt user with helpful error message.
2. Provide user with customer service number/ email for help.

Table 4.3: Use Case of Register Bills

Use Case Name**:** Register Bills	ID**: UC3**	Importance Level**: High**
Primary Actor**:** Bill Payer	Use Case Type**:** Detail, Real	

Stakeholders and Interests**:**

Bill payer – Bill payers register their bills to their account through the system so that the system can retrieve the billing information.

Billing companies – Provide and send bill details whenever there is an update of the registered bill.

Brief Description**:** This use case allows bill payers to register multiple bills from various billing companies into the system for centralized management and payment.

Trigger**:** When a bill payer wants to manage and pay the bill using the system.

Relationships:

Association	**:** Bill Payer
Include	**:** Update Bills Database
Extend	**:** Send billing details
Generalization	**:-**

Normal Flow of Events**:**

1. Bill payer selects the “register bill” option from the home page.
2. Bill payer selects the billing companies.
3. Bill payer enters the bill account number, phone number or the required information.
4. System validates the entered data and saves the bill details in database. If validation failed, perform A-1.
5. Bill payer receives confirmation message of successful registration, the bill details are shown at home page.

Sub-flows**:**
Alternate/Exceptional Flows: A-1: 1. Show bill payer with informative error message. 2. If bill is already registered, inform the bill payer and end the bill registration process. 3. If bill is rejected for registration, ask bill payer to check their email/ contact the billing companies.

Table 4.4: Use Case of View Bill

Use Case Name**:** View Bills	ID**: UC4**	Importance Level**: High**
Primary Actor**:** Bill Payer	Use Case Type**:** Detail, Real	
Stakeholders and Interests**:** Bill payer – Bill payers view their billing information to understand their bills.		
Brief Description**:** This use case allows bill payers to check the existence of their registered bills and show them the details.		
Trigger**:** Bill payers want to view their billing information.		

Relationships: Association **:** Bill Payer Include **: -** Extend **: -** Generalization **: -**
Normal Flow of Events: -Bill payers login their account -The Bill database will be checked -The billing information will be shown at the home page if there is registered bills.

Table 4.5: Use Case of Pay Bills

Use Case Name**:** Pay Bills	ID**: UC5**	Importance Level**: High**
Primary Actor**:** Bill Payer	Use Case Type**:** Detail, Real	
Stakeholders and Interests**:** Bill payer – Bill payers pay their registered bills through the system. Billing companies – Receive payment and update bill payer’s billing account.		
Brief Description**:** This use case allows bill payers to pay 1 single bill or multiple bills in one single transaction.		

Trigger**:** Billing companies issued bills to bill payers and bill payers make payments.

Relationships:

Association	**:** Bill Payer
Include	**:** Update Bills Database
Extend	**:-**
Generalization	**:-**

Normal Flow of Events**:**

1. Bill payer selects the bills by ticking the checkbox of the bills from the bills list.
2. If bill payer selects “edit”, proceed to S-1.
3. Bill payer clicks “Pay”.
4. Bill payer is shown with the bill’s payment confirmation page.
5. Bill payer has to select the payment method and press “Pay” again.
6. Bill payer is redirected to payment gateway of selected payment method.
7. Receipt is shown after the successful transaction. Perform UC6.
8. If transaction is failed, perform A-1.

Sub-flows**:**

S-1:

1. Bill payer can modify the amount to pay.
2. Bill payer clicks “confirm”.

3. Go back to step 3 of normal flow.
Alternate/Exceptional Flows: A-1: 1. The system will show the error message. 2. Bill payer can selects “Cancel” or “Try Again”. 3. If bill payer selects “Try Again”, perform step 4 of normal flow. If bill payer select “Cancel”, redirect to home page.

Table 4.6: Use Case of Update Bill Database

Use Case Name**:** Update Bill Database	ID**: UC6**	Importance Level**: High**
Primary Actor**:** System	Use Case Type**:** Detail, Real	
Stakeholders and Interests**:** Bill payer – Bill payers pay their registered bills through the system. System – Store the bills payment history and bills details received from billing companies in database. Billing companies – Receive and approve the payments.		
Brief Description**:** This use case involves the system in updating the record of bill payments and bills update in the database for future references and bills analysis.		

Trigger**:**

- Bill payer registers a bill.

- Bill payer makes payment, and the payment is successful.

- Billing company sends update of the bills.

Relationships:

Association	**:** Bill Payer, Billing Company
Include	**:** Bill Analysis
Extend	**: -**
Generalization	**: -**

Normal Flow of Events**:**

1. When bill is registered, perform S-1.
2. When bill is paid by bill payer, perform S-2.
3. When billing company send updates of the bill, perform S-3.

Sub-flows**:** S-1: 1. The bill is assigned with a bill ID and store in database. 2. The bill details such as outstanding amount and bill deadline is stored. 3. Perform UC8. S-2: 1. The bill payment history is recorded, according to the bill ID of the bills paid. S-3: 1. The bill's status and details are updated corresponding to the information sent by billing companies. 2. The billing details shown at home screen are updated. 3. Updating of billing information will trigger UC8.
Alternate/Exceptional Flows:

Table 4.7: Send Billing Details

Use Case Name**:** Send Billing Details	ID**: UC7**	Importance Level**: High**
Primary Actor**:** Billing Companies	Use Case Type**:** Detail, Real	

Stakeholders and Interests**:**

Bill payer – Bill payers register bills.

Billing companies – Send billing details of bills that are registered.

Brief Description**:** This use case involves the sending of update regarding the registered billing account to the bill payers.

Trigger**:**

- Bill payer registers a bill.

- Billing company generate a new bill of the registered billing account.

Relationships:

Association	**:** Bill Payer, Billing Company
Include	**:** Update Bill Database
Extend	**:** Register Bill
Generalization	**: -**

Normal Flow of Events**:**

1. A bill payer registers a bill under a billing company through the system.
2. The billing company generate a bill for the registered billing account.
3. The billing details are sent to the system by the billing company.
4. The system will receive the billing information and perform UC6.

Sub-flows**:**
Alternate/Exceptional Flows:

Table 4.8: Use Case of Bills Analysis

Use Case Name**: Bill Analysis**	ID**: UC8**	Importance Level**: High**
Primary Actor**:** System	Use Case Type**:** Detail, Real	
Stakeholders and Interests**:** Bill payer – Bill payers register bills and pay bills. Billing companies – Send billing details of bills that are registered.		
Brief Description**:** This use case involves the analysis of bills using the details such as outstanding amount and amount paid each month.		
Trigger**:** Changes/ Update in bill database of the system.		
Relationships:		

Association **:** Bill Payer, Billing Company Include **: -** Extend **: -** Generalization **: -**
Normal Flow of Events**:** 1. There is an update or change performed on bills database of the system. 2. The system will rerun the logic of bills analysis module. 3. User may view the result of bill analysis by performing UC9 (View Bill Analysis).
Sub-flows**:**
Alternate/Exceptional Flows:

Table 4.9: Use Case of View Bills Analysis

Use Case Name**: View Bill Analysis**	ID**: UC9**	Importance Level**: High**
Primary Actor**:** Bill Payer	Use Case Type**:** Detail, Real	
Stakeholders and Interests**:** Bill payer – View the bills analysis done by the system.		

Brief Description: This use case involves function that allows bill payer to view bills analysis performed by the system to understand his bills.

Trigger: Bill payer wants to understand more about his billing details.

Relationships:

Association	: Bill Payer
Include	: -
Extend	: Bill Analysis
Generalization	: -

Normal Flow of Events:

1. Bill payer clicks the “Bill Analysis” button.
2. System will redirect bill payer to the “Bill Analysis” module.
3. Different types of “bill analysis” are shown.

Sub-flows:

Alternate/Exceptional Flows:

Table 4.10: Use Case of Register Auto Payment

Use Case Name**: Register Auto Payment**	ID**: UC10**	Importance Level**: High**
Primary Actor**:** Bill Payer	Use Case Type**:** Detail, Real	
Stakeholders and Interests**:** Bill payer – Bill payer chooses to automate a registered bill.		
Brief Description**:** This use case involves the automation of bill payment on selected bills. The bills with auto payment registered no longer need manual bill payment from the bill payer.		
Trigger**:** Bill payer wants to automate the bill payment of a billing account.		
Relationships: Association **:** Bill Payer Include **:** - Extend **:** - Generalization **: -**		

Normal Flow of Events**:**

1. A bill payer clicks “Apply Auto Payment.”
2. The system displays the registered bills that are available for automatic bill payment.
3. Bill payers select a bill.
4. Bill payer setup the automatic bill payment, proceed to S-1.
5. Failure in setting up the automatic bill payment, proceed to A-1.

Sub-flows**:**

S-1:

1. Bill payer selects the automatic bill payment mode, either fixed amount or outstanding amount. If bill payer selects outstanding amount, set a maximum cap.
2. Bill payer selects the payment frequency type, either fixed frequency or billing dates updated by the billing companies.
3. Bill payer selects the payment methods for auto bill payment.
4. Verify the payment methods.
5. Successful auto bill payment registration message is shown.
6. If bill payer wants to register auto bill payment for next bill, perform again step 1 of normal flow.

Alternate/Exceptional Flows:

A-1:

1. System shows informative error message to bill payer.
2. System provides suggestion of solution.
3. System provides options “Cancel” and “Try Again”.
4. If bill payer selects “Try Again”, repeat step 4 of normal flow. Or else, redirect to home page.

Table 4.11: Use Case of Top Up Credit

Use Case Name**: Top Up Credits**	ID**: UC11**	Importance Level**: Medium**
Primary Actor**:** Bill Payer	Use Case Type**:** Detail, Real	
Stakeholders and Interests**:** Bill payer – Bill payer needs to reload credits to their account for bill payments.		
Brief Description**:** This use case describes the ability for bill payer to reload credits of their accounts.		
Trigger**:** Credits are insufficient and bill payer wants to reload.		
Relationships: Association **:** Bill Payer Include **:** - Extend **:** - Generalization **: -**		

Normal Flow of Events**:**

1. Bill payer clicks “Reload” button.
2. Bill payer selects or enters the desire reload amount.
3. Bill payer selects a payment method for reloading and clicks “confirm”.
4. System will navigate bill payer to the payment gateway of selected payment method.
5. If the transaction is successful, receipt will be generated and send to the bill payer. If transaction is failed, perform A-1.
6. Bill payer will receive confirmation message with the updates in account’s credits.

Sub-flows**:**

Alternate/Exceptional Flows:

A-1:

1. Error messages are shown to the bill payer.
2. System asks bill payer to select “Cancel” or “Try Again”.
3. “Cancel” will bring to homepage, “Try Again” will perform step 3 of normal flow again.

Table 4.12: Use Case of Set Up Bills Reminder

Use Case Name**:** Set Up Bills Reminder	ID**: UC12**	Importance Level**:** **Medium**
Primary Actor**:** Bill Payer	Use Case Type**:** Detail, Real	
Stakeholders and Interests**:** Bill payer – Bill payer set up the reminder to receive bill payment reminders during certain condition.		
Brief Description**:** This use case describes how bill payers are able to set up reminder for themselves.		
Trigger**:** Bill payers want to avoid late payment caused by overlooked of the payment deadline.		
Relationships: Association **:** Bill Payer Include **:** - Extend **:** Register Bill Generalization **: -**		

Normal Flow of Events: 1. After successfully register a bill, system will ask bill payer if he/she wants to set up bill reminder. 2. Bill payer selects yes or manually go to the bill reminder set up module. 3. Bill payer selects the bills and bill reminder time and medium. 4. System reminds the user about bill payment at the selected time through the selected medium.
Sub-flows:
Alternate/Exceptional Flows:

Table 4.13: Use Case of View Bills Histories

Use Case Name: View Bills Histories	ID: **UC13**	Importance Level: **Medium**
Primary Actor: Bill Payer	Use Case Type: Detail, Real	
Stakeholders and Interests: Bill payer – Bill payer wants to view his bills issued by billing companies and his bill payments made histories.		
Brief Description: This use case describes how bill payers view their bills histories, including bills issued and payments made.		

Trigger**:** Bill payers want to check his previous bills issued by billing companies and his payments made.
Relationships: Association **:** Bill Payer Include **: -** Extend **: -** Generalization **: -**
Normal Flow of Events**:** 1. Bill payer press drawer menu button to open the drawer, 2. Bill payer selects “Bill History” to navigate to that page. 3. Bill payer views the billing histories and payment histories retrieved from database.
Sub-flows**:**
Alternate/Exceptional Flows:

Table 4.14:View Notifications

Use Case Name**:** View Notifications	ID**: UC14**	Importance Level**:** **Medium**

Primary Actor**:** Bill Payer

Use Case Type**:** Detail, Real

Stakeholders and Interests**:**

Bill payer – Bill payer views the notifications sent by BillHub.

Brief Description**:** This use case describes how bill payers view his notifications message.

Trigger**:** Bill payers want to check what is the notification messages sent by BillHub.

Relationships:

Association	**:** Bill Payer
Include	**: -**
Extend	**: -**
Generalization	**: -**

Normal Flow of Events**:**

1. When a notification is sent to user, user can see the number of notifications unchecked on the bell shape icon at home page.
2. User clicks the bell shape icon to navigate to notification page.
3. User clicks the unread notification.
4. The notification message is shown.

Sub-flows**:**

Alternate/Exceptional Flows:

4.5 Requirements Traceability Matrix

Table 4.15: Requirements Traceability Matrix of the system

FR ID	**Functional Requirement**	**UC ID**	**Use Case Name**	**Roles**	**Remarks**
FR01	The system should allow bill payers to register accounts by collecting necessary information such as name, email and password.	1	Register Account	Bill Payer	Highly Priority Requirement
FR02	The system should allow bill payers to login registered account by authenticating user based on their credentials.	2	Login Account	Bill Payer	Highly Priority Requirement
FR03	The system should allow bill payers to register bills with the account numbers	3	Register Bills	Bill Payer	Highly Priority Requirement
FR04	The system should allow bill payers to view the list of registered bills with bills information such as outstanding amount and payment deadline	4	View Bill	Bill Payer	Highly Priority Requirement
FR05	The system should allow bill payers to pay one bill or multiple bills under one single transaction.	5	Pay bills	Bill Payer	Highly Priority Requirement

FR06	The system should allow bill payers to apply for auto payment on their selected bills.	10	Register Auto Payment	Bill Payer	Highly Priority Requirement
FR07	The system should remind bill payers of upcoming bills due dates.	12	Set Up Bil Reminder	Bill Payer	Highly Priority Requirement
FR08	The system should allow bill payers to view the analysis of the bills issued and payments made.	9	View Bill Analysis	Bill Payer	Highly Priority Requirement
FR09	The system should allow bill payers to view the details of their bills.	4	View Bill	Bill Payer	Highly Priority Requirement
FR10	The system should allow bill payers to view billing and payment histories of the bills	13	View Bill Histories	Bill Payer	Highly Priority Requirement
FR11	The system should allow bill payers to reload credits for bill payments.	11	Top Up Credit	Bill Payer	Highly Priority Requirement
FR12	The system should allow bill payers to check notifications received	14	View Notifications	Bill Payer	Lower Priority Requirement

FR13	The system should allow billing companies to send and update billing details.	7	Send Billing Details	Billing Company	Highly Priority Requirement

4.6 Entity Relationship Diagram

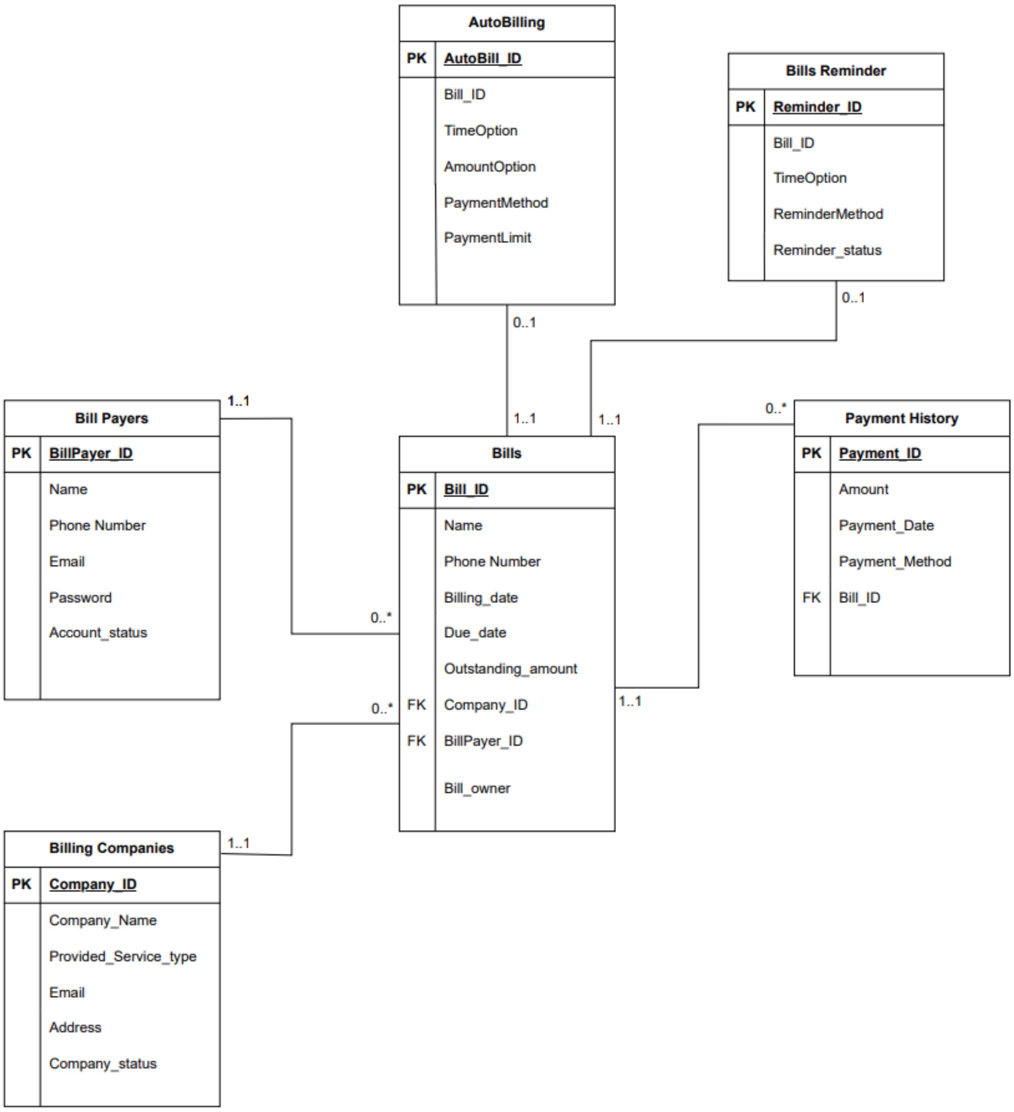


Figure 4.21: Entity Relationship Diagram (ERD) of proposal system

4.7 Interface Flow Diagram

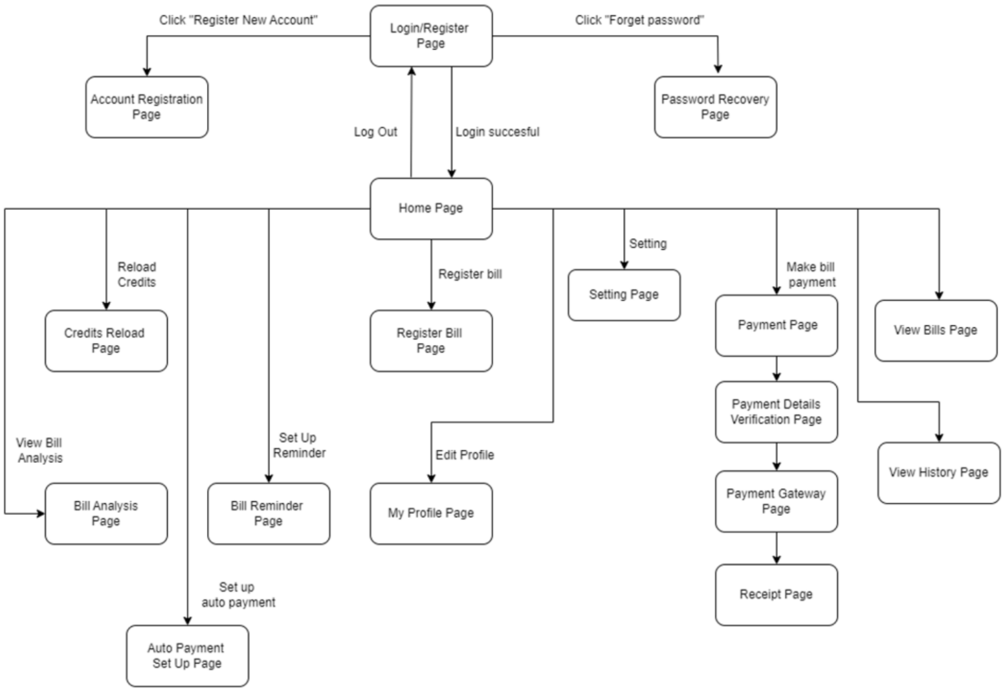
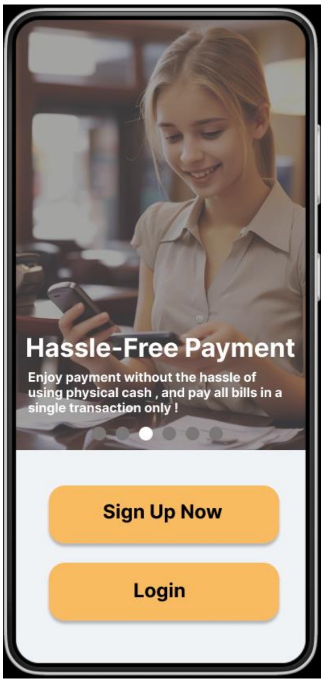


Figure 4.22: Interface flow diagram of proposed system

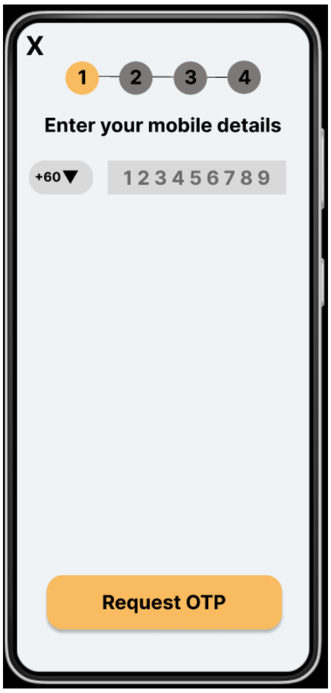
4.8 Prototype Interface



Hassle-Free Payment

Enjoy payment without the hassle of using physical cash , and pay all bills in a single transaction only !

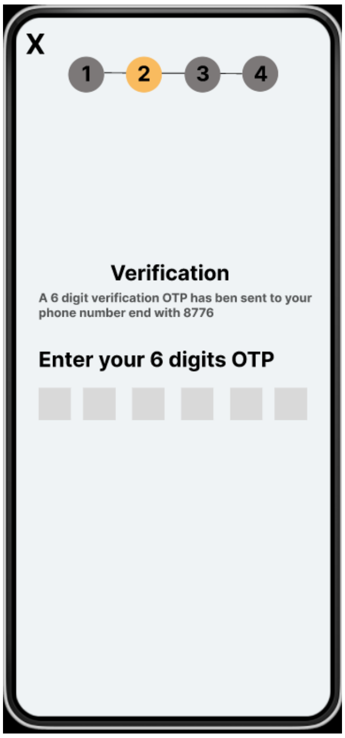
Figure 4.23: Authentication Page



Enter your mobile details

+60▼ 123456789

Figure 4.24: Registration Page 1

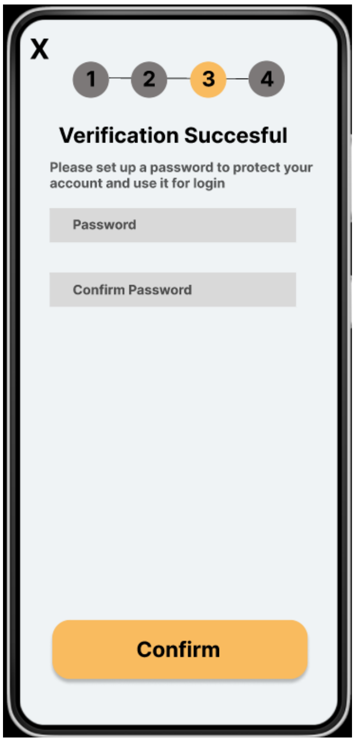


Verification

A 6 digit verification OTP has ben sent to your phone number end with 8776

Enter your 6 digits OTP

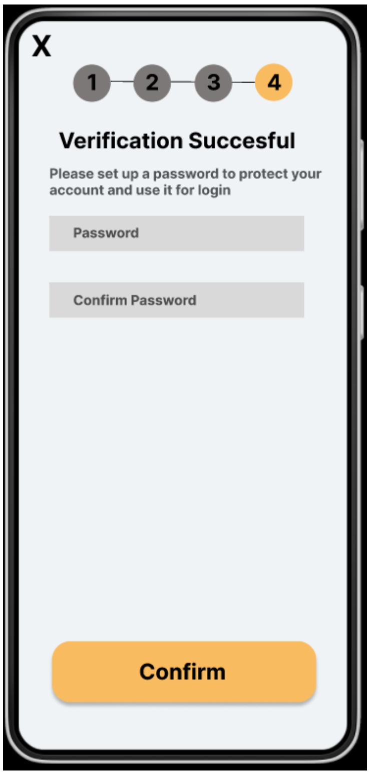
Figure 4.25: Registration Page 2



Verification Succesful

Please set up a password to protect your account and use it for login

Figure 4.26: Registration Page 3



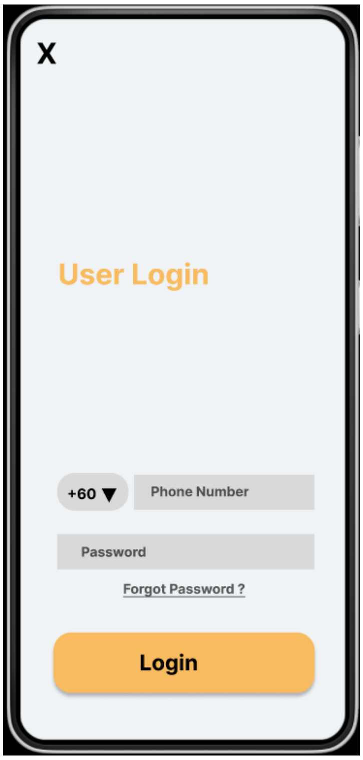
X

1 2 3 4

Verification Succesful

Please set up a password to protect your account and use it for login

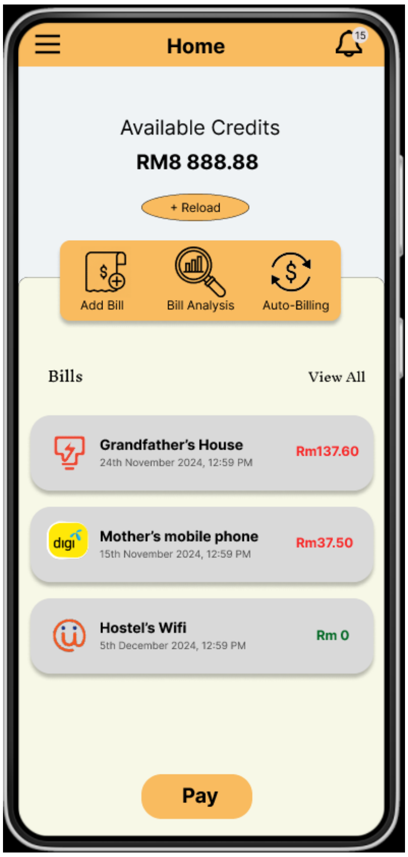
Figure 4.27: Registration Page 4



X

User Login

Figure 4.28: Registration Page 5



Home

15

Available Credits

RM8 888.88

+ Reload

Add Bill

Bill Analysis

Auto-Billing

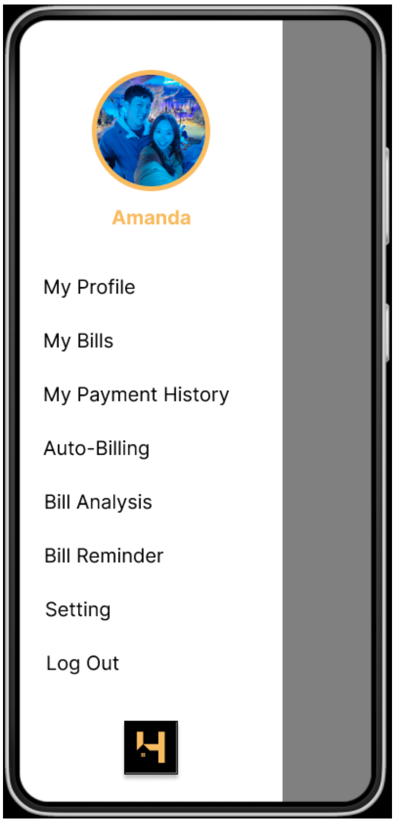
Bills

View All

Grandfather's House 24th November 2024, 12:59 PM	Rm137.60
Mother's mobile phone 15th November 2024, 12:59 PM	Rm37.50
Hostel's Wifi 5th December 2024, 12:59 PM	Rm 0

Pay

Figure 4.29: Home Page



Amanda

My Profile

My Bills

My Payment History

Auto-Billing

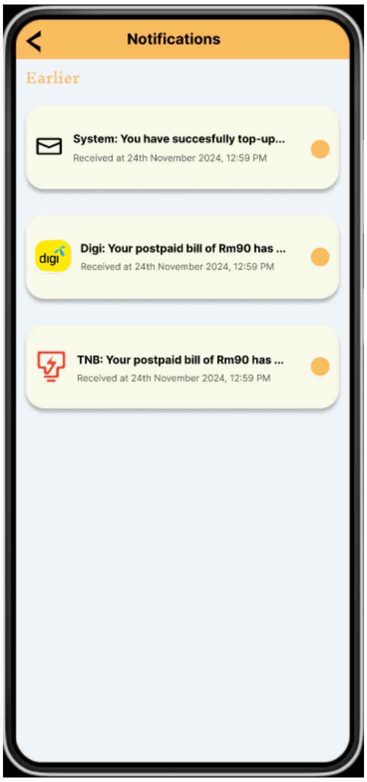
Bill Analysis

Bill Reminder

Setting

Log Out

Figure 4.30: Side Menu



Notifications

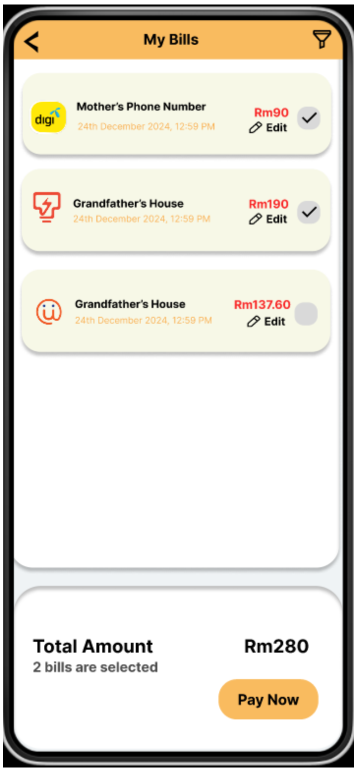
Earlier

System: You have succesfully top-up...
Received at 24th November 2024, 12:59 PM

Digi: Your postpaid bill of Rm90 has ...
Received at 24th November 2024, 12:59 PM

TNB: Your postpaid bill of Rm90 has ...
Received at 24th November 2024, 12:59 PM

Figure 4.31: Notification Page



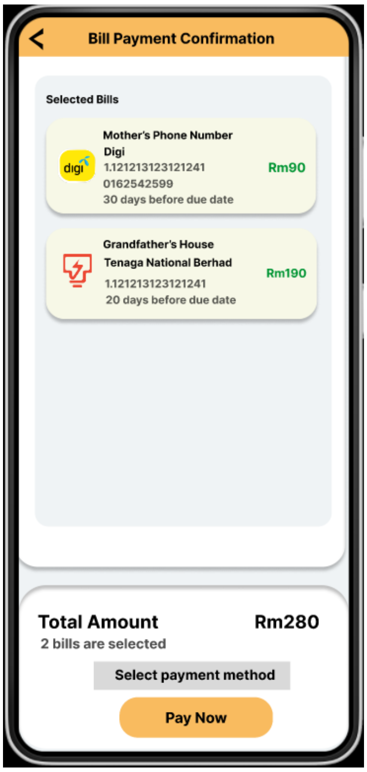
My Bills

Mother's Phone Number 24th December 2024, 12:59 PM	Rm90 Edit	✓
Grandfather's House 24th December 2024, 12:59 PM	Rm190 Edit	✓
Grandfather's House 24th December 2024, 12:59 PM	Rm137.60 Edit	○

Total Amount Rm280
2 bills are selected

Pay Now

Figure 4.32: My Bills Page



Bill Payment Confirmation

Selected Bills

Mother's Phone Number
Digi
1.121213123121241
0162542599
30 days before due date
Rm90

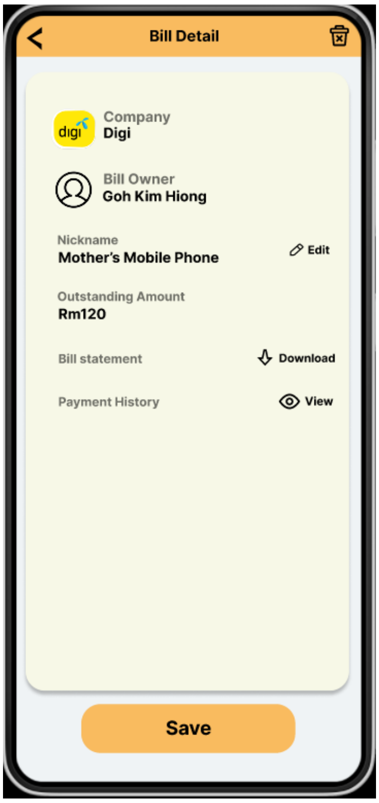
Grandfather's House
Tenaga National Berhad
1.121213123121241
20 days before due date
Rm190

Total Amount Rm280
2 bills are selected

Select payment method

Pay Now

Figure 4.33: Bill Payment Confirmation Page



Bill Detail

Company
Digi

Bill Owner
Goh Kim Hiong

Nickname
Mother's Mobile Phone
Edit

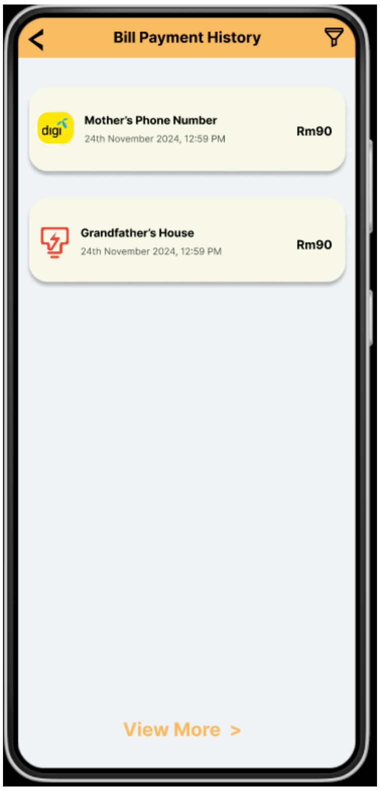
Outstanding Amount
Rm120

Bill statement
Download

Payment History
View

Save

Figure 4.34: Bill Detail Page



digi	Mother's Phone Number 24th November 2024, 12:59 PM	Rm90
	Grandfather's House 24th November 2024, 12:59 PM	Rm90

Figure 4.35: Bill Payment History Page

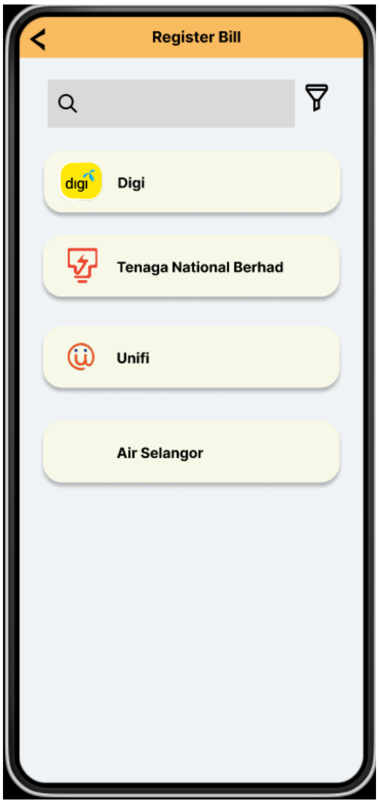


Figure 4.36: Register Bill Page 1

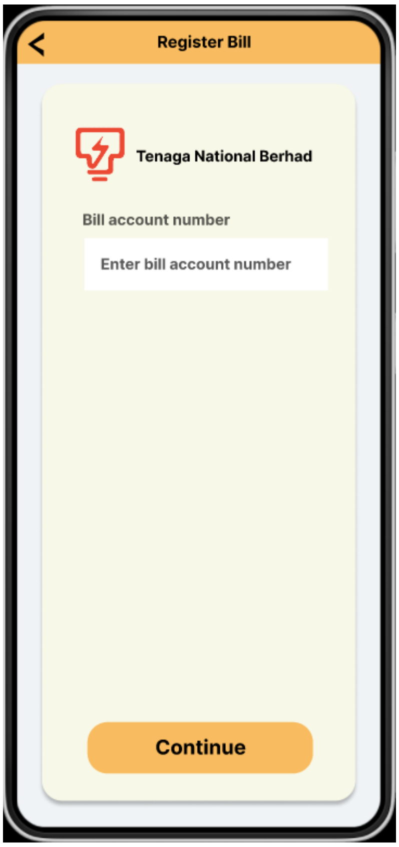
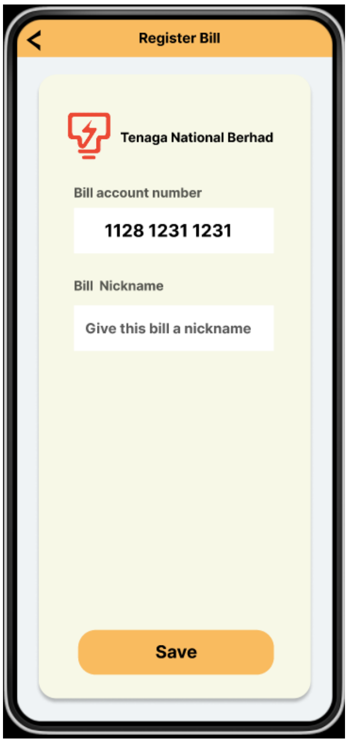


Figure 4.37: Register Bill Page 2



1128 1231 1231

Figure 4.38: Register Bill Page 3

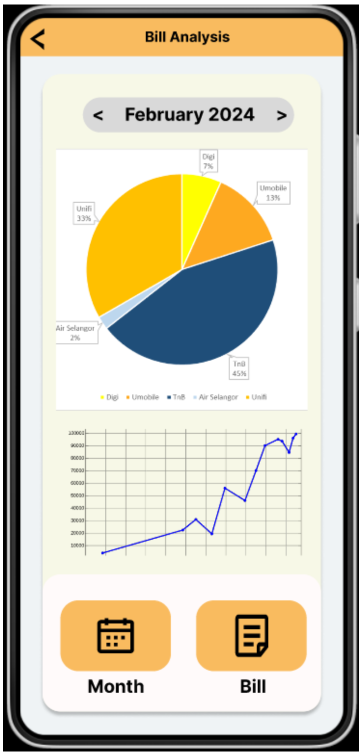
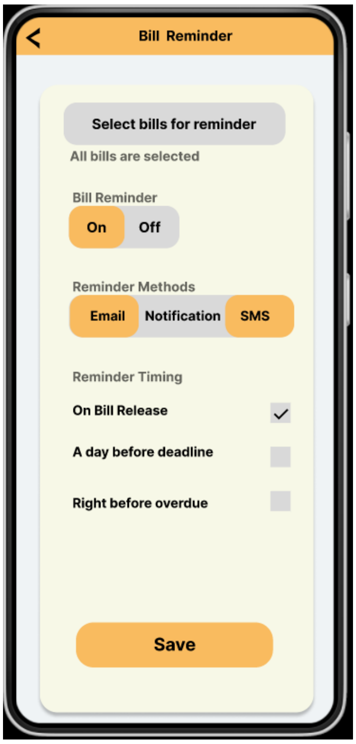


Figure 4.39: Bill Analysis Page



Select bills for reminder

All bills are selected

Bill Reminder

On Off

Reminder Methods

Email Notification SMS

Reminder Timing

On Bill Release ☑

A day before deadline □

Right before overdue □

Figure 4.40: Bill Reminder Page

CHAPTER 5

SYSTEM DESIGN

5.1 Introduction

This chapter describes the system architecture design, including the frontend, backend and communication between them.

5.2 System Architecture Design

This application facilitates communication between three key parties, bill payers, billing companies, and BillHub. The BillHub server acts as the central hub for all interactions. Bill payers use the BillHub application to send HTTP requests to the server from the frontend, while billing companies connect with BillHub by sending HTTP requests through their integrations. The BillHub server then processes these requests by performing CRUD (Create, Read, Update, Delete) operations on the database, ensuring seamless communication and data management between all parties involved.

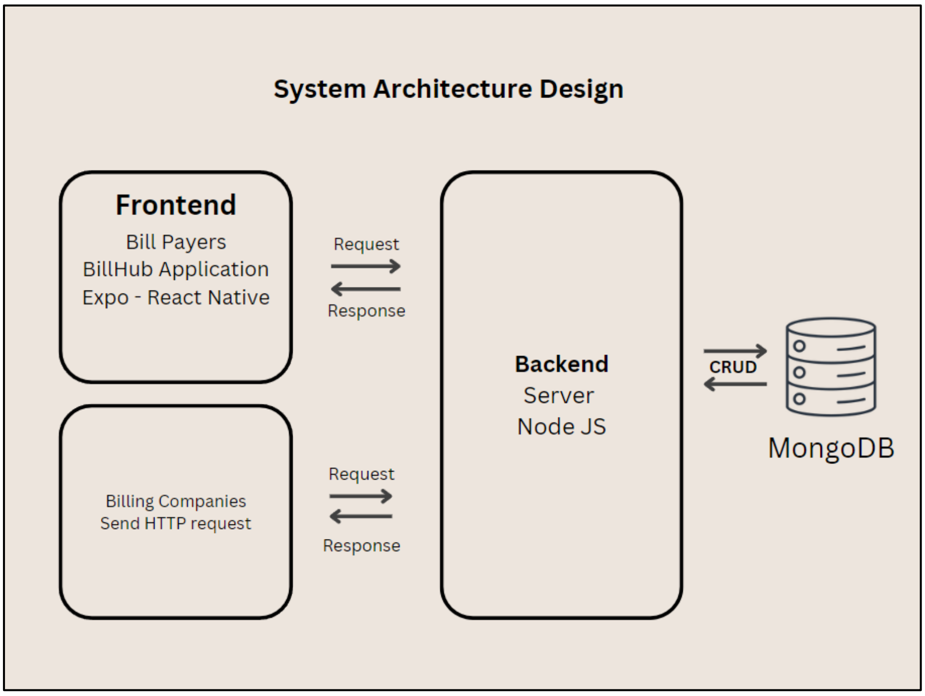


Figure 5.1: System Architecture Design

5.2.1 Front-end architecture

React Native components are dynamic elements that respond to state changes, component interactions, and user inputs, such as text entry or button clicks. These components are used to develop the frontend user interface of BillHub. The main entry point of this project, index.tsx, contains the <Navigation/> component, which manages navigation between different JavaScript and JSX files that utilize native UI elements such as <View>, <Text>, and <Image> to render the user interface. In addition, the <Navigation/> component is wrapped inside <AuthProvider> and <StripeProvider> components, allowing all pages to access user information and handle payments. The <Navigation/> setup includes a nested structure combining a drawer navigator and a stack navigator. It initially directs the user to the "SplashScreen" and then determines whether the user should be automatically logged in and navigated to the "Home" screen or redirected to the login page.

In React Native, the application's user interface (UI) is rendered on the client side, meaning it is rendered directly on the mobile device where BillHub is installed. During the installation of the application, the native code and all necessary components are installed on the device. When the application is launched, the JavaScript code, which was bundled during the build process, is loaded by the React Native runtime on the device. This JavaScript bundle is then executed, rendering the UI components and allowing users to interact with the app in real-time. This setup enables users to perform actions like making immediate payment transactions and other interactions directly on the device.

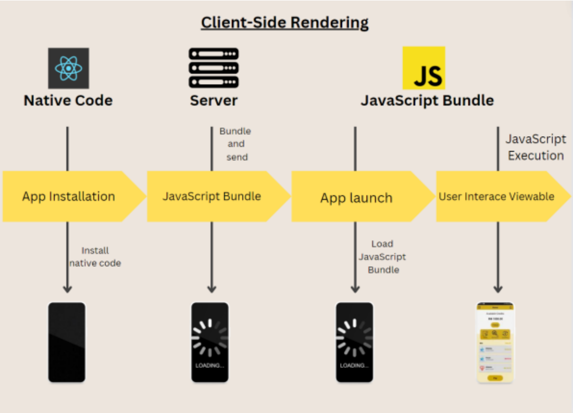


Figure 5.2: Client-Side Rendering

Besides, React Native interacts with the native components of the device’s operating system such as iOS and Android through JavaScript. There are three main threads in React Native which are JavaScript thread, native thread and bridge. JavaScript thread executes the JavaScript code, Native thread renders the UI, and bridge acts as communication layer between the JavaScript thread and native threads. The bridge serialize data in JSON format to send it between the JavaScript and native threads. Thus, it provides a native-like user experience while still using single codebase for multiple platforms. In short, the use of React Native reduces the need for developers to create separate projects for different operating systems and also reduces the workload on the server by distributing rendering tasks to the client side.

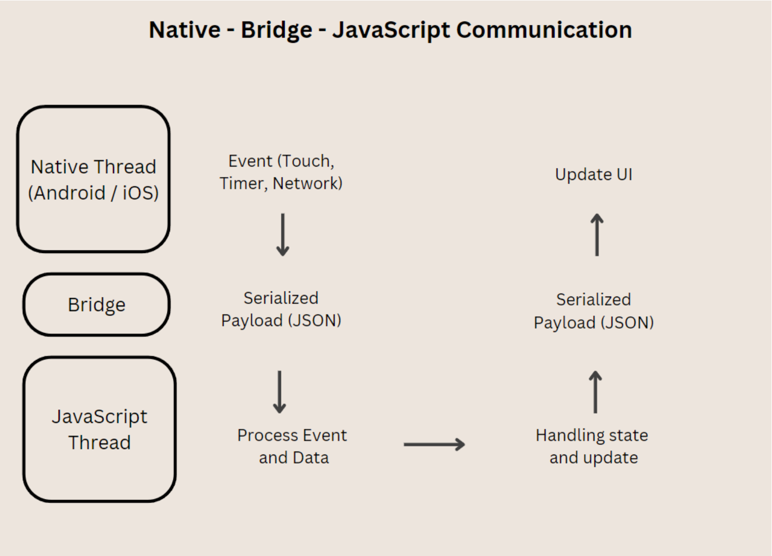

Native - Bridge - JavaScript Communication
Native Thread (Android / iOS)
Bridge
JavaScript Thread
Event (Touch, Timer, Network)
Serialized Payload (JSON)
Process Event and Data
Update UI
Serialized Payload (JSON)
Handling state and update

Figure 5.3: Native-Bridge-JavaScript Communication

Expo is a set of tools designed to be used with React Native, serving as the primary framework for bundling and developing applications in this project. The Expo Go app is installed on a physical Android mobile device, while the project is initialized using the ‘expo init’ command to start an Expo project. When the project is running, the mobile device can download and execute the JavaScript code from the development machine through a given URL. The project is then bundled, awaiting developers to connect to the Expo client by scanning a provided QR code or entering the URL. The Expo client loads the

JavaScript bundles from the localhost, which simplifies the development process by eliminating the need to run a virtual device through Android Studio. Expo streamlines the process, allowing developers to bundle and run the project more efficiently, with less time and reduced resource consumption on the development machine.

5.2.2 Back-end architecture

BillHub's backend is developed using Node.js with MongoDB as the primary database system. The server.js is written in Node.js, it consists of functions like scheduling reminder and auto-billing jobs with Agenda, sending email using Nodemailer and sending SMS through Twillio. Besides, the server also in charge of running database connection with the MongoDB. Models are created to define the structure of collections necessary for data storage. Correspondingly, controllers and routes are set up to handle HTTP requests, mapping them to the appropriate functions. The seamless integration between the server and database enables the server to efficiently process CRUD operations upon receiving HTTP requests via Axios from the frontend. The server then sends back responses, either the data or error message.

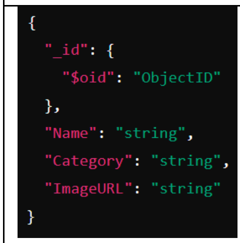
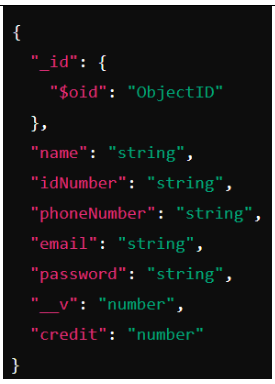
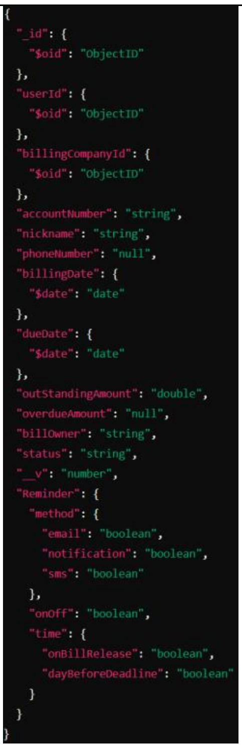
5.3 Database Architecture

MongoDB is a type of NoSQL database, which is non-relational database. Therefore, the schema design is more flexible and less structured, it behaves similarly to a key-value storage, but it also allows for more complex data structures like nested documents. Although NoSQL databases do not enforce relationships between collections like relational databases, the collections in the NoSQL database can still be related to other collections by storing the auto-generated IDs of the collection as one of its attributes. While there is no standard diagramming convention for NoSQL databases, an entity-relationship-like diagram can be used to visualize and illustrate the database schema.

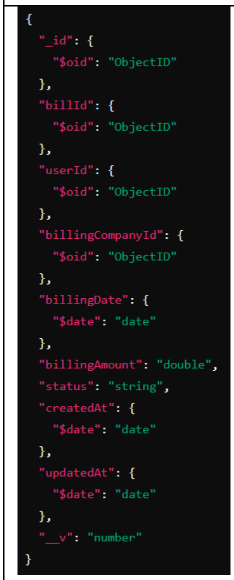
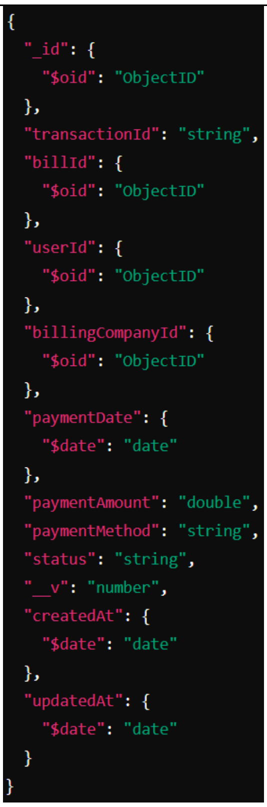
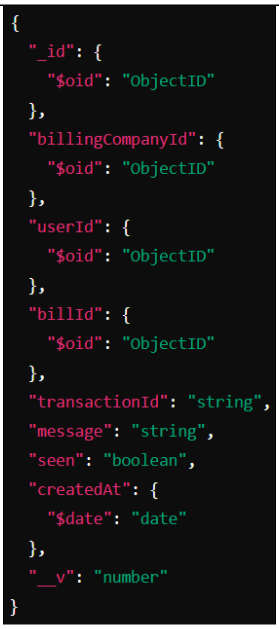
5.3.1 Database Schema

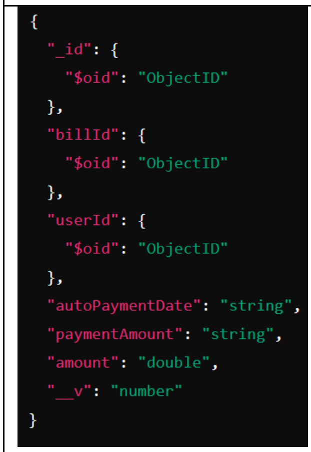
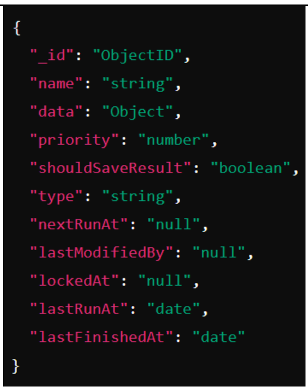
Table 5.1: Database Schema

billingcompanies	users	bills


{
  "_id": {
    "$oid": "ObjectID"
  },
  "Name": "string",
  "Category": "string",
  "ImageURL": "string"
}
	
	{
	  "_id": {
	    "$oid": "ObjectID"
	  },
	  "name": "string",
	  "idNumber": "string",
	  "phoneNumber": "string",
	  "email": "string",
	  "password": "string",
	  "__v": "number",
	  "credit": "number"
	}
		
		{
		  "_id": {
		    "$oid": "ObjectID"
		  },
		  "userId": {
		    "$oid": "ObjectID"
		  },
		  "billingCompanyId": {
		    "$oid": "ObjectID"
		  },
		  "accountNumber": "string",
		  "nickname": "string",
		  "phoneNumber": "null",
		  "billingDate": {
		    "$date": "date"
		  },
		  "dueDate": {
		    "$date": "date"
		  },
		  "outStandingAmount": "double",
		  "overdueAmount": "null",
		  "billOwner": "string",
		  "status": "string",
		  "__v": "number",
		  "Reminder": {
		    "method": {
		      "email": "boolean",
		      "notification": "boolean",
		      "sms": "boolean"
		    },
		    "onOff": "boolean",
		    "time": {
		      "onBillRelease": "boolean",
		      "dayBeforeDeadline": "boolean"
		    }
		  }
		}

billinghistories	**paymenthistories**	**notifications**


{
  "_id": {
    "$oid": "ObjectID"
  },
  "billId": {
    "$oid": "ObjectID"
  },
  "userId": {
    "$oid": "ObjectID"
  },
  "billingCompanyId": {
    "$oid": "ObjectID"
  },
  "billingDate": {
    "$date": "date"
  },
  "billingAmount": "double",
  "status": "string",
  "createdAt": {
    "$date": "date"
  },
  "updatedAt": {
    "$date": "date"
  },
  "__v": "number"
}
	
	{
	  "_id": {
	    "$oid": "ObjectID"
	  },
	  "transactionId": "string",
	  "billId": {
	    "$oid": "ObjectID"
	  },
	  "userId": {
	    "$oid": "ObjectID"
	  },
	  "billingCompanyId": {
	    "$oid": "ObjectID"
	  },
	  "paymentDate": {
	    "$date": "date"
	  },
	  "paymentAmount": "double",
	  "paymentMethod": "string",
	  "status": "string",
	  "__v": "number",
	  "createdAt": {
	    "$date": "date"
	  },
	  "updatedAt": {
	    "$date": "date"
	  }
	}
		
		{
		  "_id": {
		    "$oid": "ObjectID"
		  },
		  "billingCompanyId": {
		    "$oid": "ObjectID"
		  },
		  "userId": {
		    "$oid": "ObjectID"
		  },
		  "billId": {
		    "$oid": "ObjectID"
		  },
		  "transactionId": "string",
		  "message": "string",
		  "seen": "boolean",
		  "createdAt": {
		    "$date": "date"
		  },
		  "__v": "number"
		}

Autobillings	agendaJobs	
		

```
{
  "_id": {
    "$oid": "ObjectID"
  },
  "billId": {
    "$oid": "ObjectID"
  },
  "userId": {
    "$oid": "ObjectID"
  },
  "autoPaymentDate": "string",
  "paymentAmount": "string",
  "amount": "double",
  "__v": "number"
}
```

```
{
  "_id": "ObjectID",
  "name": "string",
  "data": "Object",
  "priority": "number",
  "shouldSaveResult": "boolean",
  "type": "string",
  "nextRunAt": "null",
  "lastModifiedBy": "null",
  "lockedAt": "null",
  "lastRunAt": "date",
  "lastFinishedAt": "date"
}
```

5.3.2 Collection Description

Table 5.2: Collections Description Table

Collection	Description
users	Stores bill payers' information, including contact details, login credentials, and account-related data such as available credits.
billingcompanies	Stores information about billing companies that collaborate with BillHub, including the company name, category, and branding elements (e.g., logo URLs).
bills	Stores details of registered bills, allowing all parties to track and update the bills' information. Also stores data for scheduling reminders.
billinghistories	Tracks the history of bills issued by billing companies.
paymenthistories	Tracks the history of bill payments made by users.
notifications	Stores messages and sending party information to be shown to users.
autobillings	Stores automatic payment settings to schedule payment automation.
agendaJobs	Stores scheduled tasks for handling reminders and auto-billings.

5.4 Data Flow Diagram

5.4.1 Context Diagram

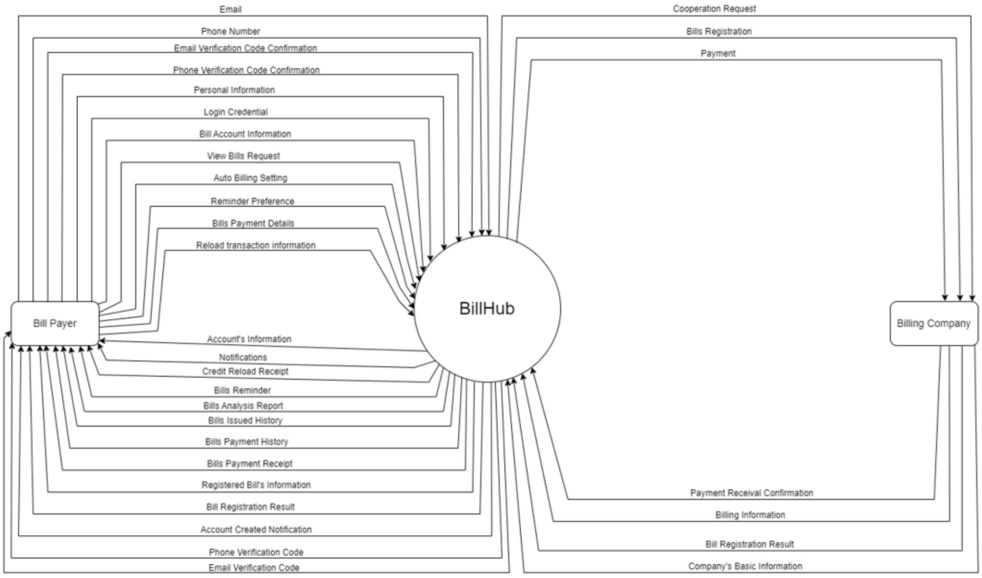


Figure 5.4: Context Diagram

5.4.2 DFD Level-0 Diagram

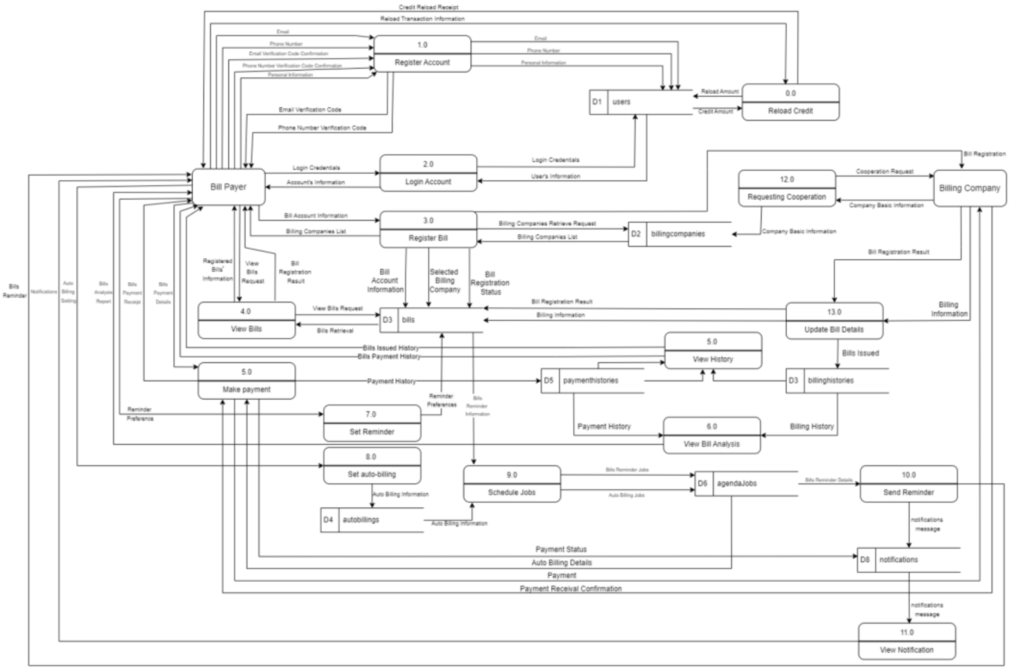


Figure 5.5: Level-0 Diagram

5.4.3 DFD Level-1 Diagram

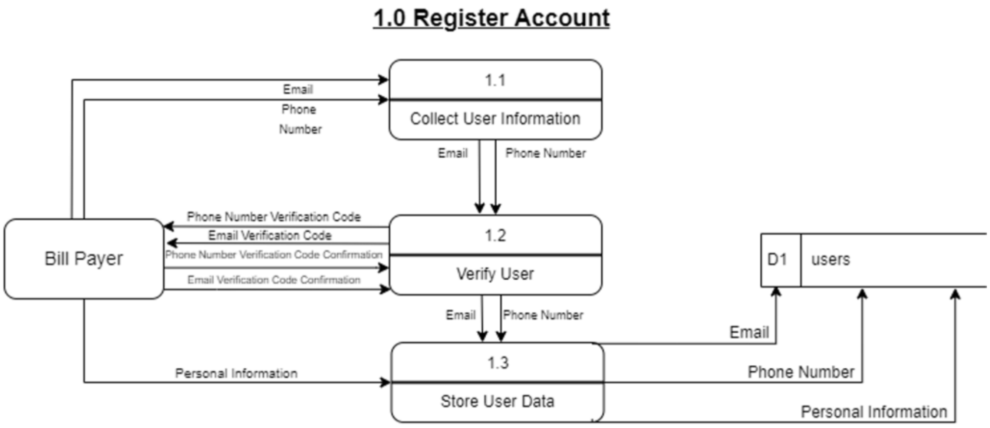


Figure 5.6: DFD Level-1 Register Account

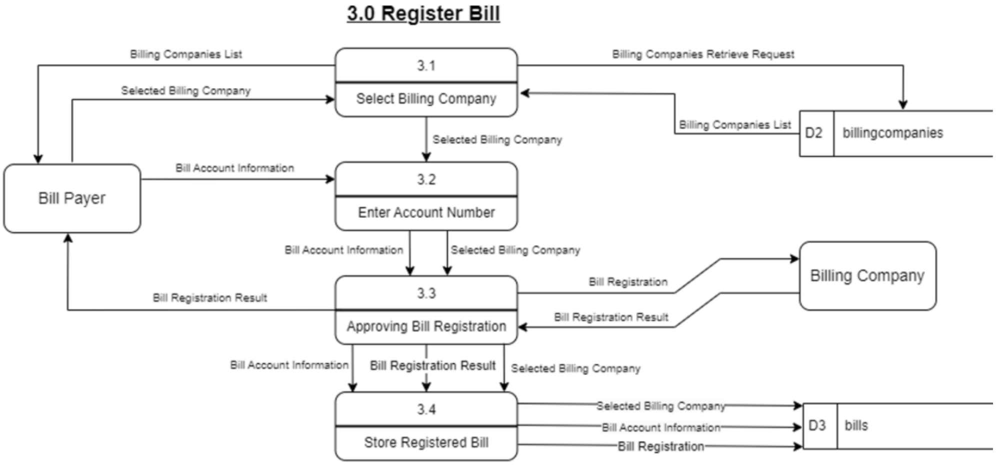


Figure 5.7: DFD Level-1 Register Bill

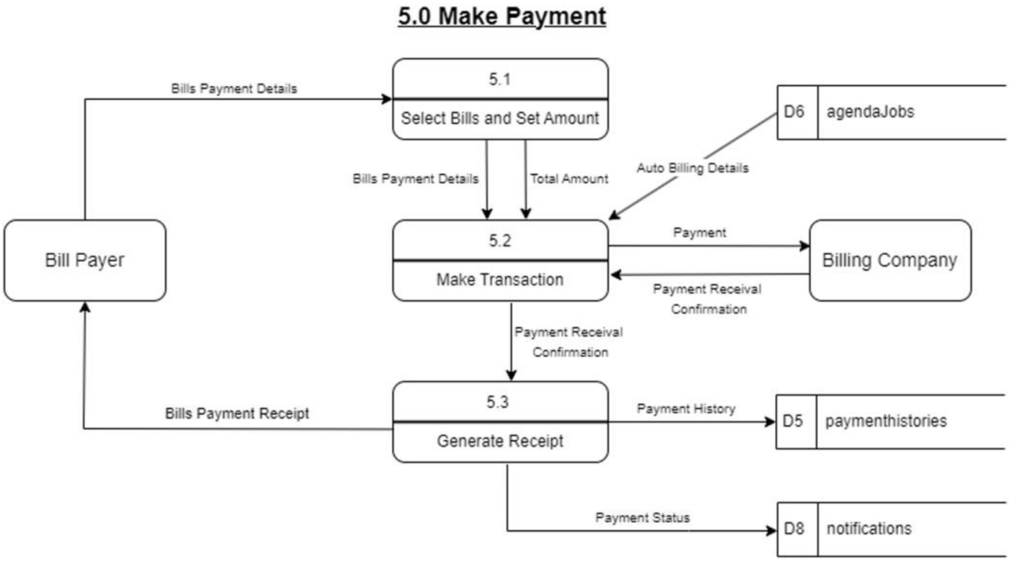


Figure 5.8:DFD Level-1 Bill Payment

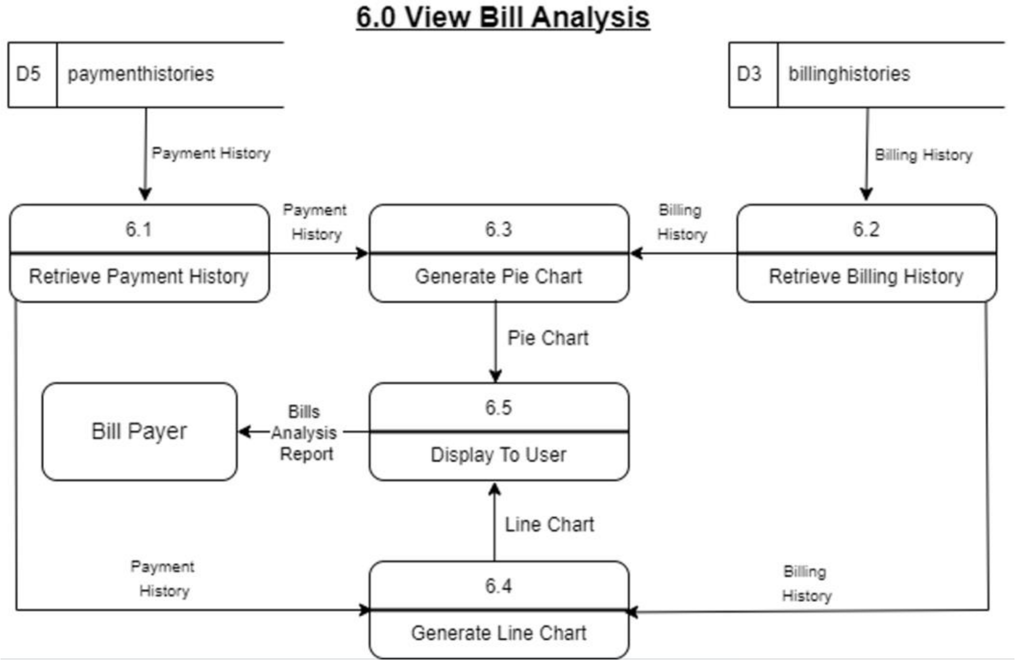


Figure 5.9: DFD Level-1 Bill Analysis

5.5 Activity Diagram

Activity diagrams are used to illustrate and visualize workflows within the developed system, showing the flows of activities and control. This activity diagrams contain three swimlanes, which represent different actors or parties involved in the process (Bill Payer, BillHub and Billing Company). The diagram provides a clear explanation of the interactions and responsibilities between the three different parties, making it easier to understand the structure of the system.

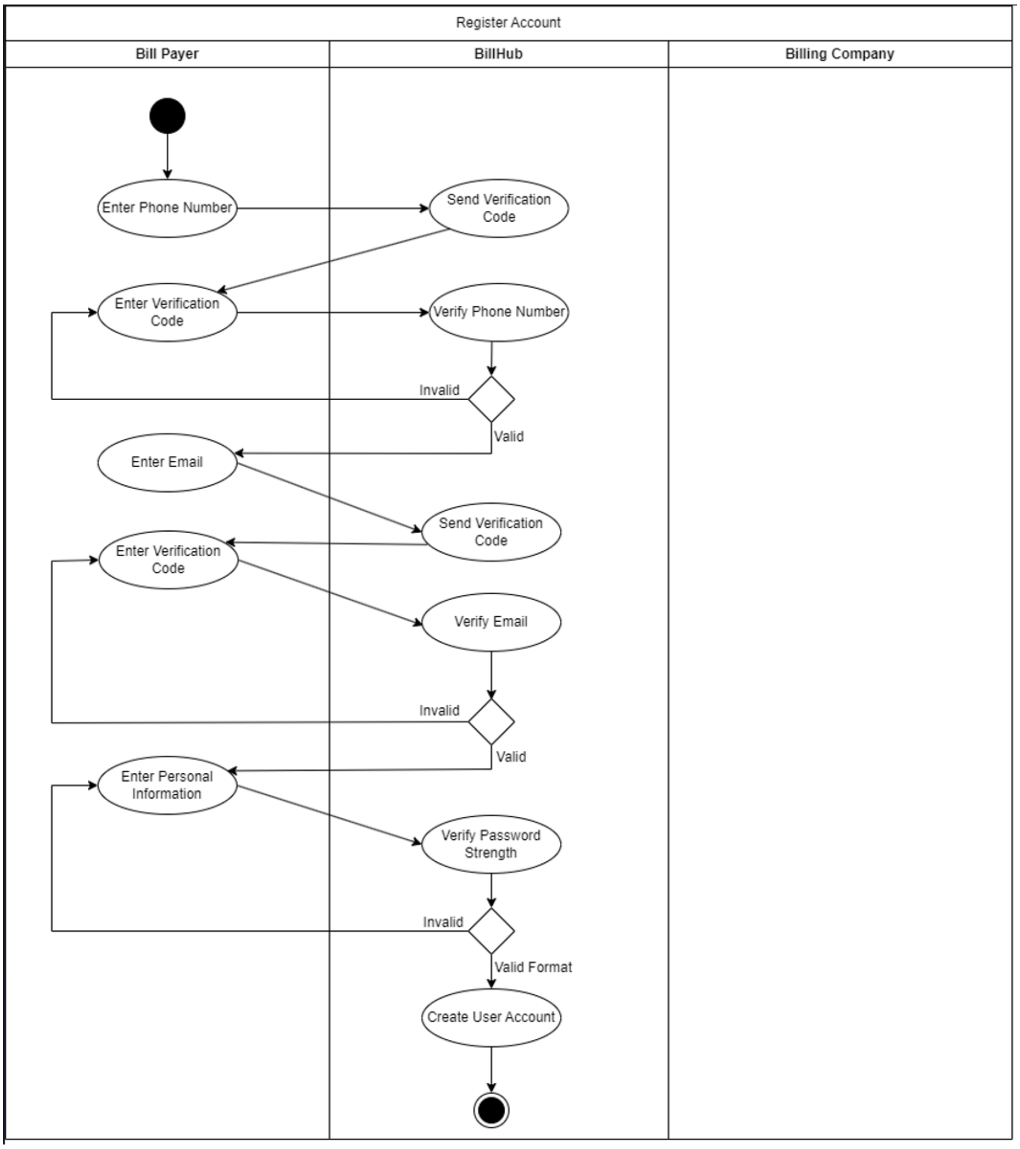


Figure 5.10: Register Account Activity

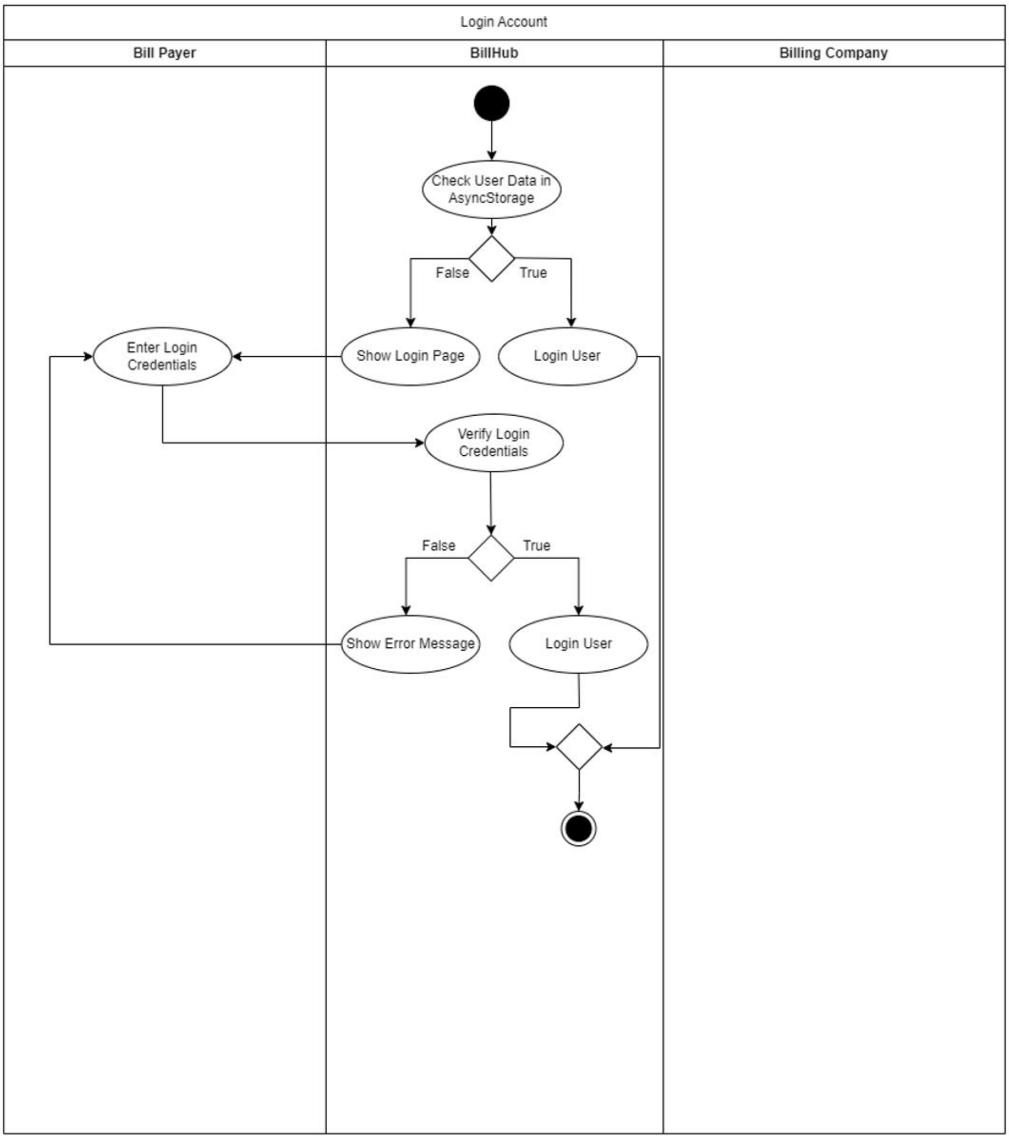


Figure 5.11: Login Account Activity

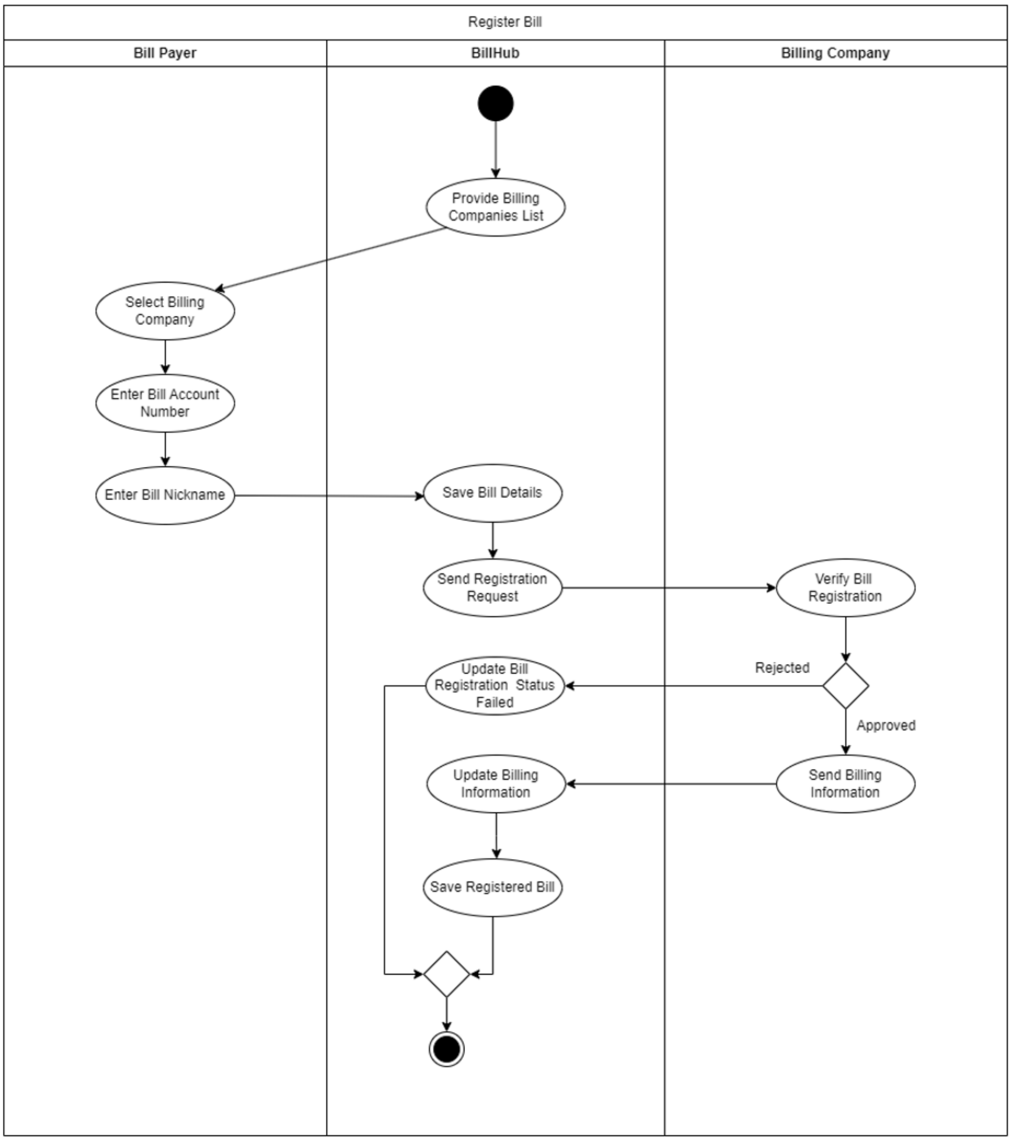


Figure 5.12: Register Bill Activity

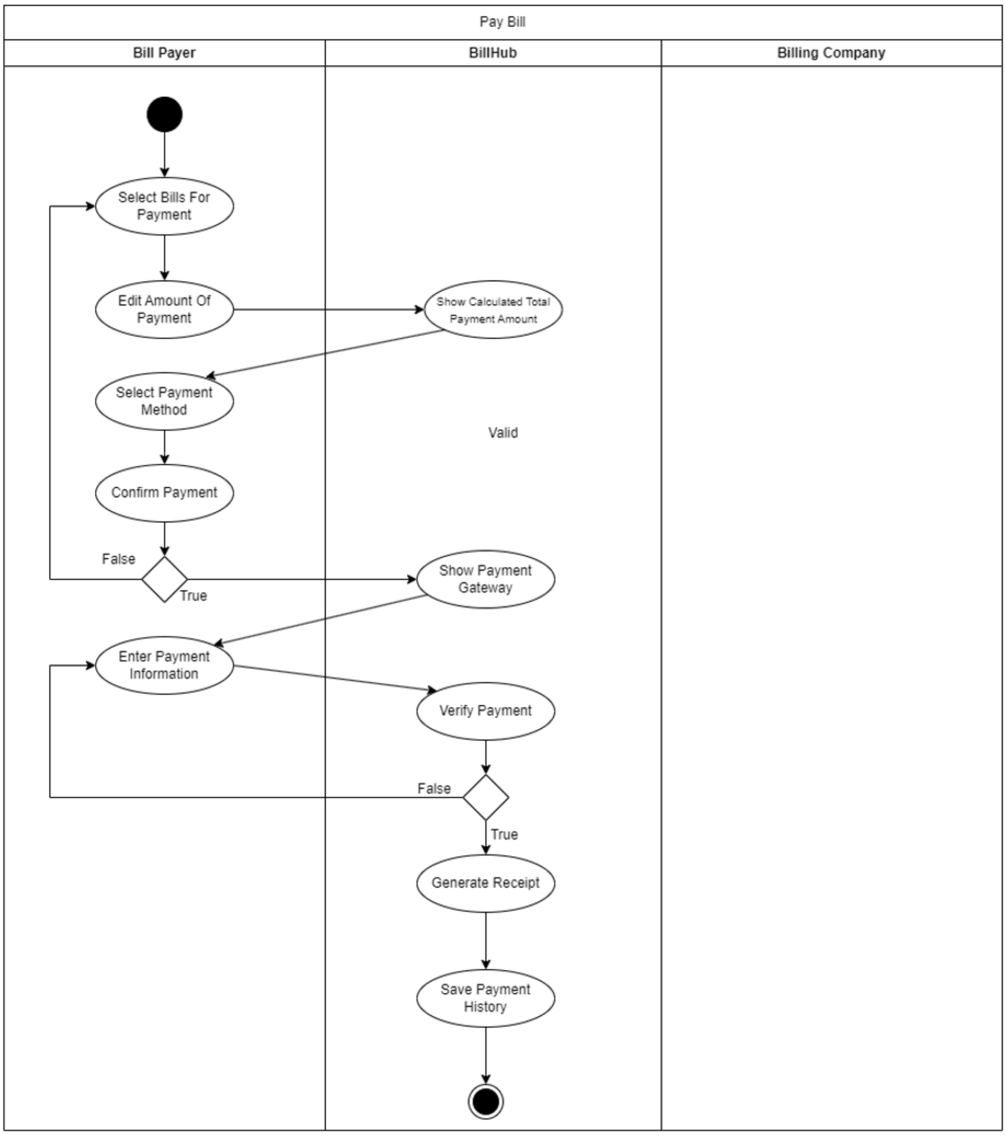


Figure 5.13: Pay Bill Activity

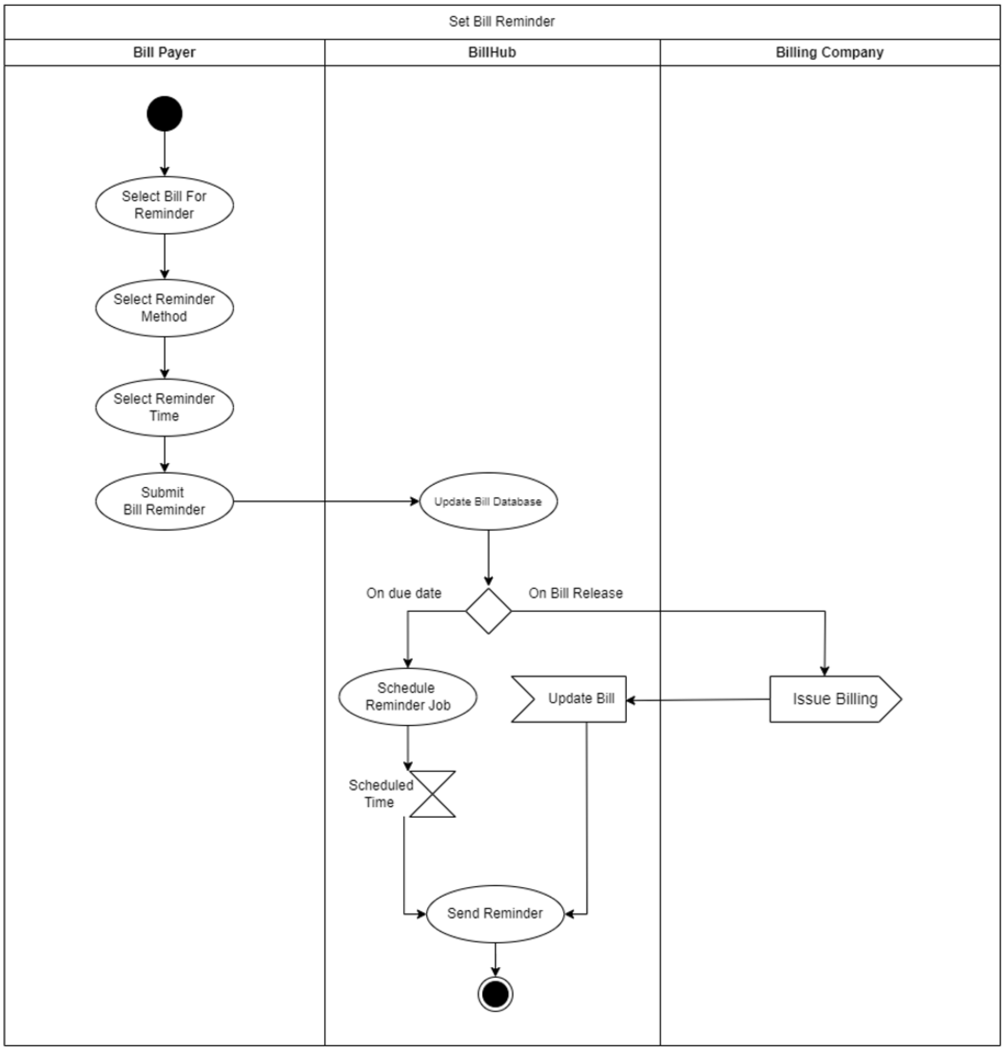


Figure 5.14: Set Bill Reminder Activity

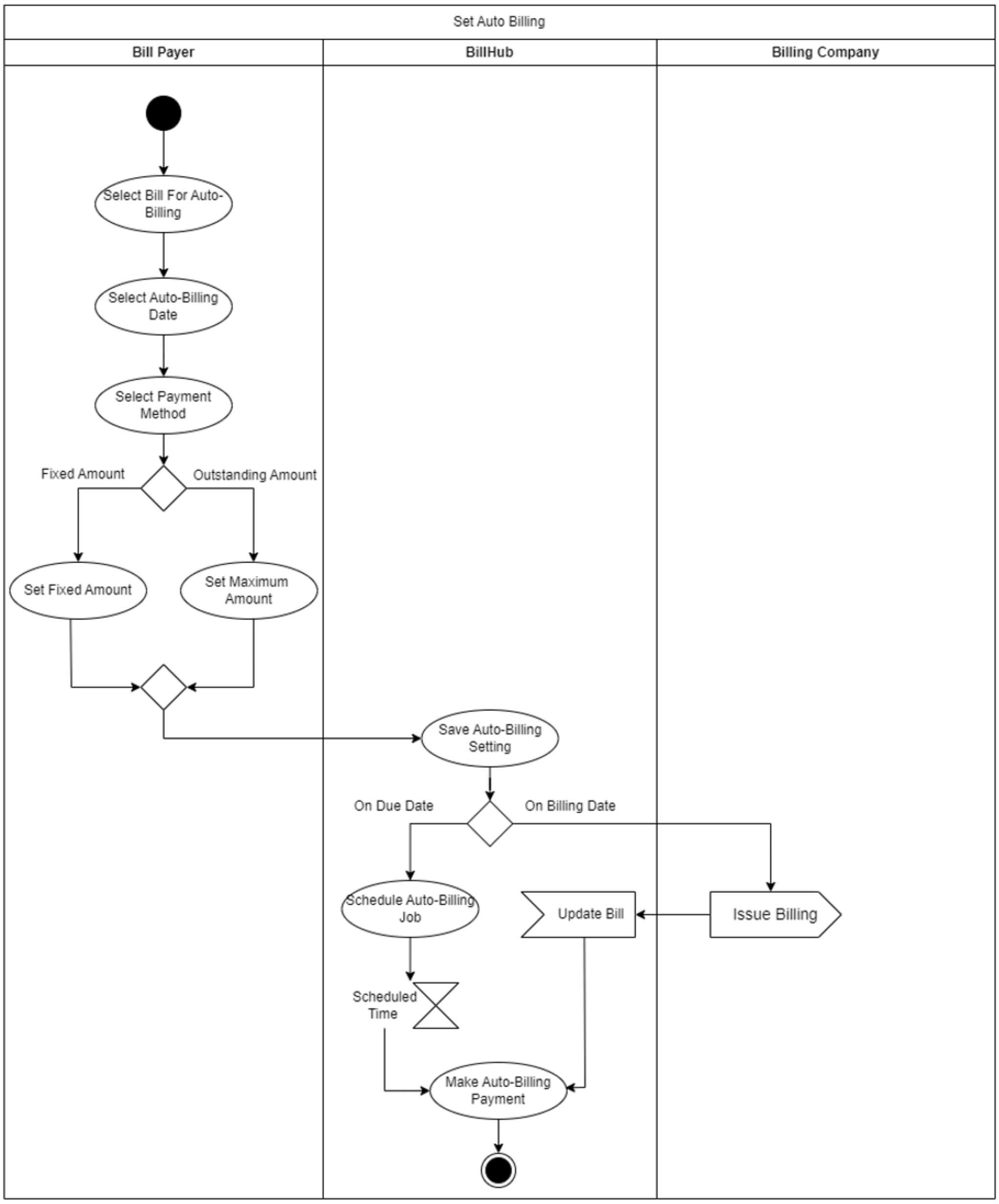


Figure 5.15: Set Auto-Billing Activity

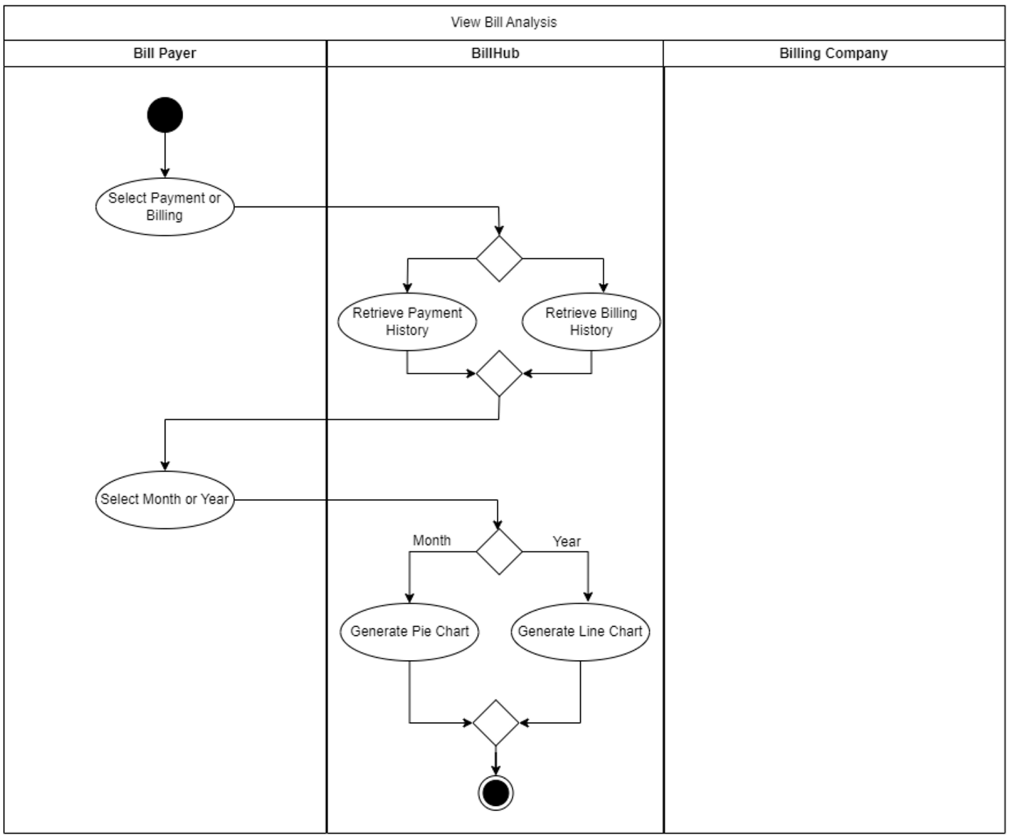


Figure 5.16: View Bill Analysis Activity

5.6 User Interface Design

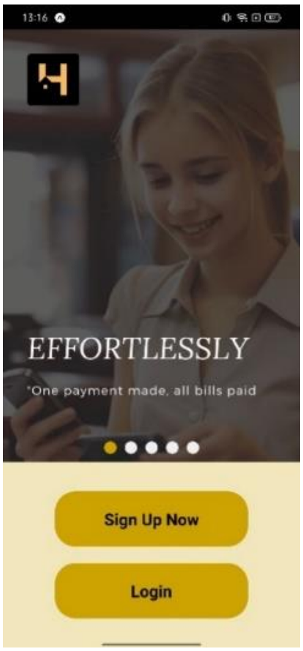
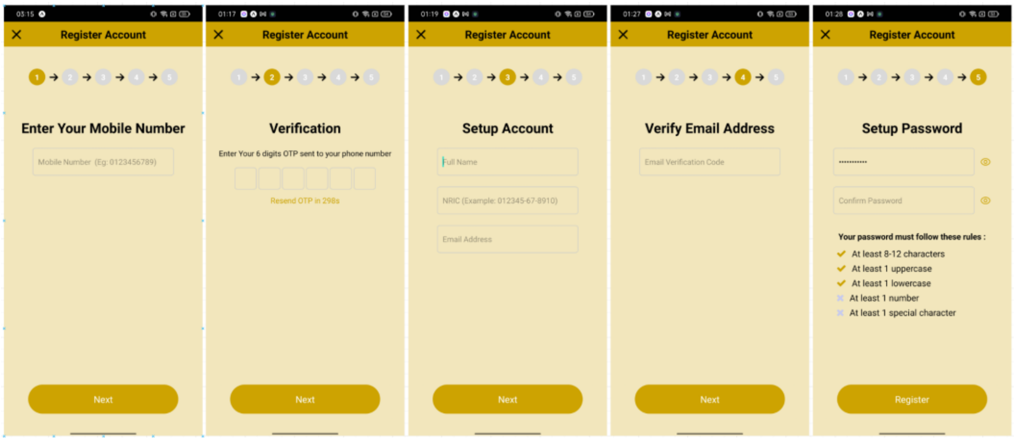


Figure 5.17: First Page

Figure 5.17 shows the first page shown to user when launching BillHub application, user will be shown with some introduction/advertisement slides while having the options to sign up an BillHub account or login if he already has one.



Enter Your Mobile Number

Mobile Number (Eg: 0123456789)

Verification

Enter Your 6 digits OTP sent to your phone number

Resend OTP in 298s

Setup Account

Full Name

NRIC (Example: 012345-67-8910)

Email Address

Verify Email Address

Email Verification Code

Setup Password

••••••••••••

Confirm Password

Your password must follow these rules :

✓ At least 8-12 characters
✓ At least 1 uppercase
✓ At least 1 lowercase
✗ At least 1 number
✗ At least 1 special character

Figure 5.18: Register Account

Figure 5.18 shows the module of “Register Account”, where user can follow the step 1 to step 5 to sign up an BillHub account.

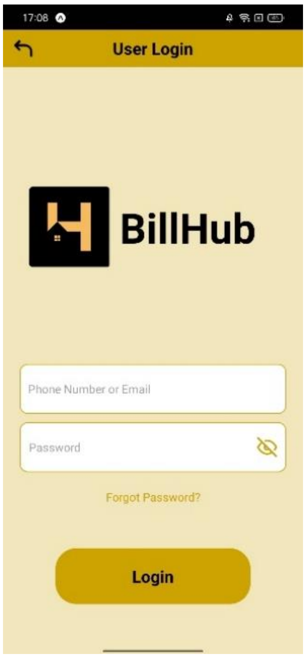


Figure 5.19: Login Page

Figure 5.19 shows the module of "Login", where user can use his email address or phone number to login his registered account with the correct password.

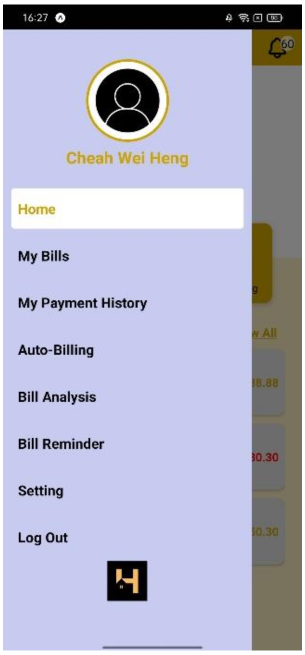
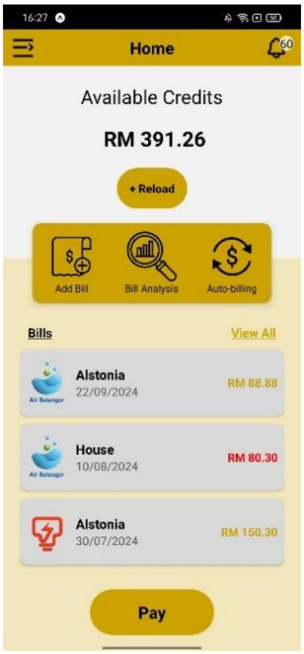


Figure 5.20: Drawer

Figure 5.20 shows the drawer navigator which consists most of the main modules of BillHub and allows user to navigate freely.



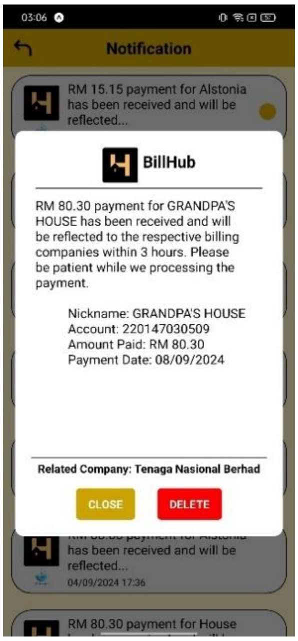
Available Credits

RM 391.26

Air Selangor	Alstonia 22/09/2024	RM 88.88
Air Selangor	House 10/08/2024	RM 80.30
	Alstonia 30/07/2024	RM 150.30

Figure 5.21: Home Page

Figure 5.21 shows the “Home” screen of BillHub which allows users to view remaining credits, registered bills information, number of unchecked notifications, and also buttons to navigate to different modules.



RM 15.15 payment for Alstonia has been received and will be reflected...

RM 80.30 payment for GRANDPA'S HOUSE has been received and will be reflected to the respective billing companies within 3 hours. Please be patient while we processing the payment.

Nickname: GRANDPA'S HOUSE
Account: 220147030509
Amount Paid: RM 80.30
Payment Date: 08/09/2024

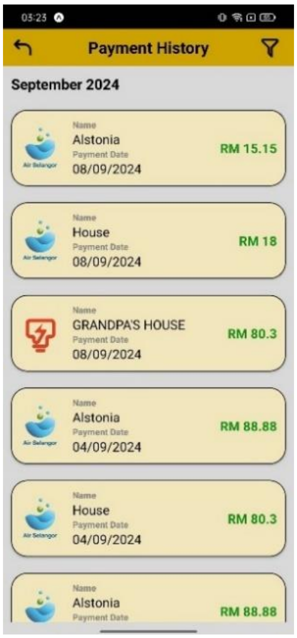
Related Company: Tenaga Nasional Berhad

has been received and will be reflected...
04/09/2024 17:36

RM 80.30 payment for House

Figure 5.22: Notifications Page

Figure 5.22 shows the “Notification” module, which allows users to check, read and delete his received notifications.



05:23

Payment History

September 2024

Name
Alstonia
Payment Date
08/09/2024
RM 15.15

Name
House
Payment Date
08/09/2024
RM 18

Name
GRANDPA'S HOUSE
Payment Date
08/09/2024
RM 80.3

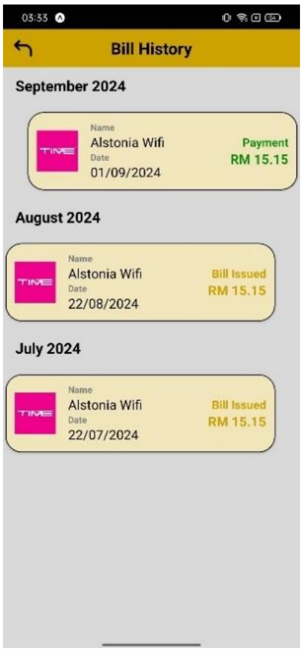
Name
Alstonia
Payment Date
04/09/2024
RM 88.88

Name
House
Payment Date
04/09/2024
RM 80.3

Name
Alstonia
Payment Date
RM 88.88

Figure 5.23: Payment History Page

Figure 5.23 shows “Payment History” module, which allows users to view the billing company, name of bill, payment date and payment amount about the payment he made, arranged starting from the recent payment.



05:55

Bill History

September 2024

Name
Alstonia Wifi
Date
01/09/2024
Payment
RM 15.15

August 2024

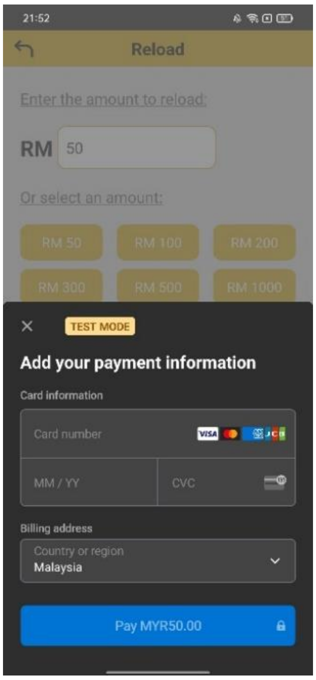
Name
Alstonia Wifi
Date
22/08/2024
Bill Issued
RM 15.15

July 2024

Name
Alstonia Wifi
Date
22/07/2024
Bill Issued
RM 15.15

Figure 5.24: Bill History Page

Figure 5.24 shows the “Bill History” module which shows the bills issued history and bills payment history of a selected bill.

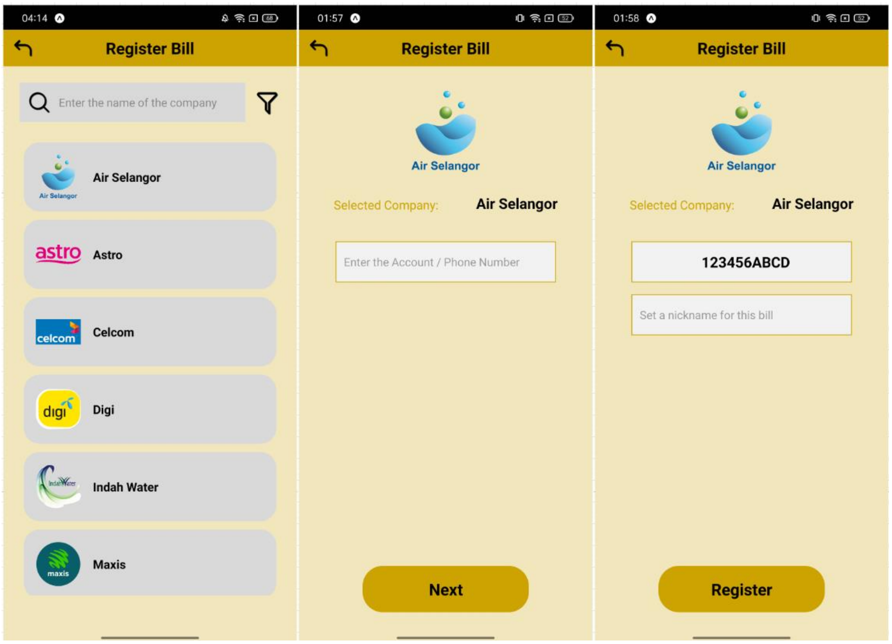


RM 50

Country or region
Malaysia

Figure 5.25: Reload Page

Figure 5.25 shows the “Reload” module which allows users to reload credits of his account through the payment gateway.



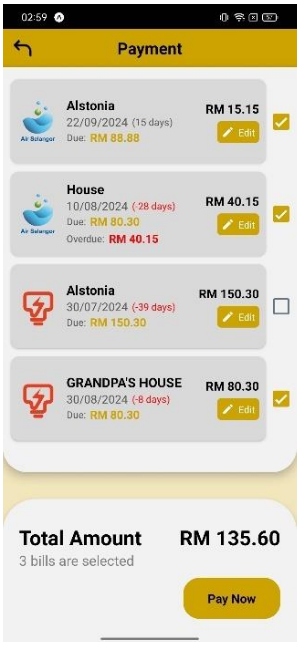
Selected Company: Air Selangor

Selected Company: Air Selangor

123456ABCD

Figure 5.26: Register Bill Page

Figure 5.26 shows the “Register Bill” module which allows users to register a bill by selecting a billing company and providing the bill’s account number.



02:59

Payment

Alstonia
22/09/2024 (15 days)
Due: RM 88.88
RM 15.15
Edit ☑

House
10/08/2024 (-28 days)
Due: RM 80.30
Overdue: RM 40.15
RM 40.15
Edit ☑

Alstonia
30/07/2024 (-39 days)
Due: RM 150.30
RM 150.30
Edit □

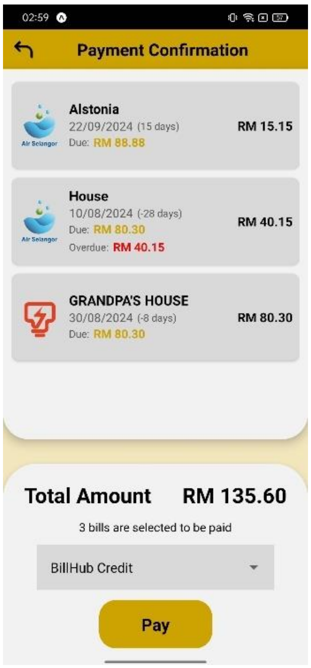
GRANDPA'S HOUSE
30/08/2024 (-8 days)
Due: RM 80.30
RM 80.30
Edit ☑

Total Amount RM 135.60
3 bills are selected

Pay Now

Figure 5.27: Payment Page

Figure 5.27 shows “Payment” module, which allows users to select bills for payment. Besides, users can also edit the amount to be paid.



02:59

Payment Confirmation

Alstonia
22/09/2024 (15 days)
Due: RM 88.88
RM 15.15

House
10/08/2024 (-28 days)
Due: RM 80.30
Overdue: RM 40.15
RM 40.15

GRANDPA'S HOUSE
30/08/2024 (-8 days)
Due: RM 80.30
RM 80.30

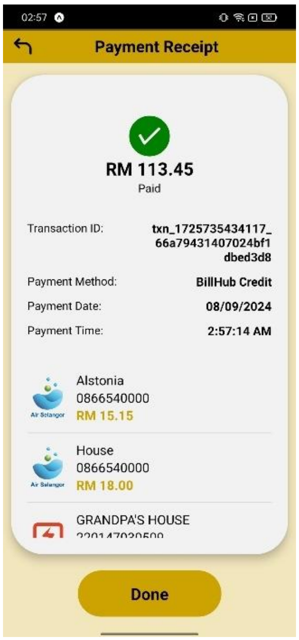
Total Amount RM 135.60
3 bills are selected to be paid

BillHub Credit

Pay

Figure 5.28: Payment Confirmation Page

Figure 5.28 shows the “Payment Confirmation” module which allows users to double check their selection and total payment amount.



02:57

Payment Receipt

RM 113.45
Paid

Transaction ID: txn_1725735434117_66a79431407024bf1dbed3d8

Payment Method: BillHub Credit

Payment Date: 08/09/2024

Payment Time: 2:57:14 AM

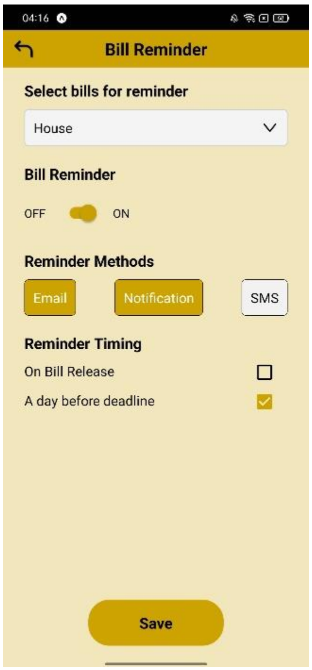
Alstonia
0866540000
RM 15.15

House
0866540000
RM 18.00

GRANDPA'S HOUSE
220147030500

Figure 5.29: Receipt Page

Figure 5.29 shows the “Receipt” module which generate a receipt for user after a payment is made.



04:16

Bill Reminder

Select bills for reminder

House

Bill Reminder

OFF ● ON

Reminder Methods

[✓ Email] [✓ Notification] SMS

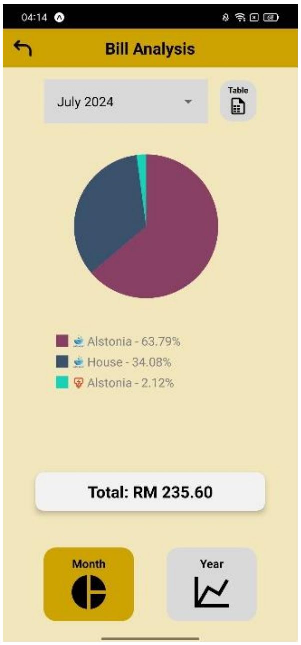
Reminder Timing

On Bill Release ☐

A day before deadline ☑

Figure 5.30: Bill Reminder Page

Figure 5.30 shows “Reminder” module which allows user to customize reminder for his registered bill.

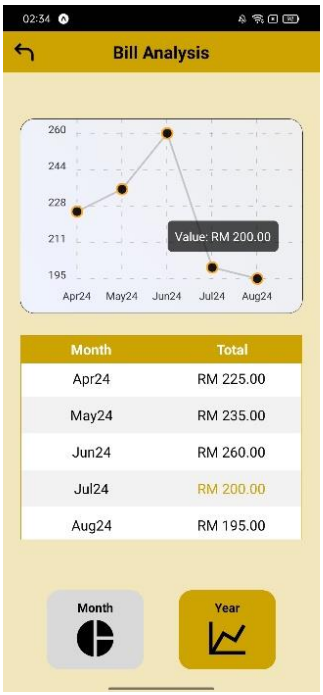


■ Alstonia - 63.79%
■ House - 34.08%
■ Alstonia - 2.12%

Total: RM 235.60

Figure 5.31: Bill Analysis Month Page

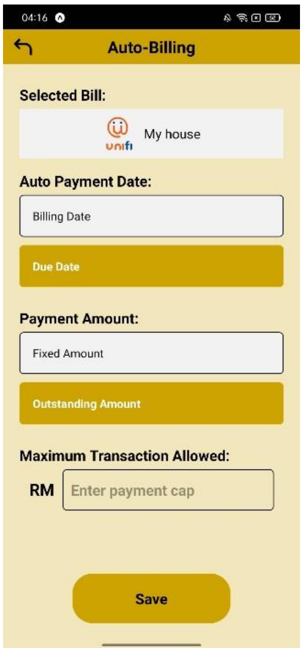
Figure 5.31 shows the "Bill Analysis" Month module, which shows the pie chart that visualizes the bills issued/ payment made in the selected month.



Month	Total
Apr24	RM 225.00
May24	RM 235.00
Jun24	RM 260.00
Jul24	RM 200.00
Aug24	RM 195.00

Figure 5.32: Bill Analysis Year Page

Figure 5.32 shows the "Bill Analysis" Year module, which shows the line chart that visualizes the changes of total bills issued / total payment made in a year.



04:16

Auto-Billing

Selected Bill:

My house

Auto Payment Date:

Billing Date

Due Date

Payment Amount:

Fixed Amount

Outstanding Amount

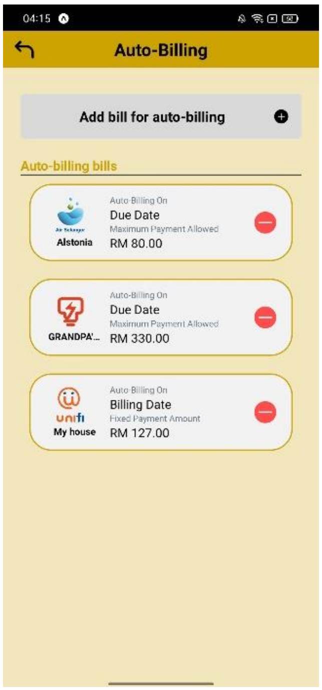
Maximum Transaction Allowed:

RM Enter payment cap

Save

Figure 5.33: Auto-Billing Setup Page

Figure 5.33 shows the “Auto-Billing” module that allows user to set up auto-billing for selected bill.



04:15

Auto-Billing

Add bill for auto-billing

Auto-billing bills

Alstonia
Auto-Billing On
Due Date
Maximum Payment Allowed
RM 80.00

GRANDPA'...
Auto-Billing On
Due Date
Maximum Payment Allowed
RM 330.00

My house
Auto-Billing On
Billing Date
Fixed Payment Amount
RM 127.00

Figure 5.34: Auto-Billing Page

Figure 5.44 shows the auto-billing bills that are successfully setup.

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 Introduction

This chapter focuses on the BillHub’s system implementation. It covers the frontend React-Native libraries used such as React Navigation, Async Storage and React Native Chart Kit. Besides, backend server implementation of Nodemailer, MongoDB, Stripe payment gateway and Agenda for automation.

6.2 React Navigation

React Navigation is a third-party library used to implement navigation functionalities in React Native projects. In BillHub, both stack and drawer navigators are implemented. By combining stack and drawer navigators, a nested navigation system has been created. The application allows users to navigate to the main pages through the drawer located on the home screen. Additionally, users can navigate to other pages by interacting with buttons on each page via the stack navigator.

Drawer navigation is one of the most common navigation systems used in mobile applications due to its hidden menu feature, which saves screen space for other content. A button with a menu icon is placed in the top left corner of the “Home” page's header, which triggers the opening of the drawer menu. When not in use, the drawer closes and remains hidden until the user triggers it again. The drawer navigation serves as the main navigation system, allowing users to navigate through different modules of BillHub.

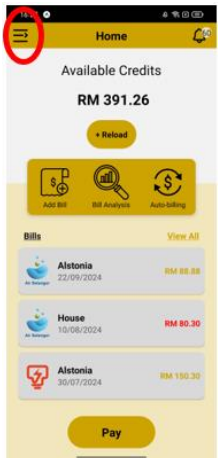


Figure 6.1: Open Drawer Navigation Buton

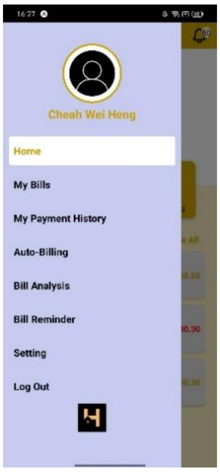


Figure 6.2: Drawer Navigation

In the nested navigation, the drawer navigator is nested inside a stack navigator as the initial screen of the stack navigator, while other pages are the other components. The hierarchy of the drawer navigator and stack navigators used are visualized in Figure 6.3 below.

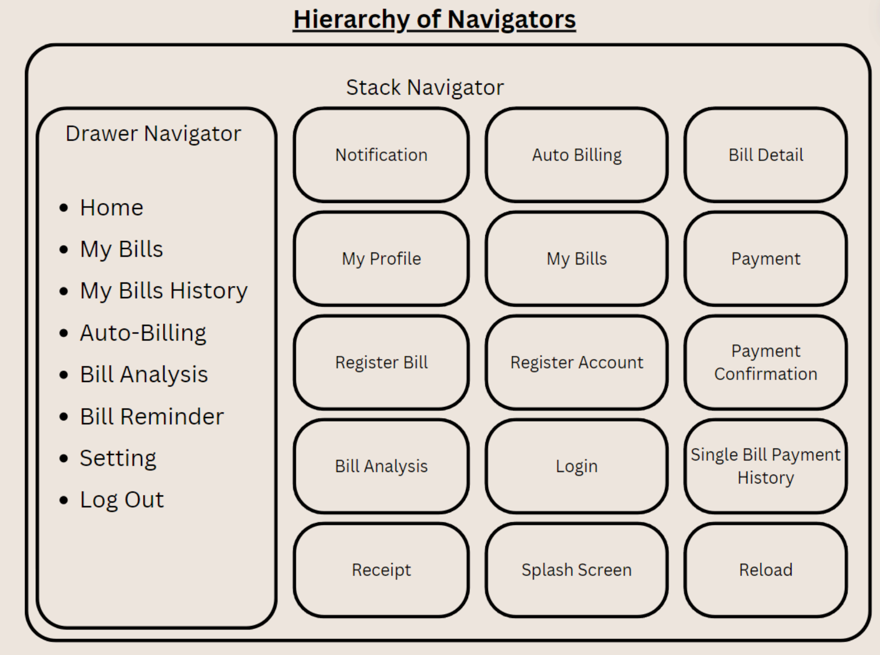


Figure 6.3: Hierarchy of nested navigators

6.3 Async Storage

Async Storage is a simple, unencrypted key-value storage system provided by React Native for storing small amounts of data locally on the user's device. In this project, Async Storage is used to store user data, such as authentication tokens. This token is essential for keeping the user logged into the application, even after they restart it.

When a user logs into their account, the system saves the authentication token in Async Storage. This token remains stored on the device even if the user closes the app, until it is manually removed through user logout. When the user reopens the app, the application retrieves the stored token from Async Storage and uses it to authenticate the user automatically. When the user logs out, the token is removed from Async Storage, effectively logging the user out of the app. It's important to note that the token will also be removed if the application is updated or reinstalled, requiring the user to log in again.

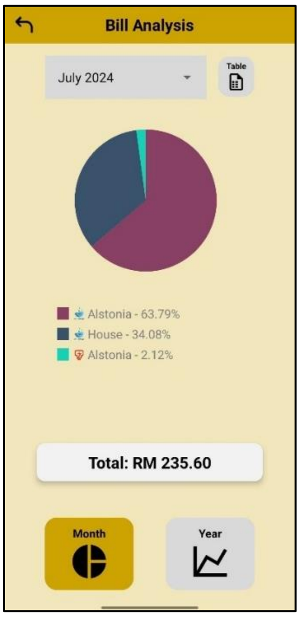
In short, the application of Async Storage allows the application to maintain user's logged-in state, eliminating the need for the user to manually log in every time they quit and restart the application.

6.4 React Native Chart Kit

In BillHub development, ‘react-native-chart-kit’ are used to create pie chart and line chart for informative bills and payments visualizations under the “Bill Analysis” module to help user understand their bills related financial data.

Library ‘react-native-chart-kit’ is a powerful charting library used in the application for quick and easy chart creation. This library simplifies the process of integrating chart into user interface by using its pre-built components. However, the charts built with this library are often less customizable compared to other complex libraries. However, the charts created are still responsive and fit its purposes.

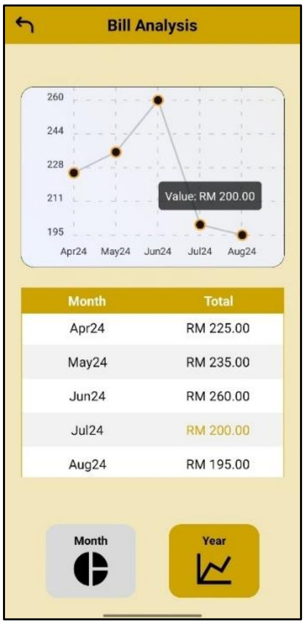
By using this library, two self-define components are created which are “BillPieChart” and “BillLineChart”, which accept array of bills issued or array of bills payments made as parameters. “BillPieChart” will break down the bills in a month and display data in Pie Chart according to its proportion and amount for user to understand their bills ratio. On the other hand, “BillLineChart” will display a line chart showing the comparison of total payments or bills issued amount in the past 6 months.



🟪 Alstonia - 63.79%
⬛ House - 34.08%
🟩 Alstonia - 2.12%

Total: RM 235.60

Figure 6.4: Pie Chart



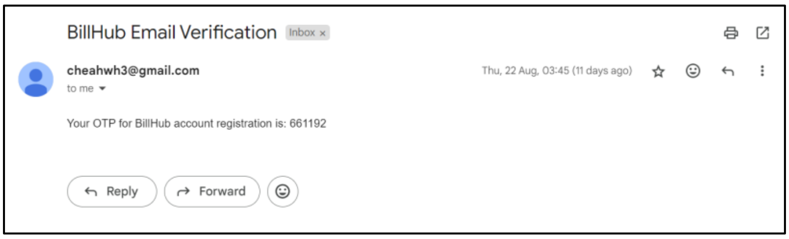
Month	Total
Apr24	RM 225.00
May24	RM 235.00
Jun24	RM 260.00
Jul24	RM 200.00
Aug24	RM 195.00

Figure 6.5: Line Chart

6.5 Node Mailer

Nodemailer is a module designed for Node.js to send emails through an application. It is a simple yet effective solution for handling email communication, it supports various email services and providing a straightforward API that integrates seamlessly with a Node.js server.

In the BillHub application, Nodemailer is used to send emails under specific conditions, such as email verification during account registration and send bills reminders. The Nodemailer service is run by the backend server and send email through an authenticated Gmail account using its email address and an application-specific password provided by Gmail.



BillHub Email Verification

cheahwh3@gmail.com
to me

Thu, 22 Aug, 03:45 (11 days ago)

Your OTP for BillHub account registration is: 661192

Figure 6.6: Email Sent by BillHub

6.6 MongoDB

To interact with MongoDB, the server is configured using the MVC (Model-View-Controller) architecture. This architecture separates the application into three core components. The models and schema that define the structure and attributes of the collections in MongoDB. Routes map API requests from the frontend to the appropriate controller functions. Routes define the endpoints that the frontend can call to perform various operations. The controllers handle the logic for processing these requests, interacting with the Models, and preparing responses. The frontend React Native (View) displays the data retrieved from the database and provides user interface to interact with users.

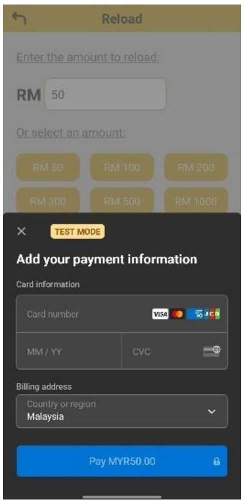
Additionally, Axios is utilized to facilitate communication between the frontend and backend. Axios is a promise-based HTTP client that sends HTTP requests from the application to the server, supporting methods such as PUT, POST, PATCH, GET, and DELETE. This enables the frontend to perform CRUD (Create, Read, Update, Delete) operations on the data by making requests to the backend API. Axios simplifies the process of making these requests and handling responses, streamlining the interaction between the React Native application and the server.

6.7 Stripe

Stripe is a widely used payment gateway that enables applications to accept online payments. Stripe offers a flexible platform for integrating payment functionalities into mobile applications including react-native application. Stripe supports a wide range of payment methods including debit cards, digital wallets and FPX (Financial Process Exchange).

To use Stripe in a React Native application, a Stripe account is registered at Stripe Official Site. The publishable key is used to collect payment information at client-side (frontend) of the application. On the other hand, the highly confidential secret key is kept on the server-side and is used to create charges and process the payments.

In BillHub, Stripe is integrated to handle payment processing for credit reloads, which allows users to transfer funds into BillHub’s business account. On the other hand, Stripe is also used to make bills payments when user choose to pay the bills with cards or online banking.



RM 50

Add your payment information

Card information

Billing address

Country or region: Malaysia

Figure 6.7: Stripe Payment Gateway

6.8 Agenda

Agenda is a Node.js library designed for scheduling jobs in applications. It provides a robust and flexible system for scheduling tasks and background jobs to perform certain actions at future event or predefined time.

Agenda handles two primary functionalities, bills reminder scheduling and auto-billing. When users configure the Reminder and Auto-Billing modules to send reminders and process payments on bill due dates, Agenda plays a crucial role. It schedules reminders for upcoming bill payments, including setting up jobs to send notifications or emails to users before their bills are due. The flexibility of Agenda allows reminders to be set for specific times or intervals, ensuring users receive timely reminder about their bills. Additionally, Agenda manages the auto-billing processes by scheduling tasks to automatically process bill payments based on user preferences, such as recurring payments or specific billing dates. This automation helps in efficiently managing financial transactions without manual intervention.

6.9 Conclusion

This chapter describes the integration of various tools and technologies used in the main modules of BillHub application. The tools are used to perform the main functionalities of the modules. Besides, it also describes the backend server architecture following the MVC pattern, which ensures a structured approach to manage backend server code. This chapter also explains the use of Axios for seamless communication between the frontend and backend through the API endpoints.

CHAPTER 7

SYSTEM TESTING

7.1 Introduction

The test results obtained from this project's testing will be shown and are described in this chapter. To ensure the functionality of the BillHub system, API testing, integrating testing and user acceptance testing are carried out.

7.2 API Testing

API testing is a crucial part of software testing, it involves testing application programming interfaces (APIs) directly to ensure they function as expected in terms of functionality, reliability, performance and security. In this project, postman was used for manual API testing, which allows developers to send HTTP requests and check the responses. The server.js file with relevant models, routes and controllers were tested. These server files are responsible for running a server that supports the BillHub application in performing CRUD operation on the database. Table 7.1 below indicates the summary of test cases and their status. Table 7.2 to Table 7.26 are the details descriptions of each of the test cases. After API testing, all test cases were passed as its actual results are the same as the expected results. In this case, the API should perform as expected without any error.

Table 7.1Summary of API test cases and its status

Test Case ID	Test Case Name	Status
TC001	Test Case of Creating User Account	Pass
TC002	Test Case of Successful User Login	Pass
TC003	Test Case of Failed User Login	Pass
TC004	Test Case of Credits Reload	Pass
TC005	Test Case of Credits Deduct	Pass
TC006	Test Case of Retrieving Billing Company List	Pass
TC007	Test Case of Viewing Notification	Pass
TC008	Test Case of Update Notification Seen	Pass

TC009	Test Case of Delete Notification	Pass
TC010	Test Case of Registering Bill	Pass
TC011	Test Case of Bill Retrieval	Pass
TC012	Test Case of Billing Company Bill Update	Pass
TC013	Test Case of Editing Bill's Nickname	Pass
TC014	Test Case of Update One Bill Reminder	Pass
TC015	Test Case of Update All Bill Reminder	Pass
TC016	Test Case of Deleting Registered Bill	Pass
TC017	Test Case of Issuing Bill	Pass
TC018	Test Case of Retrieving One Billing History	Pass
TC019	Test Case of Retrieving All Billing History	Pass
TC020	Test Case of Payment	Pass
TC021	Test Case of Retrieving One Payment History	Pass
TC022	Test Case of Retrieving All Payment History	Pass
TC023	Test Case of Setting Up Auto-Billing	Pass
TC024	Test Case of Retrieving Auto-Billing Settings	Pass
TC025	Test Case of Removing Auto-Billing	Pass

Table 7.2 Test Case of User Account Creation

Test Case ID	TC001	**Status**	Pass
Test Case Title	Test Case of Creating User Account		
Model	User		
Controller	userController		
Method	POST		
Endpoints Involved	http://${serverIPV4}:3000/users/register		
Test Scenario	**Test Data**	**Expected Result**	**Actual Result**
1. User submits account creation request after filling in personal information.	1. Name 2. idNumber 3. phoneNumber 4. email 5. password	1. JSON object with message and user created.	1. JSON object with message and user created.

Table 7.3 Test Case of Successful User Login

Test Case ID	TC002	**Status**	Pass
Test Case Title	Test Case of Successful User Login		
Model	User		
Controller	userController		
Method	GET		
Endpoints Involved	http://${serverIPV4}:3000/users/login		
Test Scenario	**Test Data**	**Expected Result**	**Actual Result**
1. User login with email/phone number and his password.	1. identifier 2. password	1. JSON object with login success message and user.	1. JSON object with login success message and user.

Table 7.4 Test Case of Failed User Login

Test Case ID	TC003	**Status**	Pass
Test Case Title	Test Case of Failed User Login		
Model	User		
Controller	userController		
Method	GET		
Endpoints Involved	http://${serverIPV4}:3000/users/login		
Test Scenario	**Test Data**	**Expected Result**	**Actual Result**
1. User login with email/phone number and his password.	1. identifier 2. password	1. JSON object with login failed message and error.	1. JSON object with login failed message and error.

Table 7.5 Test Case of Credits Reload

Test Case ID	TC004	**Status**	Pass
Test Case Title	Test Case of Credits Reload		
Model	User		
Controller	userController		
Endpoints Involved	http://${serverIPV4}:3000/users/${user._id}/credit		
Test Scenario	**Test Data**	**Expected Result**	**Actual Result**
1. User reloads the credits with payment gateway.	1. userId 2. amount	1. JSON object user with updated credit value	1. JSON object user with updated credit value

Table 7.6 Test Case of Credits Deduct

Test Case ID	TC005	**Status**	Pass
Test Case Title	Test Case of Credits Deduct		
Model	User		
Controller	userController		
Method	PATCH		
Endpoints Involved	http://${serverIPV4}:3000/users/${user._id}/deduct		
Test Scenario	**Test Data**	**Expected Result**	**Actual Result**
1. Users pays the bills with the BilHub credits.	1. userId 2. totalAmount	1. JSON object of user with updated credit value	1. JSON object of user with updated credit value

Table 7.7 Test Case of Retrieving Billing Company List

Test Case ID	TC006	**Status**	Pass
Test Case Title	Test Case of Retrieving Billing Company List		
Model	BillingCompany		
Controller	billingCompanyController		
Method	GET		
Endpoints Involved	http://${serverIPV4}:3000//billing-companies		
Test Scenario	**Test Data**	**Expected Result**	**Actual Result**
1. User wants to register a bill. 2. A billing companies list are retrieved for selection.	-	1. JSON data with a list of billing companies object	1. A list of billing companies object is retrieved

Table 7.8 Test Case of Viewing Notifications

Test Case ID	TC007	**Status**	Pass
Test Case Title	Test Case of Viewing Notifications		
Model	Notification		
Controller	notificationController		
Method	GET		
Endpoints Involved	http://${serverIPV4}:3000/notifications/${user._id}		
Test Scenario	**Test Data**	**Expected Result**	**Actual Result**
1. User wants to view his notification, The notifications are retrieved from database.	1. userId	1. JSON data of notification list with the provided userId	1. A JSON data, list of notifications with the userId provided

Table 7.9 Test Case of Update Notification Seen

Test Case ID	TC008	**Status**	Pass
Test Case Title	Test Case of Update Notification Seen		
Model	Notification		
Controller	notificationController		
Method	PATCH		
Endpoints Involved	http://${serverIPV4}:3000/notifications/${notificationId}/seen		
Test Scenario	**Test Data**	**Expected Result**	**Actual Result**
1. User clicks in his notification to view the notification detail.	1. notificationId	1. JSON data of notification object with updated “seen” value to true.	1. JSON data of notification object with updated “seen” value to true.

Table 7.10 Test Case of Delete Notification

Test Case ID	TC009	**Status**	Pass
Test Case Title	Test Case of Delete Notification		
Model	Notification		
Controller	notificationController		
Method	DELETE		
Endpoints Involved	http://${serverIPV4}:3000/notifications/${notificationId}/delete		
Test Scenario	**Test Data**	**Expected Result**	**Actual Result**
1. User clicks in his notification to view the notification detail. 2. User deletes the notification viewed.	1. notificationId	1. JSON data of “Notification successfully deleted” message	1. JSON data of “Notification successfully deleted” message

Table 7.11 Test Case of Registering Bill

Test Case ID	TC010	**Status**	Pass
Test Case Title	Test Case of Registering Bill		
Model	Bill		
Controller	billController		
Method	POST		
Endpoints Involved	http://${serverIPV4}:3000/bills		
Test Scenario	**Test Data**	**Expected Result**	**Actual Result**
1. User selects a billing company. 2. User enters the account number of his bill 3. User enter a name for his bill. 4. User submits the request to register bill.	1. userId 2. billingCompanyId 3. accountNumber 4. nickname	1. JSON data of savedBill	1. JSON data of savedBill

Table 7.12 Test Case of Bill Retrieval

Test Case ID	TC011	**Status**	Pass
Test Case Title	Test Case of Bill Retrieval		
Model	Bill		
Controller	billController		
Method	GET		
Endpoints Involved	http://${serverIPV4}:3000/bills/{userId}		
Test Scenario	**Test Data**	**Expected Result**	**Actual Result**
1. User login his account. 2. Bill lists are automatically retrieved from database.	1. userId	1. JSON data of a list of bills	1. JSON data of a list of bills

Table 7.13 Test Case of Billing Company Bill Update

Test Case ID	TC012	**Status**	Pass
Test Case Title	Test Case of Billing Company Bill Update		
Model	Bill		
Controller	billController		
Method	PUT		
Endpoints Involved	http://${serverIPV4}:3000/bills/{billId}		
Test Scenario	**Test Data**	**Expected Result**	**Actual Result**
1. Billing Company wants to update the bill registration status.	1. billingDate 2. dueDate 3. outStandingAmount 4. billOwner 5. status 6. billingAmount	1. JSON data of updated Bill object	1. JSON data of updated Bill object

Table 7.14 Test Case of Editing Bill’s Nickname

Test Case ID	TC013	**Status**	Pass
Test Case Title	Test Case of Editing Bill’s Nickname		
Model	Bill		
Controller	billController		
Method	PATCH		
Endpoints Involved	http://${serverIPV4}:3000/bills/ {billId}/nickname		
Test Scenario	**Test Data**	**Expected Result**	**Actual Result**
1. After user registered a bill, he wants to update the name of the bill. 2. User didn’t assign a name for the registered bill, and he would like to assign it afterward.	1. billId 2. nickname	1. JSON data of Bill object updated with new nickname	1. JSON data of Bill object updated with new nickname

Table 7.15 Test Case of Update One Bill Reminder

Test Case ID	TC014	**Status**	Pass
Test Case Title	Test Case of Update One Bill Reminder		
Model	Bill		
Controller	billController		
Method	PATCH		
Endpoints Involved	http://${serverIPV4}:3000/bills/ {billId}/reminder		
Test Scenario	**Test Data**	**Expected Result**	**Actual Result**
1. After user registered a bill, he wants to set up reminder for that bill.	1. billId 2. onOff 3. method 4. time	1. JSON data of Bill object updated with reminder setting	1. JSON data of Bill object updated with reminder setting

Table 7.16 Test Case of Update All Bill Reminder

Test Case ID	TC015	**Status**	Pass
Test Case Title	Test Case of Update All Bill Reminder		
Model	Bill		
Controller	billController		
Method	PATCH		
Endpoints Involved	http://${serverIPV4}:3000/bills/ {userId}/reminder		
Test Scenario	**Test Data**	**Expected Result**	**Actual Result**
1. After user registered bills, he wants to set up reminder for all his bills.	1. userId 2. onOff 3. method 4. time	1. JSON data of “Bill reminders updated successfully” message and a list of bills object updated with reminder setting	1. JSON data of “Bill reminders updated successfully” message and a list of bills object updated with reminder setting

Table 7.17 Test Case of Deleting Registered Bill

Test Case ID	TC016	**Status**	Pass
Test Case Title	Test Case of Deleting Registered Bill		
Model	Bill		
Controller	billController		
Method	DELETE		
Endpoints Involved	http://${serverIPV4}:3000/bills/${billId}		
Test Scenario	**Test Data**	**Expected Result**	**Actual Result**
1. After user registered bills, he wants to remove it from his account.	1. billId	1. JSON data of updated list of bills object updated with successful deletion message.	1. JSON data of updated list of bills object updated with successful deletion message.

Table 7.18 Test Case of Issuing Bill

Test Case ID	TC017	**Status**	Pass
Test Case Title	Test Case of Issuing Bill		
Model	Bill		
Controller	billController		
Method	PATCH		
Endpoints Involved	http://${serverIPV4}:3000/bills/${billId}		
Test Scenario	**Test Data**	**Expected Result**	**Actual Result**
1. Billing company wants to issue a bill to the person who registered the bill.	1. billId 2. billingAmount 3. billingDate 4. dueDate 5. outStandingAmount	1. JSON data of updated bill object with.	1. JSON data of updated bill object with.

Table 7.19 Test Case of Retrieving One Billing History

Test Case ID	TC018	**Status**	Pass
Test Case Title	Test Case of Retrieving One Billing History		
Model	BillingHistory		
Controller	billingHistoryController		
Method	GET		
Endpoints Involved	http://${serverIPV4}:3000/billing-history/bill/${billId}		
Test Scenario	**Test Data**	**Expected Result**	**Actual Result**
1. User wants to view the billing history of a single bill	1. billId	1. JSON data of a list of billing history of a bill.	1. JSON data of a list of billing history of a bill.

Table 7.20 Test Case of Retrieving All Billing History

Test Case ID	TC019	**Status**	Pass
Test Case Title	Test Case of Retrieving All Billing History		
Model	BillingHistory		
Controller	billingHistoryController		
Method	GET		
Endpoints Involved	http://${serverIPV4}:3000/billing-history/user/${userId}		
Test Scenario	**Test Data**	**Expected Result**	**Actual Result**
1. User wants to view bill analysis, a list of billing histories under his account will be retrieved.	1. userId	1. JSON data of a list of billing histories of bills under the same user.	1. JSON data of a list of billing histories of bills under the same user.

Table 7.21 Test Case of Payment

Test Case ID	TC020	**Status**	Pass
Test Case Title	Test Case of Payment		
Model	PaymentHistory		
Controller	paymentHistoryController		
Method	POST		
Endpoints Involved	http://${serverIPV4}:3000/payment-history		
Test Scenario	**Test Data**	**Expected Result**	**Actual Result**
1. User pays his registered bills.	1. Array of paymentHistories containing billId, paymentTime, paymentDate, paymentMethod and paymentAmount.	1. JSON data of a list of payment histories created	1. JSON data of a list of payment histories created

Table 7.22 Test Case of Retrieving One Payment History

Test Case ID	TC021	**Status**	Pass
Test Case Title	Test Case of Retrieving One Payment History		
Model	PaymentHistory		
Controller	paymentHistoryController		
Method	GET		
Endpoints Involved	http://${serverIPV4}:3000/payment-history/bill/${billId}		
Test Scenario	**Test Data**	**Expected Result**	**Actual Result**
1. User pays his registered bills.	1. billId	1. JSON data of a list of billing history of a bill.	1. JSON data of a list of billing history of a bill.

Table 7.23 Test Case of Retrieving All Payment Histories

Test Case ID	TC022	**Status**	Pass
Test Case Title	Test Case of Retrieving All Payment Histories		
Model	PaymentHistory		
Controller	paymentHistoryController		
Method	GET		
Endpoints Involved	http://${serverIPV4}:3000/payment-history/bill/${userId}		
Test Scenario	**Test Data**	**Expected Result**	**Actual Result**
1. User views his payment histories, payment histories for all bills are retrieved. 2. User views bill analysis, payment histories for all bills are retrieved for analysis.	1. userId	1. JSON data of a list of payment histories of bills under the same user.	1. JSON data of a list of payment histories of bills under the same user.

Table 7.24 Test Case of Setting Up Auto-Billing

Test Case ID	TC023	**Status**	Pass
Test Case Title	Test Case of Setting Up Auto-Billing		
Model	AutoBilling		
Controller	autoBillingController		
Method	POST		
Endpoints Involved	http://${serverIPV4}:3000/auto-billing		
Test Scenario	**Test Data**	**Expected Result**	**Actual Result**
1. User wants to set up auto-billing for one of the bill he selected.	1. billId 2. userId 3. autoPaymentDate 4. paymentAmount 5. amount	1. JSON data of a newAutoBilling Object created	1. JSON data of a newAutoBilling Object created

Table 7.25 Test Case of Retrieving Auto-Billing Setting

Test Case ID	TC024	**Status**	Pass
Test Case Title	Test Case of Retrieving Auto-Billing Setting		
Model	AutoBilling		
Controller	autoBillingController		
Method	GET		
Endpoints Involved	http://${serverIPV4}:3000/auto-billing/user/${userId}		
Test Scenario	**Test Data**	**Expected Result**	**Actual Result**
1. User wants to view the selected bills for auto-billing and the respective auto-billing customization.	1. userId	1. JSON data of an array of auto-billing objects	1. JSON data of an array of auto-billing objects

Table 7.26 Test Case of Removing Auto-Billing

Test Case ID	TC025	**Status**	Pass
Test Case Title	Test Case of Removing Auto-Billing		
Model	AutoBilling		
Controller	autoBillingController		
Method	DELETE		
Endpoints Involved	http://${serverIPV4}:3000/auto-billing/${autoBillingId}		
Test Scenario	**Test Data**	**Expected Result**	**Actual Result**
1. User wants to remove auto-billing for a specific bill.	1. autoBillingId	1. JSON data of a successful deletion message	1. JSON data of a successful deletion message

7.3 Usability Test

7.3.1 Description of Usability Test

As mentioned in Chapter 1 of the report, some bill payers find it difficult to pay bill online, especially older individuals and those with limited exposure to technology. Thus, a usability test was conducted to evaluate BillHub’s “ease of learning” and “ease of use” for managing bills. This usability testing is essential to ensure that the system meets the target users’ need and that it effectively performs the basic operations required to solve the bill payment problem. The final outcomes of the usability test included users’ feedback and a scorecard evaluating the features of the BillHub, the average SUS score was calculated.

For this usability test, a few guidelines and rules are set and were strictly followed by the testers to ensure the consistency of the test. First, participants are required to read through the test scenarios. By placing themselves in the perspective of user in the scenario, they were instructed to complete the task outlined in the scenarios under the observation of the developer. If they encountered any problem during completion of the test scenarios, minimal assistance was provided by the developer. The developer will also monitor and document the entire usability test process. After the usability testing, the participants were required to fill out the satisfaction survey form based on their honest opinions.

In this usability test, five bill payers are invited as participants for the test. They were asked to carry out the usability test by following the steps described in Table 7.27. Upon completing the test, each of them filled out the user satisfaction survey form as attached in Appendix B, the result of the surveys was attached in Appendix C and were analysed and the average SUS score of BillHub was calculated.

7.3.2 Test Scenarios of Usability Test

Table 7.27 Test Scenarios of Usability Test

No	Test Scenario Title	Test Scenario Description
1	Register an account	Imagine you are a new user who wants to use BillHub to manage your bills. Before able to access its features, you need to create an account. After opening the BillHub application for the first time, what steps would you take to register an account?
2	Login an account	You've successfully registered an account for BillHub and now want to access your account. Upon launching the BillHub app, you need to log in. What steps you would take to enter your credentials to log in?
3	Register bills	After successfully logging in to BillHub, you want to add your monthly bills (e.g., electricity, water, internet) to the platform for tracking and payment. What actions you would take to register these bills under your account, including providing billing details such as account number?
4	Edit Bill Nickname	You’ve added several bills to BillHub, but you haven’t set a name for one of your registered bills or have set it wrongly. Now you want to give the bill a recognizable nickname, such as "Home Electricity" or "Home Internet". Hence, what steps you would take to edit the nickname of one of your bills to make it easier to identify?

5	Reload Credits	You want to make a bill payment, but first, you need to reload your BillHub account with sufficient credits. Hence, what steps you would take to add credits to your account?
6	Make Bill Payment	You've registered your bills and loaded your account with credits. Now it’s time to pay one of your bills. Hence, how you would navigate through the app to select a specific bill, modify the payment amount and confirm the payment?
7	Set Bill Reminder	You want to avoid missing any bill payments, so you decide to set reminders for upcoming due dates. Hence, what steps you would take to set a bill reminder?
8	Set Auto-Billing	To simplify your payments, you want to enable auto-billing for one of your recurring bills. Hence, how you would set up auto-billing to automatically pay the bill on its due date?
9	View Bill Analysis	You have received some bills issued by billing companies and made some payment. Now, you’re interested in reviewing how much you’ve spent on your bills over the past few months. Hence, how you would navigate the BillHub interface to access your bill analysis, view trends, and track your total spending for a given period?

10	View Bill History	You want to check the payment history for a specific bill to see all past transactions including bills issued and payment made. Hence, what are the steps you would take to find this information within BillHub?

7.3.3 Results of Usability Test

After completing the usability test, the user satisfaction survey forms were collected from all the participants and attached at Appendix C. The results from each of the forms were extracted and displayed in the Table 7.28. The average SUS score obtained by this project is 92.2. This score indicates that bill payers can learn and use BillHub system easily without having to spend extra effort and time to research.

Table 7.28 Score for Usability Test Cases

ID	**Score for each test**																				**Total**
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	
P1	4	5	5	5	5	5	5	5	4	4	5	5	5	5	4	5	5	4	5	5	95
P2	4	5	5	5	5	4	5	5	4	4	4	3	5	5	5	5	5	4	4	5	91
P3	5	4	5	5	5	5	5	4	5	4	5	3	5	5	5	3	5	4	5	5	92
P4	4	5	5	5	5	3	5	5	2	3	5	4	5	5	4	4	5	5	5	5	89
P5	5	5	5	5	5	5	4	5	3	4	5	5	4	5	5	5	5	4	5	5	94
Average SUS score																					92.2

Moreover, the forms used to record user satisfaction also included subjective questions which inquired the users' opinion. The participants were asked regarding their opinions on which features and functionalities of the BillHub. The functions that were most helpful or most bothersome were provided by the testers. The views expressed were analysed and summarized as illustrated in the Table 7.29. Besides, the participants were also asked to provide idea for enhancement for BillHub if they would like to enhance the application. The comments provide understanding on what they prefer or areas they wish to improve. The enhancement ideas were shown in Table 7.30. These suggestions provided by participants are taken into consideration as parts of the future improvement of the implemented system.

Table 7.29 Liked the most feature and Like the least feature

Liked the most	**Like the least**
Automatically analyse the bills and payments, helps to understand the total expenses on bill.	Too many customizations, increases the learning curve.
The function of auto-billing can help me in paying repeating bills such as telecommunication bills which are fixed amount every month. Meanwhile the function of billing reminder helps me prevent from paying the bills late and avoid penalties.	The overall user interface design can be improved.
I am paying multiple bills every month, I really like the feature of paying multiple bills in one transaction, it can greatly save up my time.	No identity verification during the registration of NRIC process.
Able to pay multiple bills at once	Lack of auto reload functionality, this will require user to reload manually every month.
Able to see the due dates and outstanding amount of each bill at one page, enable bill payer to estimate the total amount have to be spent on bill payment.	Unable to set auto-billing with credit card.

Table 7.30 Enhancement Idea and Comments

Enhancement Idea	**Additional Comments**
Provide default reminder and auto-billing settings for those who do not want customization.	Overall, this is a very convenient application.
Include points or coins mechanism, which rewards user points or coins to redeem rewards as they pay bills using BillHub.	Improve UI and add transition animation.
Synchronize with billing company to add a feature that allow bill payers to view the usage of the bills.	-
Add auto reload when remaining balance is lower than a specific value	Convenient application for users with many bills
Enable auto-billing with other payment methods	Good, try to explore and add more E-wallet likes features.

7.4 User Acceptance Testing (UAT)

User acceptance test was conducted face-to-face with bill payers using the development device. This was done to validate the operational flow according to the UAT steps and ensured that any error encountered or comments given by testers were properly documented. Five testers were invited to participate in the UAT. The testers were required to perform the UAT according to the steps provided in the UAT test form as attached in Appendix D. During the UAT process, the developer acted as observer to record any comments and errors encountered. Additionally, if the testers raised any questions, the observer would offer minimal assistance when necessary. On the other hand, the observer is also responsible for noting down the start time and end time of the UAT process.

According to the results of UAT attached in Appendix E, there were three failed test cases where two are due to the same reason. There was also an error discovered by one of the participants when he repeated the steps for multiple times. The related test case with the error information and action taken to resolve it was recorded in Table 7.31.

Table 7.31 UAT Summary Table

Test Case ID	Test Case Title	Status	Error Occurred	Action Taken
UAT014	Log Out	Fail	Trigger an error: Unable to access “._id” of undefined.	The "useEffect" hook, which triggered functions when the user._id changed, has been removed and replaced with other logic. As a result, there is no longer any event listener for user._id, such as during user logout.
UAT006	Pay Bill	Fail	When trying to edit the payment amount, the keyboard immediately disappears, and one of the bill's checkboxes cannot be selected.	Instead of using KeyboardAvoidingView with FlatList, replace it with KeyboardAwareScrollView and mapping functions. KeyboardAvoidingView is used to manage keyboard appearance and avoid layout shifting while also avoiding issues with VirtualizedLists.

According to the Table 7.31, the errors detected during UAT were clearly documented and fixed using the methods explained.

In summary, the UAT was successful, with minimal intervention required from developers. The testers completed all UAT test cases and effectively validate the functionality of BillHub.

7.5 Conclusion

After completing API testing, usability testing, and User Acceptance Testing (UAT), the BillHub application has demonstrated strong functionality, user experience, and reliability across all key modules. These testing efforts have ensured that the application meets both functional and non-functional requirements, while also fulfilling its primary goal of providing a seamless and consolidated bill management platform. In conclusion, BillHub has passed most of the tests, and minor test failures were reworked to solve the bugs found. Additionally, user recommendations and feedback were considered for modifications and enhancements, while some suggestions were reserved for future improvements. The positive results from the testing phases indicate that BillHub is well-positioned and ready to deliver value to society by enabling bill payers to manage their bills effortlessly.

CHAPTER 8

CONCLUIONS AND RECOMMENDATIONS

8.1 Introduction

This project took approximately nine months to complete, starting from the initial idea of "Bills Consolidation." It aims to offer a future solution to bill management challenges, with the fundamental concept of simplification. As more subscription services emerge, people are paying more bills than before, leading to an increase in the number of applications and platforms needed to manage and pay them. At the start of the project, thorough planning was conducted to outline the solution, define objectives, scope, and gather requirements. To further analyze current bill payment solutions, a literature review, surveys, and an evaluation of similar applications were carried out. Afterward, a project schedule was developed, including a work breakdown structure and a Gantt chart. Before development, the system prototype was designed through interface flow diagrams, entity relationship diagrams, and high-fidelity prototype designs.

Once all preparations were completed, development began using the incremental model, where analysis iterations were repeated for each module of BillHub. Each iteration revisited the requirements for a better understanding of the server structure and frontend design, followed by coding and testing. This cycle continued until all modules were developed. Throughout the process, code was regularly pushed to GitHub for backup. The sequence of module development was started with Register Account and Login, followed by Home Page, Register Bill, View Bill, Payment, History, Bill Analysis, Bill Reminder, Auto-Billing, Reload, and finally Notification. During development, the server backend API was tested, as mentioned in Chapter 7, to ensure it performed as expected and enabled seamless communication between the frontend application and the database.

In the final phase of the project, usability testing and UAT were conducted to assess usability, ease of use, user satisfaction, and the acceptance rate of BillHub. The overall SUS score achieved was 92.2, indicating that BillHub functions well, provides a good user experience, and is easy to learn. However, a few errors were identified by participants during UAT, which triggered error messages and caused some test cases to fail. These issues were addressed and documented in the report, and fixes were made after UAT. Additionally, user comments and suggestions for enhancements were recorded and considered for future improvements, though some ideas may not be practical at this time and are being kept for consideration in future updates.

8.2 Objectives Achievement

This project is initiated with a goal and three objectives. During the end of this project, each of the objectives stated in Chapter 1 were achieved.

The first objective of this project was to examine the existing bill presentment and payment processes by reviewing similar applications in the bill payment and management systems. This objective was achieved during the planning phase, as explained in Chapter 2.3. A total of four types of bill payment applications were reviewed: e-commerce applications, e-wallet applications, banking applications, and applications provided by billing companies. One representative application was selected for each type. Further breakdowns of the introduction, steps to pay bills, pros, and cons of each application were analyzed and documented. After the review, the development of the BillHub application utilized the results of this analysis. The benefits and advantages of the reviewed applications were incorporated into BillHub, while their disadvantages were avoided. As a result, BillHub is a combination of multiple bill payment applications, incorporating their strengths and avoiding their weaknesses.

The second objective of this project was to create a centralized bill management system that is user-friendly and simplifies the process of managing multiple bills from various service providers or billing companies. To achieve this, BillHub was developed based on the idea of "Bills Consolidation," which combines bills that were previously managed through different applications into one. Additionally, BillHub was designed with the concept of "Simplification," featuring a simple user interface that allows users to navigate the application without needing tutorials or expert guidance. The payment module was created with the idea of "one payment, all bills paid," enabling users to pay all their bills in a single transaction, reducing the need for repetitive tasks. Furthermore, with the automatic bill analysis feature, users can easily understand their bills without performing any technical or manual analysis. To confirm the achievement of this objective, usability and user acceptance tests were conducted to ensure the application's ease of use.

The third objective of this project was to evaluate the bill management system (BillHub) through usability testing and a user acceptance test (UAT). This objective was achieved as the application successfully passed both the SUS test and UAT. These tests were conducted during the closing phase of the project, with the results thoroughly documented in Chapter 7. In summary, BillHub passed the SUS test with an average score of 92.2, confirming the application's ease of use. Additionally, most of the test cases in UAT were passed without issues. However, two test cases did fail, but these were promptly addressed, and the bugs were fixed after the UAT. These tests indicate that BillHub provides users with valuable features to help them efficiently manage and pay their bills.

8.3 Project Limitations

Although all the project objectives were successfully achieved, there are some project limitations worth mentioning for future enhancements. Some of these limitations were discovered during development, while most were identified during usability testing and UAT.

The first limitation is the inability to view bill usage. For example, users cannot view the exact volume of water used each day through the BillHub application. If users wish to see detailed usage information, they must log in to the "Air Selangor" application. This is because different billing services require unique usage review pages that would need significant customization and cooperation with the billing companies. Since there is no direct connection with the billing companies at this time, implementing this module is challenging.

The second limitation is the inability to verify user identity, despite collecting users' NRIC numbers. The NRIC number is collected to simplify the approval process for bill registration, especially when billing companies detect that the registration request comes from the bill owner. However, lack of cooperation with the government due to the absence of business registration, user identity verification is not currently possible.

The third limitation is the lack of automation beyond the scope of this project, specifically in areas like auto-reloading credits and auto-billing via credit cards which involve other financial parties. The reasons are similar to the previous limitation, without a formal business entity and contracts with banking partners, auto-billing through bank accounts, credit cards, or even e-wallets is not currently feasible.

8.4 Recommendations for Future Work

To address the limitations identified in this project and enhance BillHub's capabilities, several key areas for future work are listed. First, the integration of detailed usage tracking for various services, such as water, electricity, and internet consumption. This would require collaboration with billing companies to gain access to usage data. Establishing partnerships with service providers and creating specific usage review pages would allow users to monitor their consumption directly through BillHub, without needing to switch between different applications.

Another recommendation is to implement identity verification to streamline the registration process and enhance security. While BillHub currently collects users' NRIC numbers for registration purposes, future work should explore partnerships with government agencies to enable secure identity verification. This would likely require BillHub to register as a formal business entity and gain access to relevant government databases, ensuring a more seamless and secure user experience.

A key step to expand BillHub's functionality is establishing direct business partnerships with billing companies. By registering BillHub as a legal business and entering into agreements with these companies, it would be possible to integrate real-time updates and improve overall bill management. These partnerships could also facilitate future features like direct payments, enable BillHub to provide a more comprehensive solution.

Automation is another area where BillHub can grow. Future development should focus on implementing features like auto-reloading credits and auto-billing through bank accounts, credit cards, or e-wallets. To make this possible, registering BillHub as a business and forming partnerships with financial institutions are necessary steps. This will allows automate bill payments and other financial transactions, further simplifying the user experience.

In addition, it would be beneficial to provide default reminder and auto-billing settings for users who prefer simplicity over customization. Some users may not want to spend time configuring detailed settings for bill reminders or auto-payments. By offering a straightforward, default configuration that can be activated with minimal input, BillHub can provides an efficient, hassle-free experience while maintaining the option for those who want more control over their preferences.

Lastly, BillHub can also introduce a rewards system, such as coins or points, could incentivize users to pay their bills through BillHub. This could be effective if BillHub collaborates with other businesses, such as travel agencies, cinemas, and retail partners. Users could earn points or coins for each bill payment, which they can redeem for discounts or rewards at partner companies. This feature would not only encourage more frequent use of the platform but also enhance BillHub's appeal by offering tangible benefits to users.

By pursuing these recommendations, BillHub can be improved into a more powerful, fully integrated bill management solution that meets the diverse needs of its users while offering added convenience and value.

REFERENCES

Almeida, J., 2022. Pros and Cons of Xamarin App Development. *Distant Job*, [blog] 12 April. Available at: <https://distantjob.com/blog/xamarin-app-development/>.

Berger, J., 2022. Customer-centric bill payment solutions for financial institutions. Fintech Insights, [online] Available at: <https://www.fisglobal.com/insights/the-rise-of-customer-centric-bill payment-solutions> [Accessed 22 April 2024].

Dudekula, R., 2022. YouGov: Touch 'n Go eWallet ranked top payment brand in Malaysia. *Marketing-Interactive,* [online] Available at: <https://www.marketing-interactive.com/touch-n-go-ewallet-top-payment-brand-malaysia>[Accessed 28 March 2024].

FocusM, 2024. *Uproar over health premium hike as Bank Negara keeps mum.* [online] Malaysia: FocusM. Available at: https://focusmalaysia.my/uproar-over-health-premium-hike-as-bank-negara-keeps-mum/#:~:text=For%20the%20second%20time%20in,19%20pandemic%20and%20economic%20downturn. [Accessed 16 April 2024].

Humphrey, D.B., Willesson, M., Lindblom, T, and Bergendahl, G., 2003. What Does it Cost to Make a Payment?. *Review of Network Economics*, [e-journal] 2(2). https://doi.org/10.2202/1446-9022.1024.

Jayasimha, K.R. and Nargundkar, R., 2006. ADOPTION OF SELF SERVICE BILL PAYMENT TECHNOLOGIES (SSBPTS): A CONCEPTUAL MODEL. *Journal of Services Research*, [e-journal] 6(2). Available through: UTAR Library website <https://openurl.ebsco.com/EPDB%3Agcd%3A5%3A8944933/detailv2?sid=ebsco%3Aplink%3Ascholar&id=ebsco%3Agcd%3A23654499&crl=c.> [Accessed 22 march 2024].

Kagan, J., 2021. *Automatic Bill Payment: What it is, How it Works, Pros and Cons*. https://www.investopedia.com/terms/a/automatic-bill-payment.asp#:~:text=An%20automatic%20bill%20payment%20occurs,to%20pay%20their%20monthly%20bills.

Kamienski, N., 2023. React Native Pros and Cons. *React & Front-end development blog*, [blog] 26 May, Available at: <https://pagepro.co/blog/react-native-pros-and-cons/.> [Accessed 05 April 2024].

Khatun, A., 2015. *Kiosk Transaction Management & Utility Bills Collection System*. Degree of Master. East West University. Available at: http://103.133.167.11:8080/handle/123456789/1554. [Accessed 26 March 2024].

Laukkanen, T. and Lauronen, J., 2005. Consumer value creation in mobile banking services. *International Journal of Mobile Communications*, 3(4), p.325.

Mantel, B., 2001. Why do consumers pay bills electronically? An empirical analysis, *Social Science Research Network* [Preprint]. https://doi.org/10.2139/ssrn.248409.

Montaño, D., 2024. Why Use Flutter: Pros and Cons of Flutter App Development. *Waverley*, [blog] 19 January. Available at: https://waverleysoftware.com/blog/why-use-flutter-pros-and-cons/#3.
[Accessed 02 April 2024]
Most popular mobile app development Frameworks for app developers. 2024.. Technostacks. [Blog] 2 January. Available at: <https://technostacks.com/blog/mobile-app-development-frameworks/.>
[Accessed 01 April 2024].

Oduor, R.A., 2012. *Adoption of the automated bill enquiry and bill payment system by customers of the Kenya Power and Lighting Company in Kisumu*.

Available at: http://erepository.uonbi.ac.ke/handle/11295/15114. [Accessed 21 April 2024].

Ray, A., Jain, M. and Ma, L., 2023. The impact of personality traits, barriers and gamification on Gen X continuance intention for mobile credit bill payment apps. *Journal of Financial Services Marketing*. https://doi.org/10.1057/s41264-023-00264-2

Sexton, S., 2015. *Automatic Bill Payment and Salience Effects: Evidence from Electricity Consumption*. The Review of Economics and Statistics, 97(2), pp. 229–241. https://doi.org/10.1162/rest_a_00465.

Staykova, K.S. and Damsgaard, J., 2016. Adoption of mobile payment platforms: Managing reach and range. *Journal of Theoretical and Applied Electronic Commerce Research*, 11(3), pp. 66–85. https://doi.org/10.4067/s0718-18762016000300006.

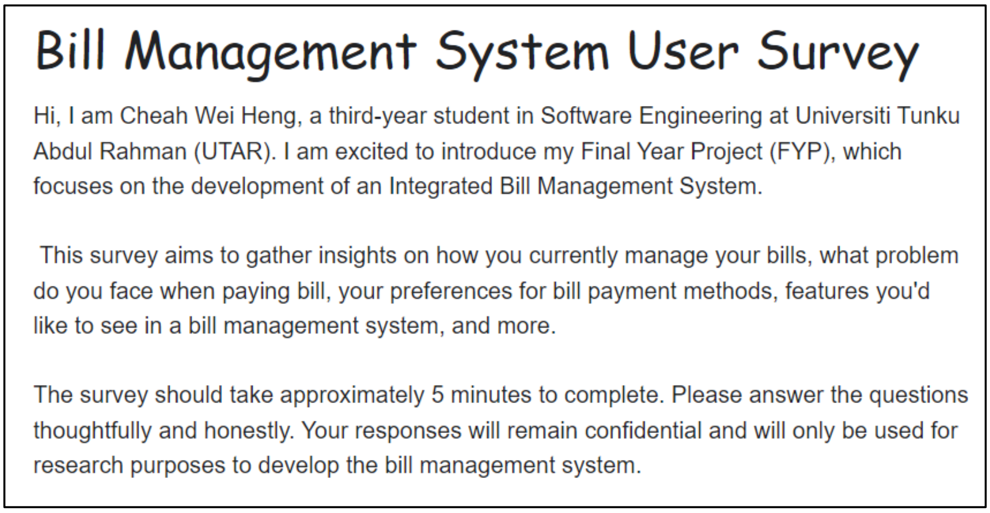
Sommerville, I., Columbus, B., New, I., San, Y., Upper, F., River, S., Cape, A., Dubai, T., Madrid, L., Munich, M., Montreal, P., Delhi, T., São, M.C., Sydney, P., Kong, H., Singapore, S. and Tokyo, T., 2011. SOFTWARE ENGINEERING. 9th ed. [online] <https://engineering.futureuniversity.com/BOOKS Available FOR Engineering-9th-Edition-by-Ian-Sommerville.pdf> [Accessed 14 Jul. 2021].

Yao, Y. and Li, L., 2016. Portfolio Insurance with a dynamic risk multiplier based on price fluctuation. *European Journal of Research and Reflection in Management Sciences*, 4(2).

APPENDICES

Appendix A: Survey Question

Figure A-1: Introduction page of the questionnaire survey form



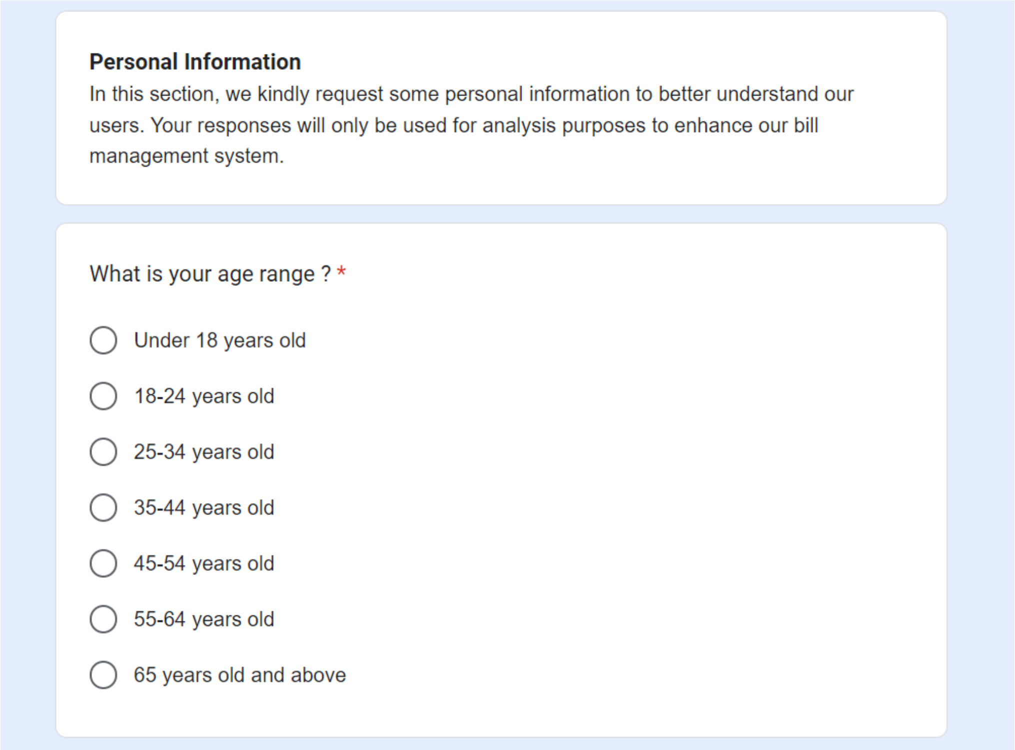
Bill Management System User Survey

Hi, I am Cheah Wei Heng, a third-year student in Software Engineering at Universiti Tunku Abdul Rahman (UTAR). I am excited to introduce my Final Year Project (FYP), which focuses on the development of an Integrated Bill Management System.

This survey aims to gather insights on how you currently manage your bills, what problem do you face when paying bill, your preferences for bill payment methods, features you'd like to see in a bill management system, and more.

The survey should take approximately 5 minutes to complete. Please answer the questions thoughtfully and honestly. Your responses will remain confidential and will only be used for research purposes to develop the bill management system.

Figure A-2: Section one of the survey

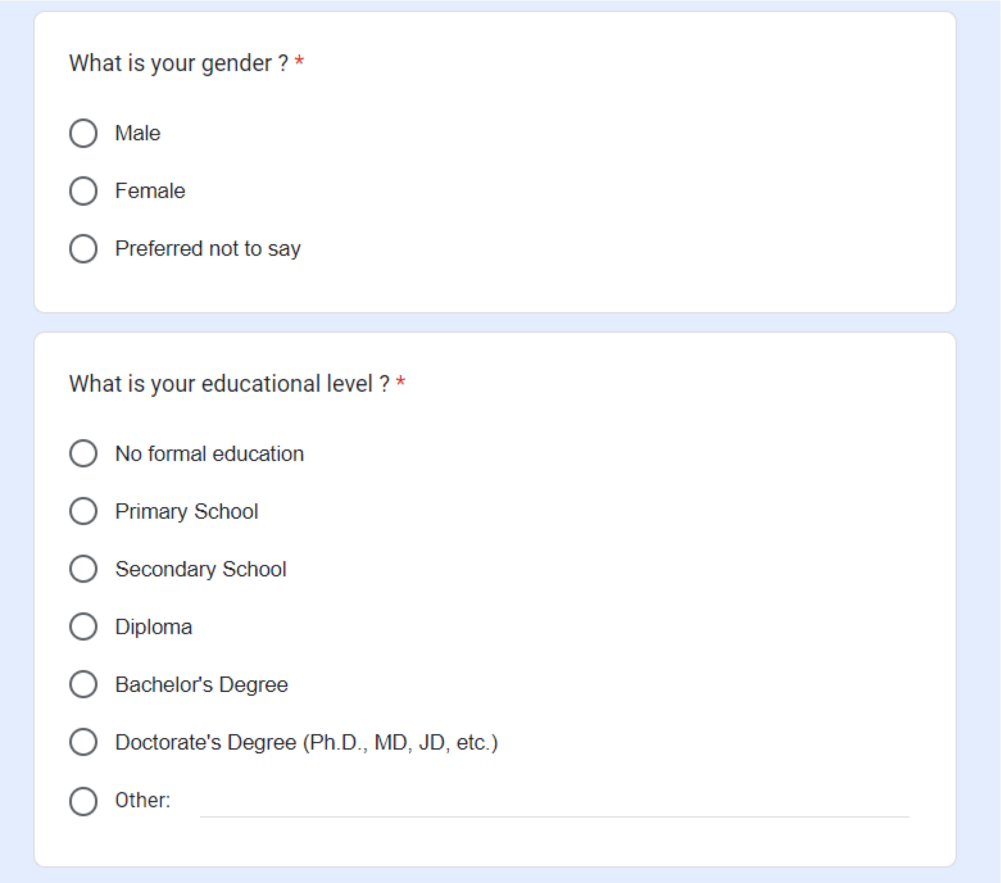


Personal Information
In this section, we kindly request some personal information to better understand our users. Your responses will only be used for analysis purposes to enhance our bill management system.

What is your age range ? *

○ Under 18 years old
○ 18-24 years old
○ 25-34 years old
○ 35-44 years old
○ 45-54 years old
○ 55-64 years old
○ 65 years old and above

Figure A-3: Section one of the survey (cont)



What is your gender ? *

○ Male

○ Female

○ Preferred not to say

What is your educational level ? *

○ No formal education

○ Primary School

○ Secondary School

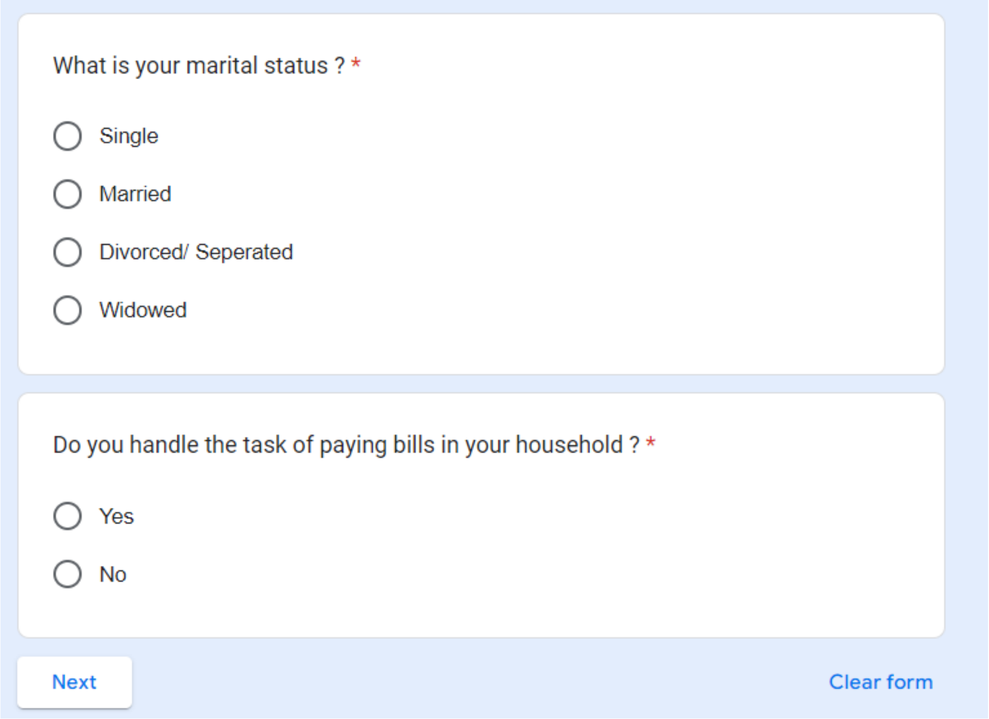
○ Diploma

○ Bachelor's Degree

○ Doctorate's Degree (Ph.D., MD, JD, etc.)

○ Other: ____________________

Figure A-4: Section one of the survey (cont)



What is your marital status ? *

○ Single

○ Married

○ Divorced/ Seperated

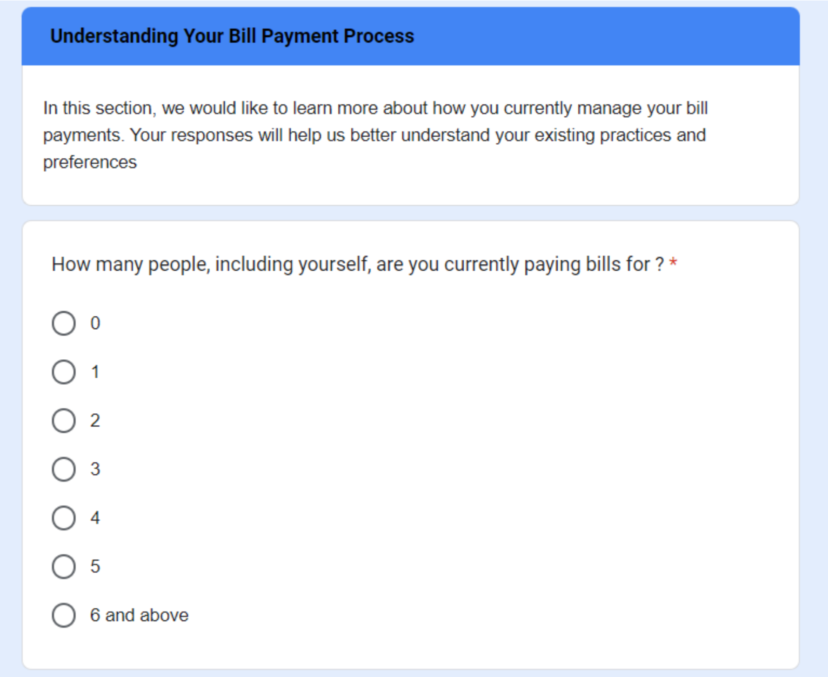
○ Widowed

Do you handle the task of paying bills in your household ? *

○ Yes

○ No

Figure A-5: Section two of the survey



Understanding Your Bill Payment Process

In this section, we would like to learn more about how you currently manage your bill payments. Your responses will help us better understand your existing practices and preferences

How many people, including yourself, are you currently paying bills for ? *

○ 0

○ 1

○ 2

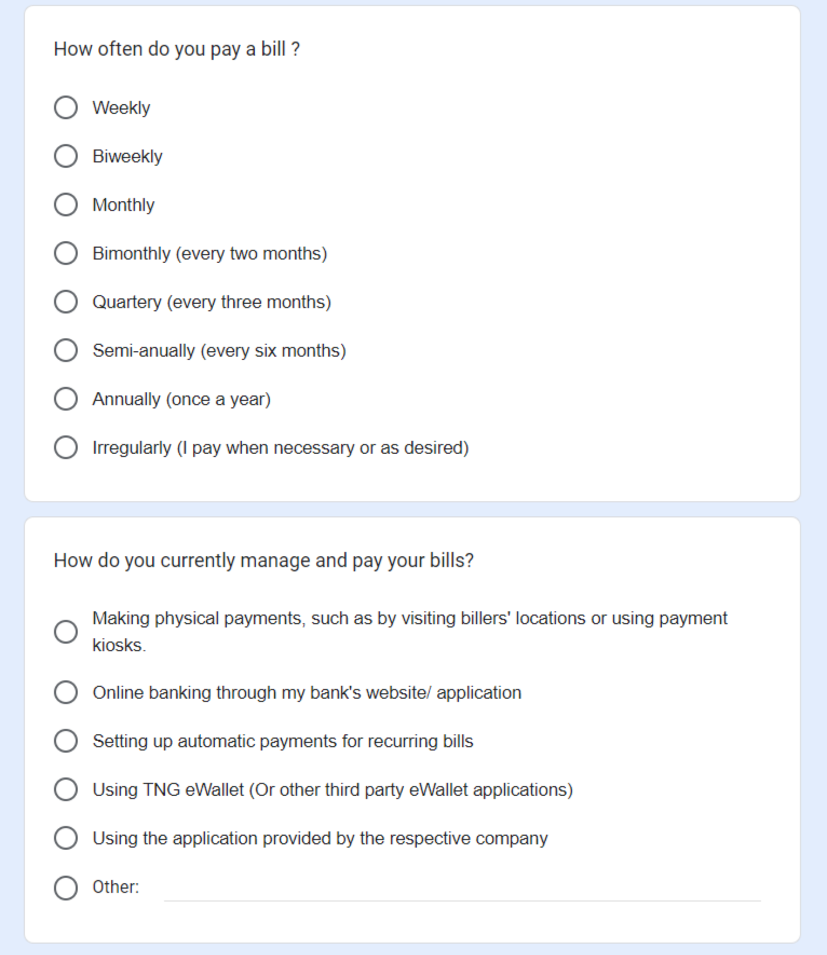
○ 3

○ 4

○ 5

○ 6 and above

Figure A-6: Section two of the survey (cont)



How often do you pay a bill ?

○ Weekly

○ Biweekly

○ Monthly

○ Bimonthly (every two months)

○ Quartery (every three months)

○ Semi-anually (every six months)

○ Annually (once a year)

○ Irregularly (I pay when necessary or as desired)

How do you currently manage and pay your bills?

○ Making physical payments, such as by visiting billers' locations or using payment kiosks.

○ Online banking through my bank's website/ application

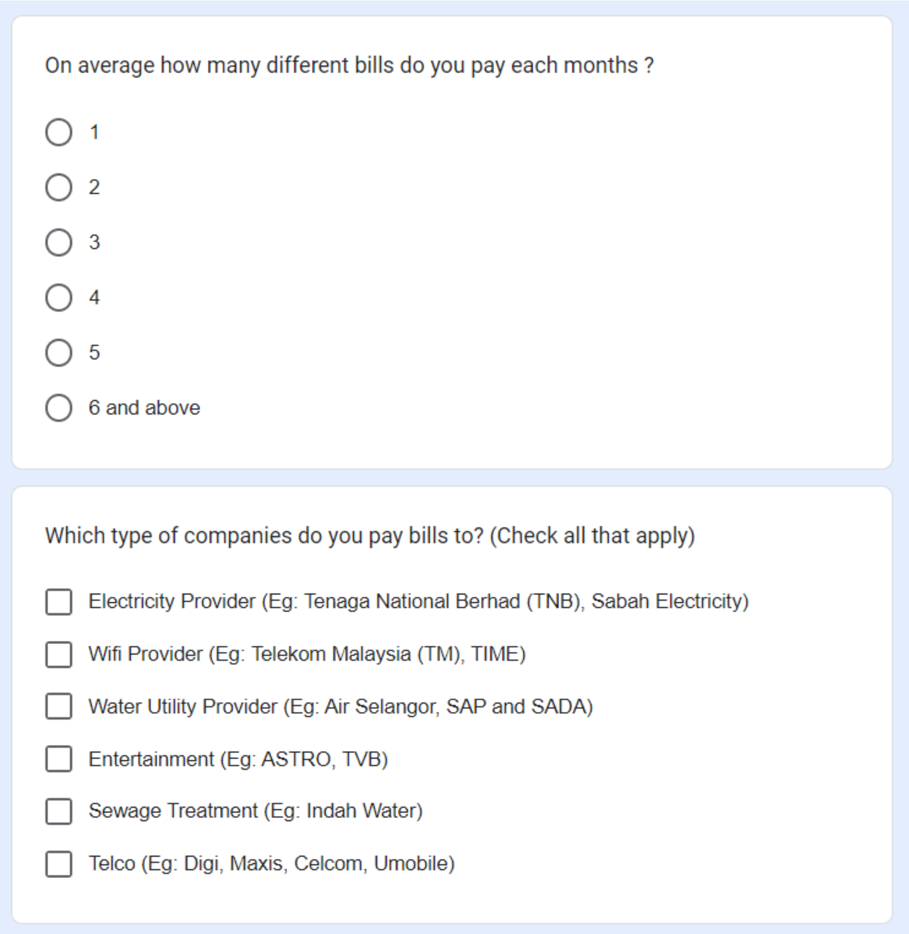
○ Setting up automatic payments for recurring bills

○ Using TNG eWallet (Or other third party eWallet applications)

○ Using the application provided by the respective company

○ Other: __________

Figure A-7: Section two of the survey (cont)



On average how many different bills do you pay each months ?

○ 1

○ 2

○ 3

○ 4

○ 5

○ 6 and above

Which type of companies do you pay bills to? (Check all that apply)

□ Electricity Provider (Eg: Tenaga National Berhad (TNB), Sabah Electricity)

□ Wifi Provider (Eg: Telekom Malaysia (TM), TIME)

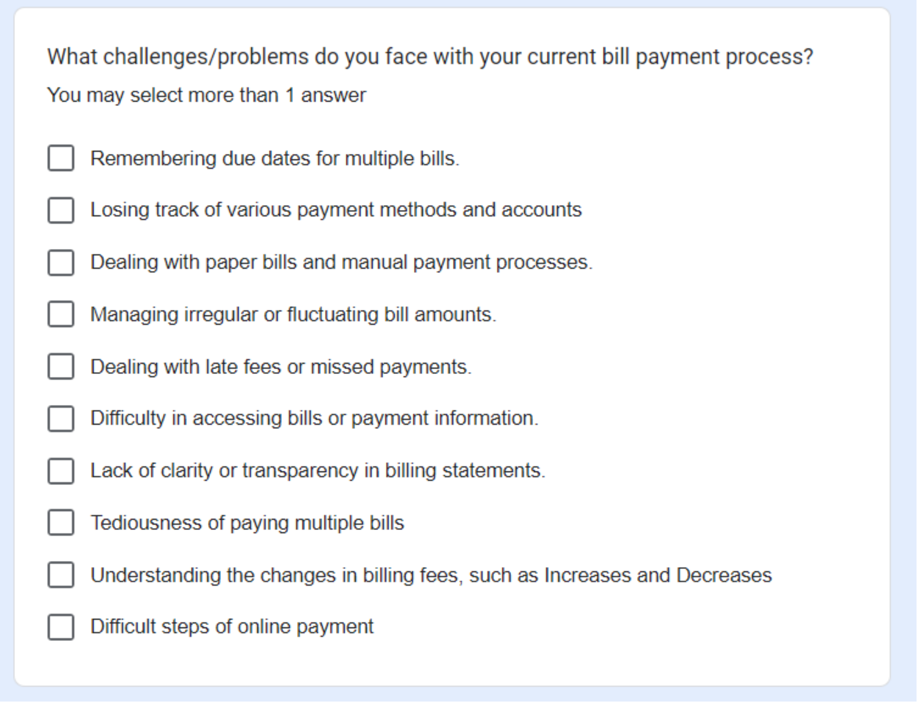
□ Water Utility Provider (Eg: Air Selangor, SAP and SADA)

□ Entertainment (Eg: ASTRO, TVB)

□ Sewage Treatment (Eg: Indah Water)

□ Telco (Eg: Digi, Maxis, Celcom, Umobile)

Figure A-8: Section two of the survey (cont)



What challenges/problems do you face with your current bill payment process?

You may select more than 1 answer

□ Remembering due dates for multiple bills.

□ Losing track of various payment methods and accounts

□ Dealing with paper bills and manual payment processes.

□ Managing irregular or fluctuating bill amounts.

□ Dealing with late fees or missed payments.

□ Difficulty in accessing bills or payment information.

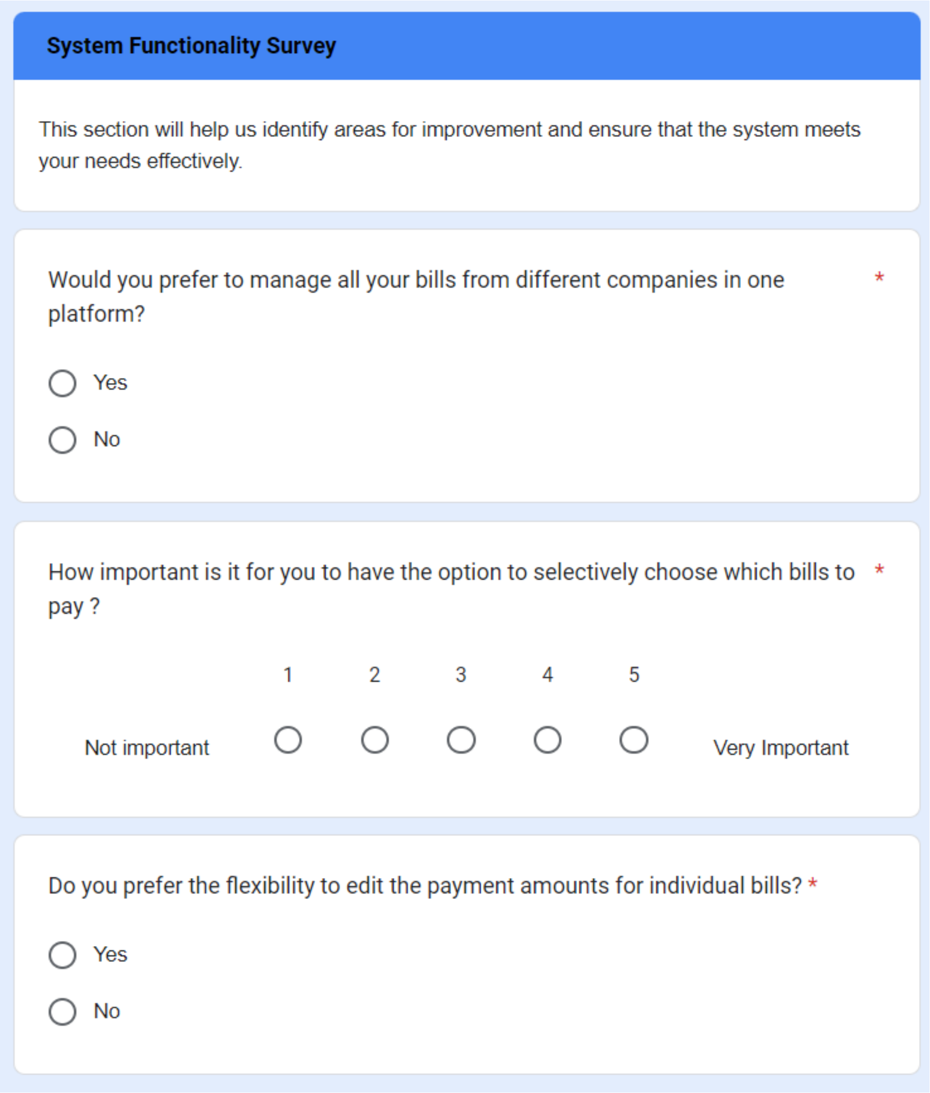
□ Lack of clarity or transparency in billing statements.

□ Tediousness of paying multiple bills

□ Understanding the changes in billing fees, such as Increases and Decreases

□ Difficult steps of online payment

Figure A-9: Section three of the survey



System Functionality Survey

This section will help us identify areas for improvement and ensure that the system meets your needs effectively.

Would you prefer to manage all your bills from different companies in one platform? *

○ Yes

○ No

How important is it for you to have the option to selectively choose which bills to pay ? *

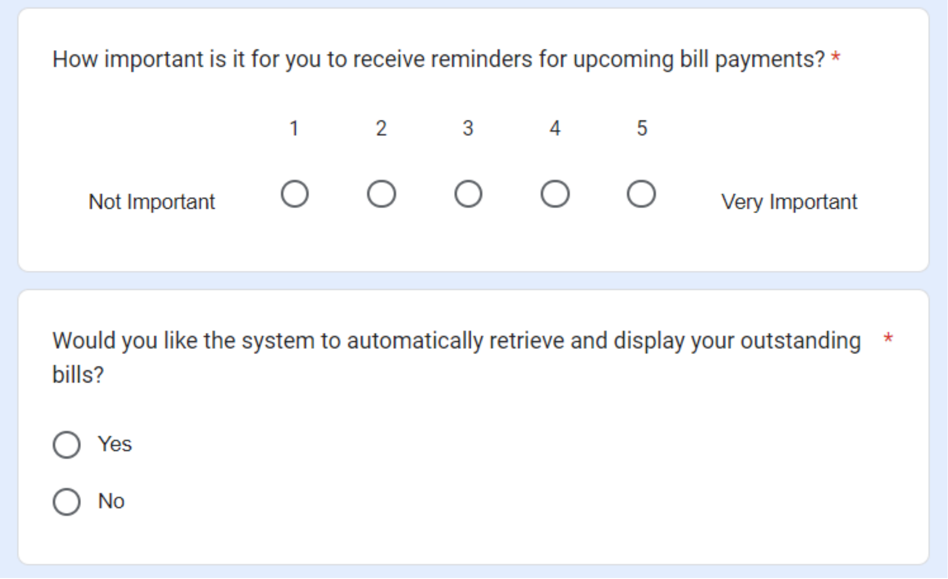
	1	2	3	4	5	
Not important	○	○	○	○	○	Very Important

Do you prefer the flexibility to edit the payment amounts for individual bills? *

○ Yes

○ No

Figure A-10: Section three of the survey (cont)



How important is it for you to receive reminders for upcoming bill payments? *

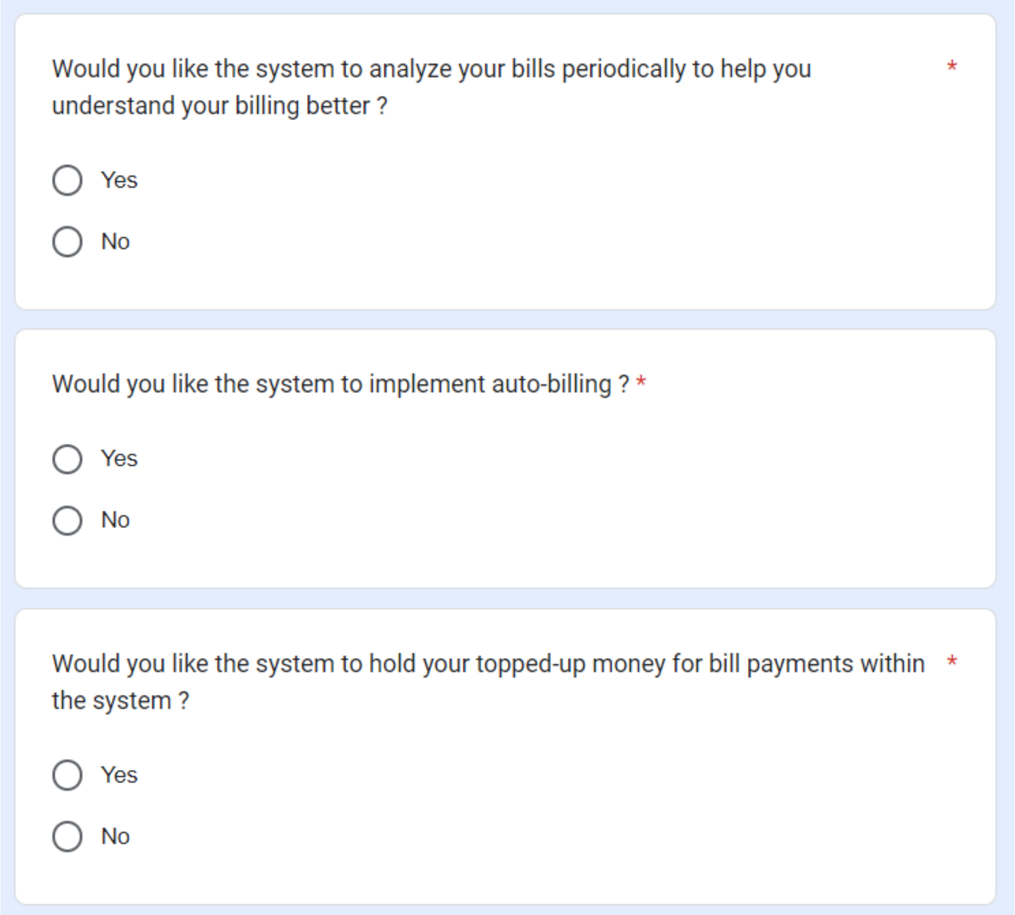
	1	2	3	4	5	
Not Important	○	○	○	○	○	Very Important

Would you like the system to automatically retrieve and display your outstanding bills? *

○ Yes

○ No

Figure A-11: Section three of the survey (cont)



Would you like the system to analyze your bills periodically to help you understand your billing better ? *

○ Yes

○ No

Would you like the system to implement auto-billing ? *

○ Yes

○ No

Would you like the system to hold your topped-up money for bill payments within the system ? *

○ Yes

○ No

Appendix B: User Satisfaction Survey Forms

Table B-1: User Acceptance and Satisfaction Form

User Acceptance and Satisfaction Form (For Bill Payer)	
ID:	**Name:**
Questions	**Rating**
1. It is easy to register an account with this application. (1 – 5, Strongly Disagree – Strongly Agree)	
2. The process of logging in with a phone number/email and password is easy. (1 – 5, Strongly Disagree – Strongly Agree)	
3. The verification code sent for mobile numbers/Email is quick and reliable. (1 – 5, Strongly Disagree – Strongly Agree)	
4. The OTP verification process is smooth and secure. (1 – 5, Strongly Disagree – Strongly Agree)	
5. The password setup process is easy to understand and clearly guided. (1 – 5, Strongly Disagree – Strongly Agree)	
6. It is easy to navigate through different sections of the application. (1 – 5, Strongly Disagree – Strongly Agree)	
7. The application makes it easy to search for and select billing companies. (1 – 5, Strongly Disagree – Strongly Agree)	
8. Registering a bill with an account number and nickname is simple and efficient. (1 – 5, Strongly Disagree – Strongly Agree)	
9. The application provides clear and detailed overview for the registered bills. (1 – 5, Strongly Disagree – Strongly Agree)	
10. The application provides timely notifications about billing dates and due dates. (1 – 5, Strongly Disagree – Strongly Agree)	

11. The application's features help in reducing the time taken to pay bills. (1 – 5, Strongly Disagree – Strongly Agree)	
12. The buttons in the application is conveniently placed and easy to access. (1 – 5, Strongly Disagree – Strongly Agree)	
13. The application effectively prevents accidental resending of OTPs by enforcing a countdown. (1 – 5, Strongly Disagree – Strongly Agree)	
14. The application provides useful filters for categorizing billing companies. (1 – 5, Strongly Disagree – Strongly Agree)	
15. The bill reminder notifications are timely and helpful. (1 – 5, Strongly Disagree – Strongly Agree)	
16. Setting up bill reminders is easy and intuitive. (1 – 5, Strongly Disagree – Strongly Agree)	
17. The reminder notifications are customizable to my preferences. (1 – 5, Strongly Disagree – Strongly Agree)	
18. The application provides clear instructions for setting up auto-billing. (1 – 5, Strongly Disagree – Strongly Agree)	
19. The bill history data is presented in a clear and organized manner which allows easy understanding. (1 – 5, Strongly Disagree – Strongly Agree)	
20. The bill payments made and bills issued analysis feature enhances my overall bill-related financial understanding. (1 – 5, Strongly Disagree – Strongly Agree)	

What did you like the best about the application?

What did you like the least about the application?

If you were asked to enhance this application, what would you add/modify?

Do you have any extra comments or questions?

Appendix C: User Satisfaction Survey Results

User Acceptance and Satisfaction Form (For Bill Payer)	
ID: P1	**Name:** Bill Payer 1
Questions	**Rating**
1. It is easy to register an account with this application. (1 – 5, Strongly Disagree – Strongly Agree)	4
2. The process of logging in with a phone number/email and password is easy. (1 – 5, Strongly Disagree – Strongly Agree)	5
3. The verification code sent for mobile numbers/Email is quick and reliable. (1 – 5, Strongly Disagree – Strongly Agree)	5
4. The OTP verification process is smooth and secure. (1 – 5, Strongly Disagree – Strongly Agree)	5
5. The password setup process is easy to understand and clearly guided. (1 – 5, Strongly Disagree – Strongly Agree)	5
6. It is easy to navigate through different sections of the application. (1 – 5, Strongly Disagree – Strongly Agree)	5
7. The application makes it easy to search for and select billing companies. (1 – 5, Strongly Disagree – Strongly Agree)	5
8. Registering a bill with an account number and nickname is simple and efficient. (1 – 5, Strongly Disagree – Strongly Agree)	5
9. The application provides clear and detailed overview for the registered bills. (1 – 5, Strongly Disagree – Strongly Agree)	4
10. The application provides timely notifications about billing dates and due dates. (1 – 5, Strongly Disagree – Strongly Agree)	4
11. The application’s features help in reducing the time taken to pay bills. (1 – 5, Strongly Disagree – Strongly Agree)	5

12. The buttons in the application is conveniently placed and easy to access. (1 – 5, Strongly Disagree – Strongly Agree)	5
13. The application effectively prevents accidental resending of OTPs by enforcing a countdown. (1 – 5, Strongly Disagree – Strongly Agree)	5
14. The application provides useful filters for categorizing billing companies. (1 – 5, Strongly Disagree – Strongly Agree)	5
15. The bill reminder notifications are timely and helpful. (1 – 5, Strongly Disagree – Strongly Agree)	4
16. Setting up bill reminders is easy and intuitive. (1 – 5, Strongly Disagree – Strongly Agree)	5
17. The reminder notifications are customizable to my preferences. (1 – 5, Strongly Disagree – Strongly Agree)	5
18. The application provides clear instructions for setting up auto-billing. (1 – 5, Strongly Disagree – Strongly Agree)	4
19. The bill history data is presented in a clear and organized manner which allows easy understanding. (1 – 5, Strongly Disagree – Strongly Agree)	5
20. The bill payments made and bills issued analysis feature enhances my overall bill-related financial understanding. (1 – 5, Strongly Disagree – Strongly Agree)	5

What did you like the best about the application?
Automatically analyse the bills and payment made, helps to understand the total expenses on bill.

What did you like the least about the application?
Too many customizations, increases the learning curve.

If you were asked to enhance this application, what would you add/modify?
Provide default reminder and auto-billing settings.

Do you have any extra comments or questions?
Overall, this is a very convenient application.

User Acceptance and Satisfaction Form (For Bill Payer)	
ID: P2	**Name:** Bill Payer 2
Questions	**Rating**
1. It is easy to register an account with this application. (1 – 5, Strongly Disagree – Strongly Agree)	4
2. The process of logging in with a phone number/email and password is easy. (1 – 5, Strongly Disagree – Strongly Agree)	5
3. The verification code sent for mobile numbers/Email is quick and reliable. (1 – 5, Strongly Disagree – Strongly Agree)	5
4. The OTP verification process is smooth and secure. (1 – 5, Strongly Disagree – Strongly Agree)	5
5. The password setup process is easy to understand and clearly guided. (1 – 5, Strongly Disagree – Strongly Agree)	5
6. It is easy to navigate through different sections of the application. (1 – 5, Strongly Disagree – Strongly Agree)	4
7. The application makes it easy to search for and select billing companies. (1 – 5, Strongly Disagree – Strongly Agree)	5
8. Registering a bill with an account number and nickname is simple and efficient. (1 – 5, Strongly Disagree – Strongly Agree)	5
9. The application provides clear and detailed overview for the registered bills. (1 – 5, Strongly Disagree – Strongly Agree)	4
10. The application provides timely notifications about billing dates and due dates. (1 – 5, Strongly Disagree – Strongly Agree)	4
11. The application's features help in reducing the time taken to pay bills. (1 – 5, Strongly Disagree – Strongly Agree)	4

12. The button in the application is conveniently placed and easy to access. (1 – 5, Strongly Disagree – Strongly Agree)	3
13. The application effectively prevents accidental resending of OTPs by enforcing a countdown. (1 – 5, Strongly Disagree – Strongly Agree)	5
14. The application provides useful filters for categorizing billing companies. (1 – 5, Strongly Disagree – Strongly Agree)	5
15. The bill reminder notifications are timely and helpful. (1 – 5, Strongly Disagree – Strongly Agree)	5
16. Setting up bill reminders is easy and intuitive. (1 – 5, Strongly Disagree – Strongly Agree)	5
17. The reminder notifications are customizable to my preferences. (1 – 5, Strongly Disagree – Strongly Agree)	5
18. The application provides clear instructions for setting up auto-billing. (1 – 5, Strongly Disagree – Strongly Agree)	4
19. The bill history data is presented in a clear and organized manner which allows easy understanding. (1 – 5, Strongly Disagree – Strongly Agree)	4
20. The bill payments made and bills issued analysis feature enhances my overall bill-related financial understanding. (1 – 5, Strongly Disagree – Strongly Agree)	5

What did you like the best about the application?
The function of auto-billing can help me in paying repeating bills such as telecommunication bills which are fixed amount every month. Meanwhile the function of billing reminder helps me prevent from paying the bills late and avoid penalties.

What did you like the least about the application?
The overall UI design can be improved. Function-wise, everything is great!

If you were asked to enhance this application, what would you add/modify?
Maybe can include points or coins mechanism, which rewards user points or coins to redeem rewards as they pay bills

Do you have any extra comments or questions?
Improve UI and add more transition or animation.

User Acceptance and Satisfaction Form (For Bill Payer)

ID: P3	**Name:** Bill Payer 3
Questions	**Rating**
1. It is easy to register an account with this application. (1 – 5, Strongly Disagree – Strongly Agree)	5
2. The process of logging in with a phone number/email and password is easy. (1 – 5, Strongly Disagree – Strongly Agree)	4
3. The verification code sent for mobile numbers/Email is quick and reliable. (1 – 5, Strongly Disagree – Strongly Agree)	5
4. The OTP verification process is smooth and secure. (1 – 5, Strongly Disagree – Strongly Agree)	5
5. The password setup process is easy to understand and clearly guided. (1 – 5, Strongly Disagree – Strongly Agree)	5
6. It is easy to navigate through different sections of the application. (1 – 5, Strongly Disagree – Strongly Agree)	5
7. The application makes it easy to search for and select billing companies. (1 – 5, Strongly Disagree – Strongly Agree)	5
8. Registering a bill with an account number and nickname is simple and efficient. (1 – 5, Strongly Disagree – Strongly Agree)	4
9. The application provides clear and detailed overview for the registered bills. (1 – 5, Strongly Disagree – Strongly Agree)	5
10. The application provides timely notifications about billing dates and due dates. (1 – 5, Strongly Disagree – Strongly Agree)	4
11. The application's features help in reducing the time taken to pay bills. (1 – 5, Strongly Disagree – Strongly Agree)	5

12. The buttons in the application is conveniently placed and easy to access. (1 – 5, Strongly Disagree – Strongly Agree)	3
13. The application effectively prevents accidental resending of OTPs by enforcing a countdown. (1 – 5, Strongly Disagree – Strongly Agree)	5
14. The application provides useful filters for categorizing billing companies. (1 – 5, Strongly Disagree – Strongly Agree)	5
15. The bill reminder notifications are timely and helpful. (1 – 5, Strongly Disagree – Strongly Agree)	5
16. Setting up bill reminders is easy and intuitive. (1 – 5, Strongly Disagree – Strongly Agree)	3
17. The reminder notifications are customizable to my preferences. (1 – 5, Strongly Disagree – Strongly Agree)	5
18. The application provides clear instructions for setting up auto-billing. (1 – 5, Strongly Disagree – Strongly Agree)	4
19. The bill history data is presented in a clear and organized manner which allows easy understanding. (1 – 5, Strongly Disagree – Strongly Agree)	5
20. The bill payments made and bills issued analysis feature enhances my overall bill-related financial understanding. (1 – 5, Strongly Disagree – Strongly Agree)	5

What did you like the best about the application?
I am paying multiple bills every month, I really like the feature of paying multiple bills in a one transaction, it greatly saved up my time.

What did you like the least about the application?
No identity verification during the NRIC registration process.

If you were asked to enhance this application, what would you add/modify?
Synchronize with billing company to add a
feature that allow bill payers to view the usage of the bills.

Do you have any extra comments or questions?
No

User Acceptance and Satisfaction Form (For Bill Payer)	
ID: P4	**Name:** Bill Payer 4
Questions	**Rating**
1. It is easy to register an account with this application. (1 – 5, Strongly Disagree – Strongly Agree)	4
2. The process of logging in with a phone number/email and password is easy. (1 – 5, Strongly Disagree – Strongly Agree)	5
3. The verification code sent for mobile numbers/Email is quick and reliable. (1 – 5, Strongly Disagree – Strongly Agree)	5
4. The OTP verification process is smooth and secure. (1 – 5, Strongly Disagree – Strongly Agree)	5
5. The password setup process is easy to understand and clearly guided. (1 – 5, Strongly Disagree – Strongly Agree)	5
6. It is easy to navigate through different sections of the application. (1 – 5, Strongly Disagree – Strongly Agree)	3
7. The application makes it easy to search for and select billing companies. (1 – 5, Strongly Disagree – Strongly Agree)	5
8. Registering a bill with an account number and nickname is simple and efficient. (1 – 5, Strongly Disagree – Strongly Agree)	5
9. The application provides clear and detailed overview for the registered bills. (1 – 5, Strongly Disagree – Strongly Agree)	2
10. The application provides timely notifications about billing dates and due dates. (1 – 5, Strongly Disagree – Strongly Agree)	3
11. The application's features help in reducing the time taken to pay bills. (1 – 5, Strongly Disagree – Strongly Agree)	5

12. The buttons in the application is conveniently placed and easy to access. (1 – 5, Strongly Disagree – Strongly Agree)	4
13. The application effectively prevents accidental resending of OTPs by enforcing a countdown. (1 – 5, Strongly Disagree – Strongly Agree)	5
14. The application provides useful filters for categorizing billing companies. (1 – 5, Strongly Disagree – Strongly Agree)	5
15. The bill reminder notifications are timely and helpful. (1 – 5, Strongly Disagree – Strongly Agree)	4
16. Setting up bill reminders is easy and intuitive. (1 – 5, Strongly Disagree – Strongly Agree)	4
17. The reminder notifications are customizable to my preferences. (1 – 5, Strongly Disagree – Strongly Agree)	5
18. The application provides clear instructions for setting up auto-billing. (1 – 5, Strongly Disagree – Strongly Agree)	5
19. The bill history data is presented in a clear and organized manner which allows easy understanding. (1 – 5, Strongly Disagree – Strongly Agree)	5
20. The bill payments made and bills issued analysis feature enhances my overall bill-related financial understanding. (1 – 5, Strongly Disagree – Strongly Agree)	5

What did you like the best about the application?
Able to pay multiple bills at once.

What did you like the least about the application?
Lack of auto-reload functionality, this will require me to reload manually every month.

If you were asked to enhance this application, what would you add/modify?
Add auto reload when remaining balance is lower than a specific value.

Do you have any extra comments or questions?
Convenient application for users with many bills.

User Acceptance and Satisfaction Form (For Bill Payer)	
ID: P5	**Name:** Bill Payer 5
Questions	**Rating**
1. It is easy to register an account with this application. (1 – 5, Strongly Disagree – Strongly Agree)	5
2. The process of logging in with a phone number/email and password is easy. (1 – 5, Strongly Disagree – Strongly Agree)	5
3. The verification code sent for mobile numbers/Email is quick and reliable. (1 – 5, Strongly Disagree – Strongly Agree)	5
4. The OTP verification process is smooth and secure. (1 – 5, Strongly Disagree – Strongly Agree)	5
5. The password setup process is easy to understand and clearly guided. (1 – 5, Strongly Disagree – Strongly Agree)	5
6. It is easy to navigate through different sections of the application. (1 – 5, Strongly Disagree – Strongly Agree)	5
7. The application makes it easy to search for and select billing companies. (1 – 5, Strongly Disagree – Strongly Agree)	4
8. Registering a bill with an account number and nickname is simple and efficient. (1 – 5, Strongly Disagree – Strongly Agree)	5
9. The application provides clear and detailed overview for the registered bills. (1 – 5, Strongly Disagree – Strongly Agree)	3
10. The application provides timely notifications about billing dates and due dates. (1 – 5, Strongly Disagree – Strongly Agree)	4
11. The application's features help in reducing the time taken to pay bills. (1 – 5, Strongly Disagree – Strongly Agree)	5

12. The buttons in the application is conveniently placed and easy to access. (1 – 5, Strongly Disagree – Strongly Agree)	5
13. The application effectively prevents accidental resending of OTPs by enforcing a countdown. (1 – 5, Strongly Disagree – Strongly Agree)	4
14. The application provides useful filters for categorizing billing companies. (1 – 5, Strongly Disagree – Strongly Agree)	5
15. The bill reminder notifications are timely and helpful. (1 – 5, Strongly Disagree – Strongly Agree)	5
16. Setting up bill reminders is easy and intuitive. (1 – 5, Strongly Disagree – Strongly Agree)	5
17. The reminder notifications are customizable to my preferences. (1 – 5, Strongly Disagree – Strongly Agree)	5
18. The application provides clear instructions for setting up auto-billing. (1 – 5, Strongly Disagree – Strongly Agree)	4
19. The bill history data is presented in a clear and organized manner which allows easy understanding. (1 – 5, Strongly Disagree – Strongly Agree)	5
20. The bill payments made and bills issued analysis feature enhances my overall bill-related financial understanding. (1 – 5, Strongly Disagree – Strongly Agree)	5

What did you like the best about the application?
Able to see the due dates and outstanding amount of each bill at one page, enable bill payer to estimate the amount have to be spent on bill payment.

What did you like the least about the application?
Unable to set auto-billing with credit cards.

If you were asked to enhance this application, what would you add/modify?
Enable auto-billing with other payment methods.

Do you have any extra comments or questions?
Good, try to explore and add more E-wallet likes features.

Appendix D: User Acceptance Testing Form

User Acceptance Test (UAT)						
Tester’s Name				Testing Start Date/Time		
				Testing End Date/Time		
Test Case ID	Module	Test Case Title	Test Steps	Expected Result	Status	Comments
UAT001	Register Account	Register an BillHub account	1. User clicks “Sign Up Now” button. 2. User enters his phone number. 3. User fills in the verification code received via SMS. 4. User enters personal information and email address. 5. User fills in the verification code received via email. 6. User make sure his password filled in is strong enough and meet the requirements. 7. User submit the create account request.	User will successfully create an account that he can use for login		
UAT002	Login Account	Login into BillHub account created	1. User clicks “Login” button. 2. User enters email/phone number and password. 3. User clicks “Login” button.	User will be navigated to “Home” page with welcome message		
UAT003	Register Bill	Register a bill from a billing company	1. User clicks “Add Bill” button at the middle of home screen. 2. User selects a billing company from the list. 3. User enters the account number of the bill. 4. User enters a name for the bill. 5. User clicks the “Register” button. 6. User repeat steps 1-5 to register 2-3 different bills.	The registered bill will be visible at “Home” page and “My Bill” page		
UAT004	Delete Bill	Remove a registered bill from the account	1. User selects a bill in home page or “My Bill” page. 2. User clicks the rubbish bin icon on the top right corner of the “Bill Detail” screen. 3. User confirms the bill deletion alert.	The registered bill will be removed from the account.		

UAT005	Reload Credit	Reload BillHub credits that can be used for bill payments and auto-billing	1. User clicks the “reload” button at the home page. 2. User selects an amount to reload or enter it via the input box. 3. User enters banking information inside payment gateway. 4. User approves the transaction.	The credits amount shown in “Home” page will be increased with exact amount that the user reloaded.		
UAT006	Pay Bill	Pay one of the registered bills or multiple bills in one transaction	1. User clicks “Pay” button at the home page. 2. User ticks the bills that he wants to pay. 3. User edits the amount that he wants to pay if he doesn’t want to pay the amount of outstanding value. 4. User clicks “Pay” button. 5. User confirms the total payment amounts and the number of selected bills for payment. 6. User selects a payment method. 7. User enter the bank information into payment gateway if he selects online FPX payment or card payment.	A payment receipt will be shown, showing the transaction ID, payment date, payment time, payment method, bills paid, and their payment amounts respectively.		
UAT007	View Bill and History	View the details of a bill including billing and payment history	1. User clicks any of the registered bills in home page or “My Bill” page. 2. User clicks on the “history” button.	The bill details such as account number, bill’s owner, phone number, billing history and payment history will be shown.		
UAT008	View Bill Analysis	View the analysis of bills issued and payment made	1. User selects the “Bill Analysis” at the middle of home page. 2. User selects to view “payment made analysis” or “bills issued analysis” 3. User selects “Month” or “Year” at the bottom of the Analysis Page. 4. If user selects “Month”, he should choose a month to view the analysis of the selected month.	The bills issued and payment made analysis are shown in the form of Pie Chart, Table Form (Month) and Line Chart (Year)		

UAT009	Set Bill Reminder	Turning on/off bill reminder and customize the reminder	1. User selects “Bill Reminder” from the drawer menu. 2. User selects a bill from the list of registered bills or select “All bills”. 3. User clicks the switch to set the reminder to on or off. 4. User selects the preferred reminder methods. 5. User selects the preferred reminder timing. 6. User clicks the “Save” button.	On the selected reminder timing, the reminder message will be sent to user via SMS or notification, or email based on their customization.		
UAT010	Set Auto-Billing	Setting up a bill to pay automatically	1. User clicks Auto-Billing located at the Home Page. 2. User selects a bill from the list of registered bills. 3. User selects a payment date. 4. User selects a payment amount. 5. User set the maximum transaction allowed for outstanding amount or set the fixed amount to be paid.	The bills will be paid during the selected payment date. The receipt will be generated and can be viewed from payment history.		
UAT011	View Notification	Check the messages or reminders sent to the user	1. User clicks the bell shape icon located at the top right corner of the home page. 2. User selects one of the notifications from the notifications list.	The notification message and details will be shown in a pop-up modal.		
UAT012	Delete Notification	Remove the notification that has been checked	1. User clicks the bell shape icon located at the top right corner of the home page. 2. User selects one of the notifications from the notifications list. 3. User clicks on the “DELETE” button 4. User clicks on the “DELETE” button again on the delete confirmation alert.	The notification deleted will be removed from the notification page.		
UAT013	Edit Bill Name	Add or change the name used the represent the registered bill	1. User selects a registered bill from the home page or “My Bills” page. 2. Inside the “Bill Detail” page, clicks the edit button	The bill in the home page will now appear with the updated name instead of its previous name.		

			located at the right-hand side of the name. 3. User enters a new name for the registered bill. 4. User clicks the "Save" button located at the bottom of the page.			
UAT014	Log Out	Quit from using the logged-in account	1. User clicks the menu button located at the top-left corner of home page. 2. User clicks the "Log Out" button from the drawer menu shown.	The user's information will be removed from the async storage, login screen will be shown.		

Appendix E: Results of User Acceptance Test

User Acceptance Test (UAT)						
Tester’s Name	Bill Payer 1			Testing Start Date/Time	09/09/2024 2pm	
				Testing End Date/Time	09/09/2024 2:27pm	
Test Case ID	Module	Test Case Title	Test Steps	Expected Result	Status	Comments
UAT001	Register Account	Register an BillHub account	1. User clicks “Sign Up Now” button. 2. User enters his phone number. 3. User fills in the verification code received via SMS. 4. User enters personal information and email address. 5. User fills in the verification code received via email. 6. User make sure his password filled in is strong enough and meet the requirements. 7. User submit the create account request.	User will successfully create an account that he can use for login	Pass	Instead of main page, can redirect user to Login page
UAT002	Login Account	Login into BillHub account created	1. User clicks “Login” button. 2. User enters email/phone number and password. 3. User clicks “Login” button.	User will be navigated to “Home” page with welcome message	Pass	Clearly shown the reason of unsuccessful attempt
UAT003	Register Bill	Register a bill from a billing company	1. User clicks “Add Bill” button at the middle of home screen. 2. User selects a billing company from the list. 3. User enters the account number of the bill. 4. User enters a name for the bill. 5. User clicks the “Register” button. 6. User repeat steps 1-5 to register 2-3 different bills.	The registered bill will be visible at “Home” page and “My Bill” page	Pass	-
UAT004	Delete Bill	Remove a registered bill from the account	1. User selects a bill in home page or “My Bill” page. 2. User clicks the rubbish bin icon on the top right corner of the “Bill Detail” screen.	The registered bill will be removed from the account.	Pass	-

			3. User confirms the bill deletion alert.			
UAT005	Reload Credit	Reload BillHub credits that can be used for bill payments and auto-billing	1. User clicks the “reload” button at the home page. 2. User selects an amount to reload or enter it via the input box. 3. User enters banking information inside payment gateway. 4. User approves the transaction.	The credits amount shown in “Home” page will be increased with exact amount that the user reloaded.	Pass	-
UAT006	Pay Bill	Pay one of the registered bills or multiple bills in one transaction	1. User clicks “Pay” button at the home page. 2. User ticks the bills that he wants to pay. 3. User edits the amount that he wants to pay if he doesn’t want to pay the amount of outstanding value. 4. User clicks “Pay” button. 5. User confirms the total payment amounts and the number of selected bills for payment. 6. User selects a payment method. 7. User enter the bank information into payment gateway if he selects online FPX payment or card payment.	A payment receipt will be shown, showing the transaction ID, payment date, payment time, payment method, bills paid, and their payment amounts respectively.	Pass	Suggest adding a button that select all bills with outstanding amount
UAT007	View Bill and History	View the details of a bill including billing and payment history	1. User clicks any of the registered bills in home page or “My Bill” page. 2. User clicks on the “history” button.	The bill details such as account number, bill’s owner, phone number, billing history and payment history will be shown.	Pass	-
UAT008	View Bill Analysis	View the analysis of bills issued and payment made	1. User selects the “Bill Analysis” at the middle of home page. 2. User selects to view “payment made analysis” or “bills issued analysis” 3. User selects “Month” or “Year” at the bottom of the Analysis Page. 4. If user selects “Month”, he should choose a month to	The bills issued and payment made analysis are shown in the form of Pie Chart, Table Form (Month) and Line Chart (Year)	Pass	Suggest to keep the colour of Pie Chart when switch between month and year.

			view the analysis of the selected month.			
UAT009	Set Bill Reminder	Turning on/off bill reminder and customize the reminder	1. User selects "Bill Reminder" from the drawer menu. 2. User selects a bill from the list of registered bills or select "All bills". 3. User clicks the switch to set the reminder to on or off. 4. User selects the preferred reminder methods. 5. User selects the preferred reminder timing. 6. User clicks the "Save" button.	On the selected reminder timing, the reminder message will be sent to user via SMS or notification, or email based on their customization.	Pass	-
UAT010	Set Auto-Billing	Setting up a bill to pay automatically	1. User clicks Auto-Billing located at the Home Page. 2. User selects a bill from the list of registered bills. 3. User selects a payment date. 4. User selects a payment amount. 5. User set the maximum transaction allowed for outstanding amount or set the fixed amount to be paid.	The bills will be paid during the selected payment date. The receipt will be generated and can be viewed from payment history.	Pass	-
UAT011	View Notification	Check the messages or reminders sent to the user	1. User clicks the bell shape icon located at the top right corner of the home page. 2. User selects one of the notifications from the notifications list.	The notification message and details will be shown in a pop-up modal.	Pass	-
UAT012	Delete Notification	Remove the notification that has been checked	1. User clicks the bell shape icon located at the top right corner of the home page. 2. User selects one of the notifications from the notifications list. 3. User clicks on the "DELETE" button 4. User clicks on the "DELETE" button again on the delete confirmation alert.	The notification deleted will be removed from the notification page.	Pass	-
UAT013	Edit Bill Name	Add or change the	1. User selects a registered bill from	The bill in the home page will	Pass	-

		name used the represent the registered bill	the home page or “My Bills” page. 2. Inside the “Bill Detail” page, clicks the edit button located at the right-hand side of the name. 3. User enters a new name for the registered bill. 4. User clicks the “Save” button located at the bottom of the page.	now appear with the updated name instead of its previous name.		
UAT014	Log Out	Quit from using the logged-in account	1. User clicks the menu button located at the top-left corner of home page. 2. User clicks the “Log Out” button from the drawer menu shown.	The user’s information will be removed from the async storage, login screen will be shown.	Fail	Trigger an error: Unable to access “._id” of undefined.

User Acceptance Test (UAT)						
Tester’s Name	Bill Payer 2			Testing Start Date/Time	09/09/2024 4:15pm	
				Testing End Date/Time	09/09/2024 4:30pm	
Test Case ID	Module	Test Case Title	Test Steps	Expected Result	Status	Comments
UAT001	Register Account	Register an BillHub account	1. User clicks “Sign Up Now” button. 2. User enters his phone number. 3. User fills in the verification code received via SMS. 4. User enters personal information and email address. 5. User fills in the verification code received via email. 6. User make sure his password filled in is strong enough and meet the requirements. 7. User submit the create account request.	User will successfully create an account that he can use for login	Pass	Do not have options such as “Resend” for user who doesn’t receive the verification code. No format provided for NRIC, is - needed ?
UAT002	Login Account	Login into BillHub account created	1. User clicks “Login” button. 2. User enters email/phone number and password. 3. User clicks “Login” button.	User will be navigated to “Home” page with welcome message	Pass	Should use different welcome message for first time login
UAT003	Register Bill	Register a bill from a billing company	1. User clicks “Add Bill” button at the middle of home screen.	The registered bill will be visible at “Home” page	Pass	Can guide user where to find their account number

			2. User selects a billing company from the list. 3. User enters the account number of the bill. 4. User enters a name for the bill. 5. User clicks the “Register” button. 6. User repeat steps 1-5 to register 2-3 different bills.	and “My Bill” page		
UAT004	Delete Bill	Remove a registered bill from the account	1. User selects a bill in home page or “My Bill” page. 2. User clicks the rubbish bin icon on the top right corner of the “Bill Detail” screen. 3. User confirms the bill deletion alert.	The registered bill will be removed from the account.	Pass	-
UAT005	Reload Credit	Reload BillHub credits that can be used for bill payments and auto-billing	1. User clicks the “reload” button at the home page. 2. User selects an amount to reload or enter it via the input box. 3. User enters banking information inside payment gateway. 4. User approves the transaction.	The credits amount shown in “Home” page will be increased with exact amount that the user reloaded.	Pass	-
UAT006	Pay Bill	Pay one of the registered bills or multiple bills in one transaction	1. User clicks “Pay” button at the home page. 2. User ticks the bills that he wants to pay. 3. User edits the amount that he wants to pay if he doesn’t want to pay the amount of outstanding value. 4. User clicks “Pay” button. 5. User confirms the total payment amounts and the number of selected bills for payment. 6. User selects a payment method. 7. User enter the bank information into payment gateway if he selects online FPX payment or card payment.	A payment receipt will be shown, showing the transaction ID, payment date, payment time, payment method, bills paid, and their payment amounts respectively.	Pass	Do not need to show the registered bills which are pending

UAT007	View Bill and History	View the details of a bill including billing and payment history	1. User clicks any of the registered bills in home page or “My Bill” page. 2. User clicks on the “history” button.	The bill details such as account number, bill’s owner, phone number, billing history and payment history will be shown.	Pass	Can add status of payment such as “Pending”, “Progressing” or “Completed”
UAT008	View Bill Analysis	View the analysis of bills issued and payment made	1. User selects the “Bill Analysis” at the middle of home page. 2. User selects to view “payment made analysis” or “bills issued analysis” 3. User selects “Month” or “Year” at the bottom of the Analysis Page. 4. If user selects “Month”, he should choose a month to view the analysis of the selected month.	The bills issued and payment made analysis are shown in the form of Pie Chart, Table Form (Month) and Line Chart (Year)	Pass	Can combine the percentage in table instead of legend
UAT009	Set Bill Reminder	Turning on/off bill reminder and customize the reminder	1. User selects “Bill Reminder” from the drawer menu. 2. User selects a bill from the list of registered bills or select “All bills”. 3. User clicks the switch to set the reminder to on or off. 4. User selects the preferred reminder methods. 5. User selects the preferred reminder timing. 6. User clicks the “Save” button.	On the selected reminder timing, the reminder message will be sent to user via SMS or notification, or email based on their customization.	Pass	-
UAT010	Set Auto-Billing	Setting up a bill to pay automatically	1. User clicks Auto-Billing located at the Home Page. 2. User selects a bill from the list of registered bills. 3. User selects a payment date. 4. User selects a payment amount. 5. User set the maximum transaction allowed for outstanding amount or set the fixed amount to be paid.	The bills will be paid during the selected payment date. The receipt will be generated and can be viewed from payment history.	Pass	-
UAT011	View Notification	Check the messages or reminders sent to the user	1. User clicks the bell shape icon located at the top right corner of the home page.	The notification message and details will be shown in a pop-up modal.	Pass	Remove redundant notifications so user can pay attention

			2. User selects one of the notifications from the notifications list.			on important one
UAT012	Delete Notification	Remove the notification that has been checked	1. User clicks the bell shape icon located at the top right corner of the home page. 2. User selects one of the notifications from the notifications list. 3. User clicks on the "DELETE" button 4. User clicks on the "DELETE" button again on the delete confirmation alert.	The notification deleted will be removed from the notification page.	Pass	-
UAT013	Edit Bill Name	Add or change the name used the represent the registered bill	1. User selects a registered bill from the home page or "My Bills" page. 2. Inside the "Bill Detail" page, clicks the edit button located at the right-hand side of the name. 3. User enters a new name for the registered bill. 4. User clicks the "Save" button located at the bottom of the page.	The bill in the home page will now appear with the updated name instead of its previous name.	Pass	-
UAT014	Log Out	Quit from using the logged-in account	1. User clicks the menu button located at the top-left corner of home page. 2. User clicks the "Log Out" button from the drawer menu shown.	The user's information will be removed from the async storage, login screen will be shown.	Pass	-

User Acceptance Test (UAT)						
Tester's Name	Bill Payer 3			Testing Start Date/Time	10/09/2024 10:01am	
				Testing End Date/Time	10/09/2024 10:15am	
Test Case ID	Module	Test Case Title	Test Steps	Expected Result	Status	Comments
UAT001	Register Account	Register an BillHub account	1. User clicks "Sign Up Now" button. 2. User enters his phone number. 3. User fills in the OTP received via SMS. 4. User enters email address.	User will successfully create an account that he can use for login	Pass	-

			5. User fills in the OTP received via email. 6. User fills in his personal information. 7. User make sure his password filled in is strong enough and meet the requirements. 8. User submit the create account request.			
UAT002	Login Account	Login into BillHub account created	1. User clicks “Login” button. 2. User enters email and password. 3. User clicks “Login” button.	User will be navigated to “Home” page with welcome message	Pass	-
UAT003	Register Bill	Register a bill from a billing company	1. User clicks “Add Bill” button at the middle of home screen. 2. User selects a billing company from the list. 3. User enters the account number of the bill. 4. User enters a name for the bill. 5. User clicks the “Register” button. 6. User repeat steps 1-5 to register 2-3 different bills.	The registered bill will be visible at “Home” page and “My Bill” page	Pass	-
UAT004	Delete Bill	Remove a registered bill from the account	1. User selects a bill in home page or “My Bill” page. 2. User clicks the rubbish bin icon on the top right corner of the “Bill Detail” screen. 3. User confirms the bill deletion alert.	The registered bill will be removed from the account.	Pass	-
UAT005	Reload Credit	Reload BillHub credits that can be used for bill payments and auto-billing	1. User clicks the “reload” button at the home page. 2. User selects an amount to reload or enter it via the input box. 3. User enters banking information inside payment gateway. 4. User approves the transaction.	The credits amount shown in “Home” page will be increased with exact amount that the user reloaded.	Pass	-
UAT006	Pay Bill	Pay one of the registered bills or multiple bills	1. User clicks “Pay” button at the home page.	A payment receipt will be shown, showing the	Fail	One of the checkboxes of the bill was unable

		in one transaction	2. User ticks the bills that he wants to pay. 3. User edits the amount that he wants to pay if he doesn't want to pay the amount of outstanding value. 4. User clicks "Pay" button. 5. User confirms the total payment amounts and the number of selected bills for payment. 6. User selects a payment method. 7. User enter the bank information into payment gateway if he selects online FPX payment or card payment.	transaction ID, payment date, payment time, payment method, bills paid, and their payment amounts respectively.		to be selected
UAT007	View Bill and History	View the details of a bill including billing and payment history	1. User clicks any of the registered bills in home page or "My Bill" page. 2. User clicks on the "history" button.	The bill details such as account number, bill's owner, phone number, billing history and payment history will be shown.	Pass	-
UAT008	View Bill Analysis	View the analysis of bills issued and payment made	1. User selects the "Bill Analysis" at the middle of home page. 2. User selects to view "payment made analysis" or "bills issued analysis" 3. User selects "Month" or "Year" at the bottom of the Analysis Page. 4. If user selects "Month", he should choose a month to view the analysis of the selected month.	The bills issued and payment made analysis are shown in the form of Pie Chart, Table Form (Month) and Line Chart (Year)	Pass	-
UAT009	Set Bill Reminder	Turning on/off bill reminder and customize the reminder	1. User selects "Bill Reminder" from the drawer menu. 2. User selects a bill from the list of registered bills or select "All bills". 3. User clicks the switch to set the reminder to on or off. 4. User selects the preferred reminder methods. 5. User selects the preferred reminder timing. 6. User clicks the "Save" button.	On the selected reminder timing, the reminder message will be sent to user via SMS or notification, or email based on their customization.	Pass	-

UAT010	Set Auto-Billing	Setting up a bill to pay automatically	1. User clicks Auto-Billing located at the Home Page. 2. User selects a bill from the list of registered bills. 3. User selects a payment date. 4. User selects a payment amount. 5. User set the maximum transaction allowed for outstanding amount or set the fixed amount to be paid.	The bills will be paid during the selected payment date. The receipt will be generated and can be viewed from payment history.	Pass	-
UAT011	View Notification	Check the messages or reminders sent to the user	1. User clicks the bell shape icon located at the top right corner of the home page. 2. User selects one of the notifications from the notifications list.	The notification message and details will be shown in a pop-up modal.	Pass	-
UAT012	Delete Notification	Remove the notification that has been checked	1. User clicks the bell shape icon located at the top right corner of the home page. 2. User selects one of the notifications from the notifications list. 3. User clicks on the "DELETE" button 4. User clicks on the "DELETE" button again on the delete confirmation alert.	The notification deleted will be removed from the notification page.	Pass	-
UAT013	Edit Bill Name	Add or change the name used the represent the registered bill	1. User selects a registered bill from the home page or "My Bills" page. 2. Inside the "Bill Detail" page, clicks the edit button located at the right-hand side of the name.	The bill in the home page will now appear with the updated name instead of its previous name.	Pass	-

			3. User enters a new name for the registered bill. 4. User clicks the “Save” button located at the bottom of the page.			
UAT014	Log Out	Quit from using the logged-in account	1. User clicks the menu button located at the top-left corner of home page. 2. User clicks the “Log Out” button from the drawer menu shown.	The user’s information will be removed from the async storage, login screen will be shown.	Fail	Trigger an error: Unable to access “._id” of undefined.

User Acceptance Test (UAT)						
Tester’s Name	Bill Payer 4			Testing Start Date/Time	10/09/2024 8:30pm	
				Testing End Date/Time	10/09/2024 8:55pm	
Test Case ID	Module	Test Case Title	Test Steps	Expected Result	Status	Comments
UAT001	Register Account	Register an BillHub account	1. User clicks “Sign Up Now” button. 2. User enters his phone number. 3. User fills in the OTP received via SMS. 4. User enters email address. 5. User fills in the OTP received via email. 6. User fills in his personal information. 7. User make sure his password filled in is strong enough and meet the requirements. 8. User submit the create account request.	User will successfully create an account that he can use for login	Pass	Unable to revert to previous step during wrong input, must start over again.
UAT002	Login Account	Login into BillHub account created	1. User clicks “Login” button. 2. User enters email and password. 3. User clicks “Login” button.	User will be navigated to “Home” page with welcome message	Pass	-
UAT003	Register Bill	Register a bill from a billing company	1. User clicks “Add Bill” button at the middle of home screen. 2. User selects a billing company from the list. 3. User enters the account number of the bill.	The registered bill will be visible at “Home” page and “My Bill” page	Pass	-

			4. User enters a name for the bill. 5. User clicks the “Register” button. 6. User repeat steps 1-5 to register 2-3 different bills.			
UAT004	Delete Bill	Remove a registered bill from the account	1. User selects a bill in home page or “My Bill” page. 2. User clicks the rubbish bin icon on the top right corner of the “Bill Detail” screen. 3. User confirms the bill deletion alert.	The registered bill will be removed from the account.	Pass	-
UAT005	Reload Credit	Reload BillHub credits that can be used for bill payments and auto-billing	1. User clicks the “reload” button at the home page. 2. User selects an amount to reload or enter it via the input box. 3. User enters banking information inside payment gateway. 4. User approves the transaction.	The credits amount shown in “Home” page will be increased with exact amount that the user reloaded.	Pass	-
UAT006	Pay Bill	Pay one of the registered bills or multiple bills in one transaction	1. User clicks “Pay” button at the home page. 2. User ticks the bills that he wants to pay. 3. User edits the amount that he wants to pay if he doesn’t want to pay the amount of outstanding value. 4. User clicks “Pay” button. 5. User confirms the total payment amounts and the number of selected bills for payment. 6. User selects a payment method. 7. User enter the bank information into payment gateway if he selects online FPX payment or card payment.	A payment receipt will be shown, showing the transaction ID, payment date, payment time, payment method, bills paid, and their payment amounts respectively.	Pass	Unable to share the payment receipt
UAT007	View Bill and History	View the details of a bill including billing and payment history	1. User clicks any of the registered bills in home page or “My Bill” page. 2. User clicks on the “history” button.	The bill details such as account number, bill’s owner, phone number, billing history and	Pass	-

				payment history will be shown.		
UAT008	View Bill Analysis	View the analysis of bills issued and payment made	1. User selects the “Bill Analysis” at the middle of home page. 2. User selects to view “payment made analysis” or “bills issued analysis” 3. User selects “Month” or “Year” at the bottom of the Analysis Page. 4. If user selects “Month”, he should choose a month to view the analysis of the selected month.	The bills issued and payment made analysis are shown in the form of Pie Chart, Table Form (Month) and Line Chart (Year)	Pass	-
UAT009	Set Bill Reminder	Turning on/off bill reminder and customize the reminder	1. User selects “Bill Reminder” from the drawer menu. 2. User selects a bill from the list of registered bills or select “All bills”. 3. User clicks the switch to set the reminder to on or off. 4. User selects the preferred reminder methods. 5. User selects the preferred reminder timing. 6. User clicks the “Save” button.	On the selected reminder timing, the reminder message will be sent to user via SMS or notification, or email based on their customization.	Pass	-
UAT010	Set Auto-Billing	Setting up a bill to pay automatically	1. User clicks Auto-Billing located at the Home Page. 2. User selects a bill from the list of registered bills. 3. User selects a payment date. 4. User selects a payment amount. 5. User set the maximum transaction allowed for outstanding amount or set the fixed amount to be paid.	The bills will be paid during the selected payment date. The receipt will be generated and can be viewed from payment history.	Pass	-
UAT011	View Notification	Check the messages or reminders sent to the user	1. User clicks the bell shape icon located at the top right corner of the home page. 2. User selects one of the notifications from the notifications list.	The notification message and details will be shown in a pop-up modal.	Pass	-
UAT012	Delete Notification	Remove the notification	1. User clicks the bell shape icon	The notification deleted will be	Pass	-

		that has been checked	located at the top right corner of the home page. 2. User selects one of the notifications from the notifications list. 3. User clicks on the "DELETE" button 4. User clicks on the "DELETE" button again on the delete confirmation alert.	removed from the notification page.		
UAT013	Edit Bill Name	Add or change the name used the represent the registered bill	1. User selects a registered bill from the home page or "My Bills" page. 2. Inside the "Bill Detail" page, clicks the edit button located at the right-hand side of the name. 3. User enters a new name for the registered bill. 4. User clicks the "Save" button located at the bottom of the page.	The bill in the home page will now appear with the updated name instead of its previous name.	Pass	-
UAT014	Log Out	Quit from using the logged-in account	1. User clicks the menu button located at the top-left corner of home page. 2. User clicks the "Log Out" button from the drawer menu shown.	The user's information will be removed from the async storage, login screen will be shown.	Pass	-

User Acceptance Test (UAT)						
Tester's Name	Bill Payer 5			Testing Start Date/Time	10/09/2024 9:50pm	
				Testing End Date/Time	10/09/2024 10:11pm	
Test Case ID	Module	Test Case Title	Test Steps	Expected Result	Status	Comments
UAT001	Register Account	Register an BillHub account	1. User clicks "Sign Up Now" button. 2. User enters his phone number. 3. User fills in the OTP received via SMS. 4. User enters email address. 5. User fills in the OTP received via email. 6. User fills in his personal information.	User will successfully create an account that he can use for login	Pass	-

			7. User make sure his password filled in is strong enough and meet the requirements. 8. User submit the create account request.			
UAT002	Login Account	Login into BillHub account created	1. User clicks “Login” button. 2. User enters email and password. 3. User clicks “Login” button.	User will be navigated to “Home” page with welcome message	Pass	-
UAT003	Register Bill	Register a bill from a billing company	1. User clicks “Add Bill” button at the middle of home screen. 2. User selects a billing company from the list. 3. User enters the account number of the bill. 4. User enters a name for the bill. 5. User clicks the “Register” button. 6. User repeat steps 1-5 to register 2 -3 different bills.	The registered bill will be visible at “Home” page and “My Bill” page	Pass	Not clearly shown the registration is successful or not.
UAT004	Delete Bill	Remove a registered bill from the account	1. User selects a bill in home page or “My Bill” page. 2. User clicks the rubbish bin icon on the top right corner of the “Bill Detail” screen. 3. User confirms the bill deletion alert.	The registered bill will be removed from the account.	Pass	-
UAT005	Reload Credit	Reload BillHub credits that can be used for bill payments and auto-billing	1. User clicks the “reload” button at the home page. 2. User selects an amount to reload or enter it via the input box. 3. User enters banking information inside payment gateway. 4. User approves the transaction.	The credits amount shown in “Home” page will be increased with exact amount that the user reloaded.	Pass	-
UAT006	Pay Bill	Pay one of the registered bills or multiple bills in one transaction	1. User clicks “Pay” button at the home page. 2. User ticks the bills that he wants to pay. 3. User edits the amount that he wants to pay if he doesn’t want to pay	A payment receipt will be shown, showing the transaction ID, payment date, payment time, payment method, bills paid, and their	Pass	-

			the amount of outstanding value. 4. User clicks “Pay” button. 5. User confirms the total payment amounts and the number of selected bills for payment. 6. User selects a payment method. 7. User enter the bank information into payment gateway if he selects online FPX payment or card payment.	payment amounts respectively.		
UAT007	View Bill and History	View the details of a bill including billing and payment history	1. User clicks any of the registered bills in home page or “My Bill” page. 2. User clicks on the “history” button.	The bill details such as account number, bill’s owner, phone number, billing history and payment history will be shown.	Pass	-
UAT008	View Bill Analysis	View the analysis of bills issued and payment made	1. User selects the “Bill Analysis” at the middle of home page. 2. User selects to view “payment made analysis” or “bills issued analysis” 3. User selects “Month” or “Year” at the bottom of the Analysis Page. 4. If user selects “Month”, he should choose a month to view the analysis of the selected month.	The bills issued and payment made analysis are shown in the form of Pie Chart, Table Form (Month) and Line Chart (Year)	Pass	Unable to view bills issued histories in “Payment History” module, only can for a selected bill in “Bill History” module.
UAT009	Set Bill Reminder	Turning on/off bill reminder and customize the reminder	1. User selects “Bill Reminder” from the drawer menu. 2. User selects a bill from the list of registered bills or select “All bills”. 3. User clicks the switch to set the reminder to on or off. 4. User selects the preferred reminder methods. 5. User selects the preferred reminder timing. 6. User clicks the “Save” button.	On the selected reminder timing, the reminder message will be sent to user via SMS or notification, or email based on their customization.	Pass	Easy to use
UAT010	Set Auto-Billing	Setting up a bill to pay automatically	1. User clicks Auto-Billing located at the Home Page.	The bills will be paid during the selected payment date. The receipt will be generated	Pass	Easy to use

			2. User selects a bill from the list of registered bills. 3. User selects a payment date. 4. User selects a payment amount. 5. User set the maximum transaction allowed for outstanding amount or set the fixed amount to be paid.	and can be viewed from payment history.		
UAT011	View Notification	Check the messages or reminders sent to the user	1. User clicks the bell shape icon located at the top right corner of the home page. 2. User selects one of the notifications from the notifications list.	The notification message and details will be shown in a pop-up modal.	Pass	Too many notifications regarding payment made
UAT012	Delete Notification	Remove the notification that has been checked	1. User clicks the bell shape icon located at the top right corner of the home page. 2. User selects one of the notifications from the notifications list. 3. User clicks on the “DELETE” button 4. User clicks on the “DELETE” button again on the delete confirmation alert.	The notification deleted will be removed from the notification page.	Pass	Cannot delete multiple notification at once
UAT013	Edit Bill Name	Add or change the name used the represent the registered bill	1. User selects a registered bill from the home page or “My Bills” page. 2. Inside the “Bill Detail” page, clicks the edit button located at the right-hand side of the name. 3. User enters a new name for the registered bill. 4. User clicks the “Save” button located at the bottom of the page.	The bill in the home page will now appear with the updated name instead of its previous name.	Pass	-
UAT014	Log Out	Quit from using the logged-in account	1. User clicks the menu button located at the top-left corner of home page. 2. User clicks the “Log Out” button from the drawer menu shown.	The user’s information will be removed from the async storage, login screen will be shown.	Pass	-