**CONSPRING: A WATER BILLING AND COLLECTION SYSTEM**

**FOR BARANGAY CONALUM, INOPACAN, LEYTE**

A Research / Capstone Project

Presented to the Faculty of the

College of Computer Studies and Information Technology,

Southern Leyte State University

In Partial Fulfillment of the Requirements for the

Degree of Bachelor of Science in Information Technology

By

Abrantes, Jexter Jay A.

Bayhonan, Crissha P.

Sarong, Johann Von E.

Tafalla, Arce L.

Mr. Rhoderick D. Malangsa

Adviser

May 2022

Republic of the Philippines

**SOUTHERN LEYTE STATE UNVERSITY** Sogod, Southern Leyte



Website: [**www.slsuonline.edu.ph**](http://www.slsuonline.edu.ph/)

Email: **slsumaincampus@gmail.com** op@slsuonline.edu.ph

Telefax No. (053) 382-3294

***College of Computer Studies and Information Technology***

**APPROVAL SHEET**

The Capstone Project Study entitled **CONSPRING: A Water Billing and Collection System for Barangay Conalum, Inopacan, Leyte**  prepared and submitted by Abrantes, Bayhonan, Sarong and Tafalla has been examined and is recommended for approval and acceptance.

RECOMMENDED:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RHODERICK D. MALANGSA** \_ | |  |  | **RHODERICK D. MALANGSA** |
|  |  |  |  |  |
| Adviser |  | ITSO Manager | | Research Facilitator |



APPROVED by the Committee on Oral Examination with a grade of PASSED on \_\_\_\_\_\_\_\_\_\_\_

**ALEX C. BACALLA**

Chairman

|  |  |  |
| --- | --- | --- |
| **MR. JORTON TAGUD, MSIT** |  | **MR. RENE RADAZA, MSIT** |
| Member |  | Member |



ACCEPTED and APPROVED in partial fulfillment of the requirements in Bachelor of Science in Information Technology.

**ALEX C. BACALLA**

Dean, CCSIT

Date: \_\_\_\_\_\_\_\_\_\_\_\_

**DEDICATION**

This research is dedicated to our Almighty God, who has provided us with wisdom, motivation, and strength throughout the process.

To our instructors, who guided and prepared us to achieve our future goals.

To our parents and family members, who have always supported and assisted us in our challenges, as well as provided us with inspiration in both good and tough times.

To the members of this team who have never wavered in their determination to overcome all of the obstacles we have faced during our collegiate careers.

To our Capstone Adviser, who dedicated his time to guiding, teaching, and motivating us to complete this project.

Finally, we want to thank our Alma Mater, Southern Leyte State University, for nurturing and molding us into the people we are today.

**ACKNOWLEDGEMENT**

This study would not be feasible without the assistance and guidance of everyone who stands beside the proponents. The proponents wished to express their heartfelt gratitude to each and every one of them.

To our very kind and supportive Capstone Adviser, Mr. Rhoderick D. Malangsa, who guided and advised us about the things that we should do in this project.

To our Capstone Instructor for consistently motivating us to complete the study and for carefully monitoring our progress.

And to our friends, classmates, and schoolmates who stayed with us day and night to help us with our studies.

**ABSTRACT/EXECUTIVE SUMMARY**

Barangay Conalum is one of the 20 barangays in the municipality of Inopacan, province of Leyte. As one of the proponents being a resident of the barangay, she have noticed that the water billing and collection system there is still done manually. The manual technique is prone to errors, takes a long time, and frequently loses records or papers. This also places a lot of responsibility and pressure on the treasurer and the barangay's lone water tender. Not to mention the inconvenience it causes to the customers. Establishing this system will aid the treasurer and water tender in their duties thus being able to address the said issues. If the proposed initiative proves to be a success, the water billing system in the barangay will be surely improved. It will allow faster searching of data, issuing of bills, tracking of records, more reliable data gathering and checking, and safer storing of important customer information.

Keywords: Water Billing System, water billing and collection, manual technique, tracking of records, important customer information

**TABLE OF CONTENTS**

**Content Page No.**

Title Page

Approval Sheet 2

Dedication 3

Acknowledgement 4

Abstract 5

Table of Contents 6

Lists of Figures 8

Lists of Tables 9

Chapter I – INTRODUCTION

1.1 Project Context 10

1.2 Purpose and Description of the Project 10

1.3 Objectives of the Project 10

1.4 Scope and Limitations of the Project 11

Chapter II – REVIEW OF RELATED LITERATURE

2.1 Related Literature/ Theoretical Background 13

2.2 Related Studies 13

Chapter III – TECHNICAL BACKGROUND

3.1 Technicality of the Project 17

3.2 Details of the Technologies to be Used 17

3.3 How the Project will Work 19

Chapter IV – METHODOLOGY

4.1 Environment 20

4.2 Requirements Specification 21

4.2.1 Operational Feasibility 21

Fishbone Diagram 21

Functional Decomposition Diagram 21

4.2.2 Technical Feasibility 22

Compatibility Checking 22

Relevance of the Technologies 22

4.2.3 Economic Feasibility 22

Cost and Benefit Analysis 22

Cost Recovery Scheme 23

4.2.4 Requirements Modeling 23

Data and Process Modeling 24

Object Modeling 25

4.2.5 Risks Assessment Analysis 29

4.3 Design 30

4.3.1 Output and User Interface Design 33

4.4 Development 40 4.4.1 Software Specification 40

4.4.2 Hardware Specification 40

4.4.3 Program Specification 40 4.4.4 Programming Environment 41

4.4.5 Deployment Plan 41

4.4.6 Test Plan 42

Conclusion and Recommendations 42

Implementation Plan 43

Bibliography 45

Appendices 46

Glossary 89

**LIST OF FIGURES**

Figure 1 Structural Layout Design

Figure 2 Waterfall Model

Figure 3 Organizational Chart

Figure 4 Fishbone Diagram

Figure 5 Functional Decomposition Diagram

Figure 6 Requirements Model

Figure 7 Context Diagram

Figure 8 Data Flow Diagram

Figure 9 Use Case Diagram

Figure 10 Class Diagram

Figure 11 Sequence Diagram

Figure 12 Activity Diagram

Figure 13 Homepage

Figure 14 Login Page

Figure 15 Dashboard

Figure 16 Add New Customer

Figure 17 Customer List

Figure 18 Customer Record – View

Figure 19 Customer Record - Edit

Figure 20 Customer Record – Delete

Figure 21 Send Message

Figure 22 Add New Billing Period

Figure 23 Billing Records

Figure 24 Billing Records – View

Figure 25 Billing Records – Edit

Figure 26 Billing Records – Delete

Figure 27 Reports

Figure 28 Account Management

Figure 29 Customer Dashboard

Figure 30 Customer Payment Records

Figure 31 Customer Messages

Figure 32 Customer Account Management

**LIST OF TABLES**

Table 1 Cost and Benefit Analysis

Table 2 Cost Recovery Scheme

Table 3 Risk Assessment Analysis

Table 4 Deployment Plan

Table 5 Test Plan

Table 6 Project Implementation Checklist

Table 7 Implementation Contingency Table

Table 8 Infrastructure/Deployment Table

**CHAPTER I. INTRODUCTION**

**Project Context**

Barangay Conalum is one of the 20 barangays in the municipality of Inopacan, province of Leyte. As one of the proponents being a resident of the barangay, she have noticed that the water billing and collection system there is still done manually. Because of this, there have been different issues like consumers' handwritten receipts were lost resulting to confusion about their payment records, miswritten amounts, and having unorganized collection of records. The manual method is prone to errors, time-consuming and often misplaces records or documents. This also puts a lot of responsibility and pressure for the treasurer and the only water tender that works for the barangay. Not to mention the inconvenience that it also gives to the consumers. To resolve these problems, I, together with my groupmates, have came up with the idea of developing a system that will help the treasurer and water tender in their work. If the proposed project will be successful, the researchers will be planning to expand the project to other barangays in the municipality.

**Purpose and Description of the Project**

The main purpose of this project is to develop a water billing and collection system for Barangay Conalum, in the municipality of Inopacan. This system will be used by the water tender to be able to read water meters and collect payments more accurately and efficiently compared to the manual way before. All of the consumers' data will also be stored at the database of the system allowing a more organized storage of information rather than just being written on a piece of paper or on a notebook. The proposed system will be comprised of the billing, collecting, and storing of the records in a database. First, the system will generate a billing statement indicating the amount of water consumed and its corresponding cost. Then, the collection process which includes the receiving of payments by the water tender and then recording it later by the treasurer. Lastly, the transaction records will be saved into the system's database.

Neumann (2010) states that water billing system is a necessary tool to assist small municipalities, utility providers, bookkeepers, operators, managers, and auditors in unifying their water billing services, in order to provide consistent and accurate billing information to clients and service recipients. It is essentially software that processes data and produces invoices in a given format, facilitating and unifying the billing process.

**Objectives of the Project**

Nowadays, almost everything is automated. Most IT organizations use multiple automation technologies. An automated system is composed of elements designed to perform a set of tasks that have been programmed. Operational and repetitive tasks become less of a burden and makes your life simpler and easier.

Advantages commonly attributed to automation include higher production rates and increased productivity, more efficient use of materials, better product or service quality, improved safety, and shorter workweeks for labor. Higher output and increased productivity have been two of the biggest reasons in justifying the use of automation.

The general objective of this study is to develop an automated system of calculating water bills, collecting payments, sending notifications to consumers and storing all user information in a database.

This will make the job of the water tender easier, helping him to read meters and collect payments faster compared to the manual method before. The website can also provide real-time data to ensure a more reliable information to the consumers.

Specific objectives:

1. To create a module that will help the water tender for quick billing and collecting of payment from the consumers.
2. To make a system with a database for easy viewing of files and to prevent data manipulation and possible data loss.
3. To develop a module that can help the consumers to view if they have a remaining balance or if they have already paid.
4. To integrate the water billing and collection system from the water tender and treasurer’s office to the consumers.
5. To develop functionality to make it easier to issue bills to water users and to alleviate human errors in the water tender’s work.

**Scope and Limitations of the Project**

**Scope of the Study**

There are currently 580 water consumers in the barangay. The current system involves manually computing the total balance after reading the meter. Then afterwards, collecting the payment from the consumer. This method consumes more time and effort for the water tender. With the proposed system, it will already be automated, reducing the manual work needed to be done.

One of the features of the system is that after reading the meter, the water tender just needs to input the necessary information on his mobile phone, and then the receipt will automatically be printed using the thermal receipt printer which is connected to the phone. If the consumer wants to pay their bill immediately, the water tender will just click the “paid” button on the system and if not, the water tender will choose “pending payment”. The consumer can later pay their bill at the treasurer’s office. Aside from that, the water tender, as well as the treasurer, can access all the information about the consumers' status using a laptop or cellphone. Consumers will also be notified a day before the due date and disconnection date in case they haven’t paid yet. They can access the website and log into their account to check their payment status or pending balance.

In order to complete the project, first, we need to coordinate with the barangay officials and ask for the list of all the consumers in the barangay. After gathering the data, we need to make an account for each consumer which they will use in order to access the website and login to their respective accounts. After the programming is done, the system will be installed at the treasurer’s office. And lastly, the thermal receipt printer will be given to the water tender. This will be done on a time frame of three months.

**Limitations of the Study**

The study only includes households from Barangay Conalum. Only those that have water connections will be included. Aside from that, every household can only have two connections. The system is about water billing and collection only and doesn’t include other bills like electricity, internet, etc. The payment will only be in the form of cash. Other modes like online payments or via credit cards will not be accepted.

**CHAPTER II - REVIEW OF RELATED LITERATURE**

In this chapter, the proponents gathered a review from different studies and literature that are related to the proposed study.

**Related Literature**

According to an article by mics@dmin, titled “Advantages of Automated Billing Software”, The importance of efficient billing cannot be overstated. It is through this task that a company collects payment from customers. One way of ensuring accurate and timely billing is through automation. Automated billing systems are designed to manage the entire cash collection process from data gathering to sending invoices. The system contains critical elements of billing such as data input and verification, billing codes, tracking payments, and security. The advantages of electronic billing are far reaching. Although migrating from manual to automated process seem daunting, its numerous benefits make it all worthwhile.

It is also stated on the article "Critical Analysis of Billing System" (2016), "For a water billing, which is complex, repetitive and has voluminous data, computerization is recommended. Computerization overcomes many of the defects in the manual system, is fast and gives a control on the system."

According also to the book entitled “Principle of Water Rates, Fees and Charges” which is published by AWWA on 2017, if the utility billing system is computerized, the time required for the preparation of the bill tabulation may be relatively short, perhaps only one or two weeks.

Computerized System and improved efficiency have been the focus of entrepreneurs. As with many business scenarios, getting rid of paper improves efficiency, reduces human error and allows information to flow to an infrastructure without a time-consuming data input process. There is also less chance of handwritten orders being misread and a higher customer turnaround as customers will be served faster. (Ambre, 2016)

Scott (2011) concludes that automation increases the accuracy of the data, as human error is less likely to occur. The completeness of the data is also improved as the automated systems will process all the data in an efficient manner.

**Related Studies (Local)**

According to Pagaduan A., Raguindin F., and Garbin J. in their study, "As people go along through life, it is inevitable to encounter many changes that can help in one’s daily needs. One of these changes is technology. It presents big help to lessens one’s work and make it more efficient especially when applied to business establishments. Information technology plays a big role in processing such needs. Its growing uses are becoming more essential, that it has already become a trend. It is easy to use, fast and accurate, and improves the efficiency of an individual or an organization. One such application that benefits most from computerization is the processing of an assessment and billing for an organization’s clients, since the process ensures that fast and accurate processing of data to accommodate a client’s request the soonest possible time."

A water billing system is an automated system that makes the complex task of billing easy, fast and accurate. This paper presents the development of an Automated Water Billing System for the local government unit of Hinunangan in the province of Southern Leyte, Philippines. The system constitutes two applications - Desktop and Mobile Reader. The System Development Life Cycle (SDLC) framework was adopted in the development process utilizing the Waterfall Model. Survey and interviews were also done as a supplementary technique of the fact-finding. The purposive sampling method was employed in the selection of respondents, and a researcher-made questionnaire was used in the systems evaluation. Weighted mean was used as a statistical treatment. Based on the results of the study, the developed system contributes a greater advantage in providing and delivering better operation, reporting, and services to the clientele. (Manun-og, M., Manun-og, M., Claridad, N., Tereso, R., & Libarios, J., 2018).

According to the Trece Martires Water District Billing and Collection System. In March 2002, Cruto, Legaspi and Rosero made a study on Trece Martires Water District Billing and Collection System. Based on their study, the problem is that the Trece Martires Water District Billing and Collection System apply the manual processing of its billing and collection clerks and other office personnel. The records of the customers were record in paper works that are less reliable to keep those files. When a customer asks for his/her records, it will add up the workloads of the employees. To accurately keep those records, they developed an automated Transaction Processing System that speed up the customer’s daily transaction, utilized efficient and safe storage of customer’s records and a more reliable and accurate computation of customer’s water bills.

According to the Casimiro Development Corporation Water Billing System. In March 2001, Bartolome, Alonde, Yanuaria and Atendido made a study about the development of Casimiro’s Water Billing System. Based on their study, the problem was that Casimiro Development Corporation water billing system operates 13 projects and approximately 450 units per project that run in a manual-based environment. To develop more accurate files, they created a well-structured database that lessened and speed up the works in computations in voluminous files with accurate and error-free files.

By automating document processes instead of relying on manual methods, businesses reduce the cost, increase the Speed and improve the quality of information exchange with clients. These Outcomes result in greater efficiency in billing and collections, ultimately contributing to overall profitability and return on IT investments. (Gentugao, L., Baclas, R.J., Pasicolan, J., Monsanto, J., Ogatis, M.A., 2014).

**Related Studies (International)**

The case studies indicate that any successful billing practice must ensure that bills are raised on a monthly basis and should be volumetric-based, such that customers pay for what they consume. This makes it mandatory for the service provider to adopt 100 percent metering of all its customer connections. Service providers must realize that an effective billing and collection system that rides on these principles can bring about immediate improvements in revenue streams. However, to ensure that such practices remain effective, it is absolutely essential that providers have updated, robust, and computerized customer databases such that the billing function can be easily implemented. A computerized system of billing and an updated and complete customer database is a must if a service provider is looking to maintain high billing efficiency. Providers must also ensure that customer databases are updated and computerized, through robust accounting, record-keeping, regular systematized checks, and billing procedures. Effective billing and collection systems are a critical component for ensuring the viability of a service provider. Improving these has an immediate impact on the revenue streams of a service provider that can, in turn, help in improving services. (Agrawal, P.C., Shukla, S., 2005 & 2006).

Billing is a critical function of most Water Boards especially towards sending the bill to their consumers. Most problems, currently seen, result from the manual processes followed. Calculation errors, delays in system updating and bill sending, also report tracking issues are the major problems that water boards find difficult to find answers for. (Bello, Z., 2014).

While according to Dhumale, Thombare and Bangare (2018), Wastage of water in process of manually operated water pump, human error associated with manually operated system, improper bill creation, delay in bill availability and delay in payments process are problems that lead to the development of an “Android based Automatic Water Billing (AWB)” system. In AWB system a low cost water flow meter is proposed, which measures flow rate of water passed through the supply pipe of particular user and bills are created according the usage of water by that particular user. This provides control on usage of water without applying more charges to the user. The deadline is given to the user to pay bill otherwise the water supply will be stopped by supplier. Bills and notifications regarding due payment are forwarded to the user.

The billing development involves receiving billing report from a variety of networks, formative the billing rates associated with the billing records, calculate the cost for each billing record, aggregating this records periodically to produce invoices, sending invoices to the customer, and collecting payments received from the customer. The overall maintenance cost of these systems is smaller and the results will be helpful and optimistic (Sakarde,C., Wath, B., Thakre T., Borkar R., Burange R.A., 2014).

Water is a important and limited source. Therefore collecting the data of the water which is being used is very important. The system implemented will provide accurate and real time water billing. This will overcome existing system by providing accurate data of usage of water. This will reduce the cost of the whole system and will limit and reduce the usage of the water. Monthly billing cycle can be maintained to limit the use of precious natural resource water (Priyanka, C., Muneeb, C., Piyush, I., Chaudhari, H.P., 2019).

**References:**

http://mics.org/advantages-of-automated-billing-software/

Critical Analysis of Billing System. (2016, Sep 13). Retrieved from <https://studymoose.com/critical-analysis-of-billing-system-essay>

https://www.awwa.org/Portals/0/files/publications/documents/M1Ed7LookInside.pdf

Ambre, J (2016). ASIST Automated Water Billing System. International Journal of Research in Management & Business Studies (IJRMBS 2016). Vol. 3 Issue 1 Jan. - Mar. 2016 Retrieved from http://ijrmbs.com > janelyn.pdf

<http://jhelai4b.blogspot.com/2007/11/lan-based-assessment-and-billing-system.html>

Manun-og, M., Manun-og, M., Claridad, N., Tereso, R., & Libarios, J. (2018). AUTOMATED WATER BILLING SYSTEM OF HINUNANGAN MUNICIPALITY. Innovative Technology and Management Journal, 1(1). Retrieved from [https://journal.evsu.edu.ph/index.php/itmj/article/view/41](https://journal.evsu.edu.ph/index.php/itmj/article/view/41?fbclid=IwAR2YK_hPzUX3vSv8w0TtU-coOvhX8gdttCizAQa_NvKhoJIdv6IR2fbseSs)

<https://www.academia.edu/28274021/WATER_BILLING_SYSTEM>

https://studylib.net/doc/9284799/2.1.1-computerized-water-billing-system

Agrawal, P. & Shukla S. (2005 & 2006). Developing Effective Billing and Collection Practices. Retrieved from [http://documents1.worldbank.org](https://l.facebook.com/l.php?u=http://documents1.worldbank.org/?fbclid=IwAR2LvSkM1gvngXFSryaLkvDO_bdVPFXrPopdWdnuKqyeDcYM1xrqjqTTc50&h=AT1iPnw2cEeQJzNcfeC9AAuKC1qV5Z_23bVBWSJU2cG_ojVau6ROQ4Dhz2GjeEY12yjUjri6WxoVQOVRl8w6rUElNnqQCExmnk31Y-6rL0r5LUdqpW7XbRpPo9xnBfx66kLvlw) > 441190WSP0IN0P1ive0billing01PUBLIC1.pdf

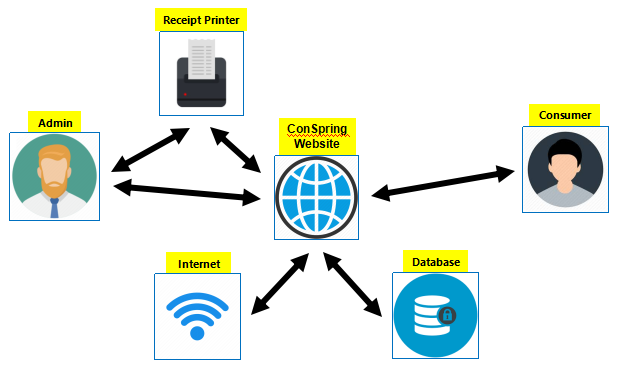
Bello, Z. (2014). DESIGN AND IMPLEMENTATION OF SMS BASED WATER BILLING SYSTEM (A CASE STUDY OF BAGUDO LOCAL GOVERNMENT WATER BOARD, KEBBI STATE). Retrieved from https://www.academia.edu/37775337/DESIGN\_AND\_IMPLEMENTATION\_OF\_SMS\_BASED\_WATER\_BILLING\_SYSTEM\_A\_CASE\_STUDY\_OF\_BAGUDO\_LOCAL\_GOVERNMENT\_WATER\_BOARD\_KEBBI\_STATE

<https://www.irjet.net/archives/V5/i8/IRJET-V5I8169.pdf>

"AUTOMATIC WATER BILLING SYSTEM USING ARDUINO", International Journal of Emerging Technologies and Innovative Research (www.jetir.org | UGC and issn Approved), ISSN:2349-5162, Vol.7, Issue 3, page no. pp1146-1149, March-2020, Available at : <http://www.jetir.org/papers/JETIR2003170.pdf>

http://www.dynamicpublisher.org/gallery/21-ijsrr-d-2139.final.pdf

**CHAPTER III. TECHNICAL BACKGROUND**

**Technicality of the Project**

After gathering all related research about the proposed project, the proponents have decided to use a website for the system. One major advantage of using a website is it is very accessible across different devices for users. Other necessary tools and software will also be used along the way in developing the system. With the use of the website, the consumers can easily view their pending balance using any device with a browser. The treasurer and the water tender can also view and update the records easily compared to the traditional process.

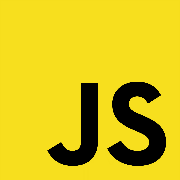
**Structural Layout Design**

Figure 1. Structural Layout Design

The diagram shows the structural layout of our proposed study. The system needs to be clear and easily understandable by the user in order to provide fast and reliable transmission and storing of information. The admin enters the website through the internet to login to the system and manage the customer records stored in the database. The bills will be printed with the use of the receipt printer. The customers can also access their account through the website.

**Details of the technologies to be used**

The software that the researchers will be using is JavaScript for the front-end, Laravel for the back-end and MySQL Database for the database. JavaScript will be used both on the client and the server-side that allows the web pages to interact. Laravel can be used for authentication, routing, sessions, and caching. MySQL will be used for the system to record the data in the database.



JavaScript is commonly used for creating web pages. It allows us to add dynamic behavior to the webpage and add special effects to the webpage. On websites, it is mainly used for validation purposes. JavaScript helps us to execute complex actions and also enables the interaction of websites with visitors.

Laravel is primarily used for building custom web apps using PHP. It’s a web framework that handles many things that are annoying to build yourself, such as routing, templating HTML, and authentication. It aims to make the development process a pleasing one for the developer without sacrificing application functionality.

MySQL is an open-source relational database management system. Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.

The hardware includes the smartphone, portable thermal printer, personal computer and WiFi router. The smartphone will be used in calculating the balance after reading the meter. Then the thermal printer for printing the details on the receipt. Personal computers or laptops will be used in the treasurer’s office that are connected to the internet in order to access consumers’ information from the database.



A smartphone is a cell phone that allows you to do more than make phone calls and send text messages. Smartphones can browse the Internet and run software programs like a computer. Smartphones use a touch screen to allow users to interact with them.



Unlike inkjet printers, thermal printers don't spray liquid ink through a nozzle to produce images. Rather, thermal printers use tiny heating elements to activate or transfer pigments.

A personal computer is a general-purpose computer whose size, capabilities and original sale price make it useful for individuals, and is intended to be operated directly by an end-user with no intervening computer operator. A personal computer may be a desktop computer or a laptop,netbook, tablet or a handheld PC.



A WiFi router is an electronic device that sends data received from an Internet cable to other devices. It also acts as a wireless access point from which it shares data through the use of radio signals.

**How the project will work**



Figure 2. Waterfall Model

In developing the system, the waterfall model will be used. Waterfall model is a linear (sequential) software development life cycle model that describes development as a chain of successive steps. No phase can be started before or simultaneously with the previous or current one.

The first phase of the project development is the Requirement Analysis. The researchers first identified the problems of the current system and analyze different solutions that could be done to improve it. All of the needed requirements for the project will also be evaluated during this time.

Next is the second phase which is the Design. In this phase, the proponents will plan how the system will run or what step-by-step process will be followed. This also includes the designing of the overall look of the website and the user interface using JavaScript.

Then the third phase would be the Development or Implementation Phase. The actual source code is finally written in the third phase, implementing all models, business logic, and service integrations that were specified in the prior steps. The hard coding will be done during this stage using Laravel and MySQL Database.

The fourth phase is the Testing Phase. After the development phase, the product should experience a thorough quality assurance and software testing to discover defects in a system. Testers are involved in finding and reporting issues that need to be resolved. At this time, the system will be run multiple times to see if something goes wrong and to discover possible bugs and malfunctions. The treasurer and the water tender will be requested to try the system for the final acceptance testing and evaluation. Once everything is fixed, the project will be ready for deployment.

The fifth and last phase is the Maintenance Phase. As soon as the product is for placement to a live environment, it enters the ready maintenance phase of the software development life cycle. The maintenance phase engages not just the deployment of the software, but also support and maintenance that may be required to keep it functional and up-to-date. In this phase, the proponents will make sure that the system is running smoothly after the installation. The water tender’s smartphone and portable thermal printer should be functioning properly, as well as the treasurer’s personal computer. All needed instructions will be given by the proponents including tutorials and manuals if necessary.

**CHAPTER IV. METHODOLOGY**

**4. 1 ENVIRONMENT**

* **Locale**

The system is intended for the 2,203 people living in Barangay Conalum, Inopacan, Leyte. The selected barangay is chosen to test the effectiveness of the system.

* **Population of Study**

Barangay Conalum has 9 puroks which have 2,203 residents in total and approximately 200 households that are connected to the current barangay water system.

* **Organizational Chart/Profile**



Figure 3. Organizational Chart

This figure shows the organizational chart of Barangay Conalum Officials.

**4. 2 REQUIREMENT SPECIFICATIONS**

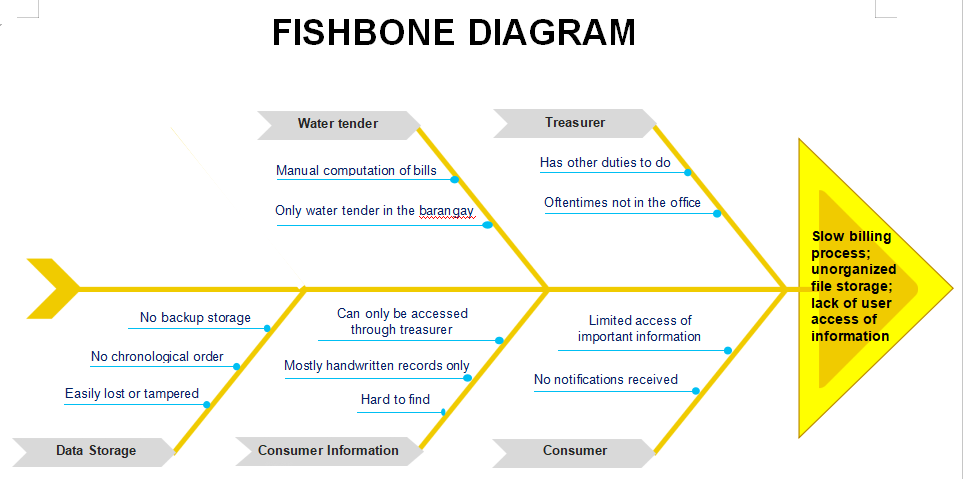
* + 1. **Operational Feasibility**
* **Fishbone Diagram**

Figure 4. Fish Bone Diagram

The left side of the diagram shows the different causes/factors that results on the right side of the diagram – inside the arrow, which are the negative impacts of the situation.

* **Functional Decomposition Diagram**

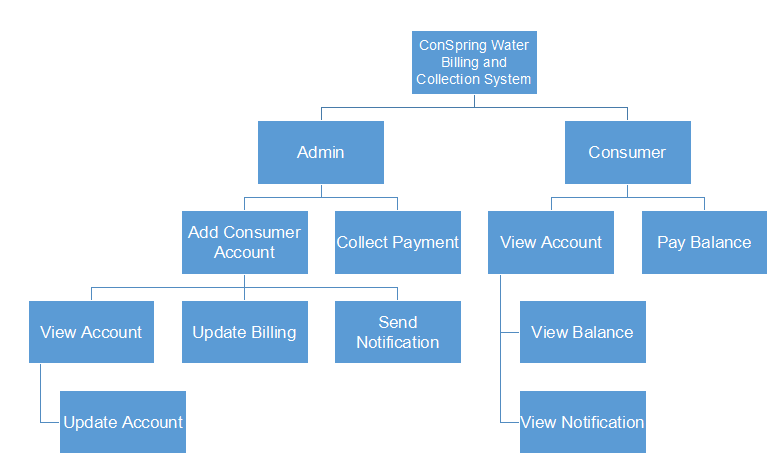
****

Figure 5. Functional Decomposition Diagram

Functional Decomposition Diagram shows all the major functions that can be encountered in the system.

* + 1. **Technical Feasibility**

**Compatibility Checking(Hardware/Software and other Technologies)**

The proposed project is a website-based program. To access the site, the user must have a stable internet connection using android phones/tablets that have android version not lowering from 5.0 and also laptops or PC with at least Intel core i3 processor with 2GB DDR3 memory and Windows 7, 8 or 10 versions. It can also be accessed using other software as long as it has an updated browser. In an admin’s computer or laptop, it must have a Windows OS that supports MySQL Database for the saving of the consumer information.

**Relevance of the technologies**

The manual record keeping and billing takes so much time and is prone to errors. Most of the time, these records are not secured and can easily be altered by making unauthorized changes by any individual, either intentional or accidental. Disasters like fire or flash floods can also cause damages or even result to total data loss. This project is created to make the work of the water tender easier and faster. To think of it, making hundred receipts is very time consuming. Through the implementation of the proposed system, it is expected that it will provide accurate data and increase productivity in the workplace since most of the manual computations and issuing of receipts will be automated. Consumers can easily access their account by logging into the website instead of going to the treasurer’s office to ask about their balances. The database will provide an enhanced and updated consumer records and will help prevent misplacing of records or documents. The researchers believe that the proposed system will be a contributing factor to the growth and enhancement of services in Barangay Conalum, for the benefit of the community and its stakeholders.

* + 1. **Economic Feasibility**

**Cost and Benefit Analysis**

Table 1. Cost and Benefit Analysis

|  |  |
| --- | --- |
| **EXPENSES** | **AMOUNT** |
| Internet Expenses | ₱ 4,000.00 |
| Paper and Photocopy Expenses | ₱ 1,000.00 |
| Transportation | ₱ 4,000.00 |
| Miscellaneous Expenses | ₱ 3,000.00 |
| TOTAL | ₱ 12,000.00 |

This table reflects the list of expenses or cost incurred in order to sustain the creation of the project.

**Cost Recovery Scheme**

Table 2. Cost Recovery Scheme

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EXPENSES** | **AMOUNT** | **AMOUNT** | **AMOUNT** | **AMOUNT** |
| Internet Expenses | ₱ 1,000.00 | ₱ 1,000.00 | ₱ 1,000.00 | ₱ 1,000.00 |
| Paper and Photocopy Expenses | ₱ 250.00 | ₱ 250.00 | ₱ 250.00 | ₱ 250.00 |
| Transportation | ₱ 1,000.00 | ₱ 1,000.00 | ₱ 1,000.00 | ₱ 1,000.00 |
| Miscellaneous Expenses | ₱ 750.00 | ₱ 750.00 | ₱ 750.00 | ₱ 750.00 |
| TOTAL | ₱ 3,000.00 | ₱ 3,000.00 | ₱ 3,000.00 | ₱ 3,000.00 |

This table reflects the division of expenses in order to gradually pay the cost incurred

upon the creation of the project.

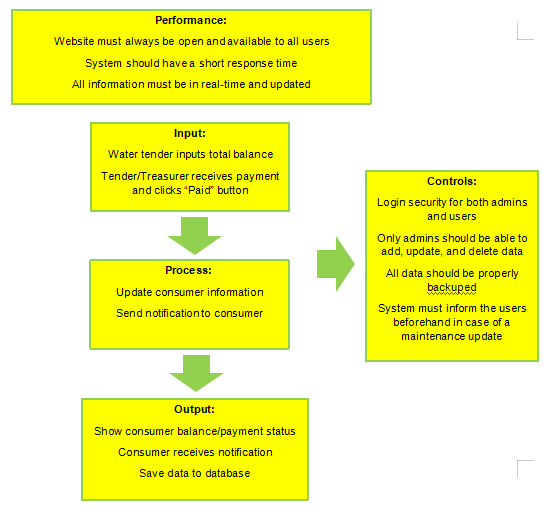
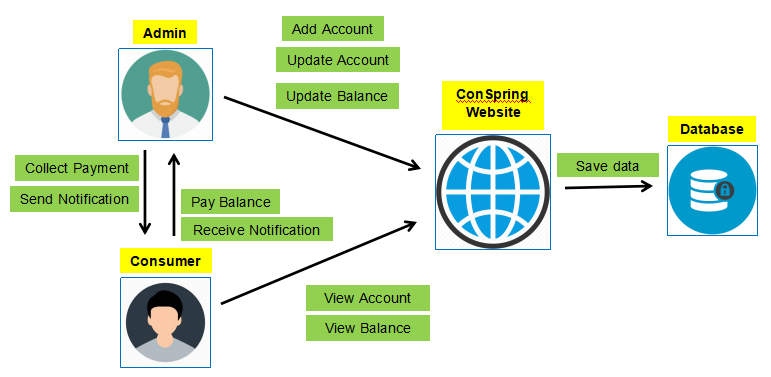
* + 1. **Requirements Modeling**

Figure 6. Requirements Model

This figure shows the needed requirements for the system.

**Data and Process Modeling**

**Context Diagram**

****Figure 7. Context Diagram

This figure shows the overall context of the system.

**Data Flow Diagram**

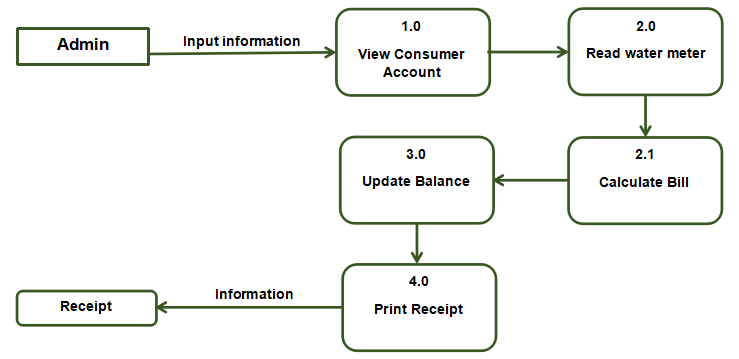
****

Figure 8. Data Flow Diagram

This figure shows the flow of data of the system.

**Object Modeling**

**Use Case Diagram**

**Water billing and Collection System**

**Water Tender and Treasurer**

**Consumer**

Figure 9. Use Case Diagram

This figure shows the description of the ways in which a user interacts with a system.

**Class Diagram**

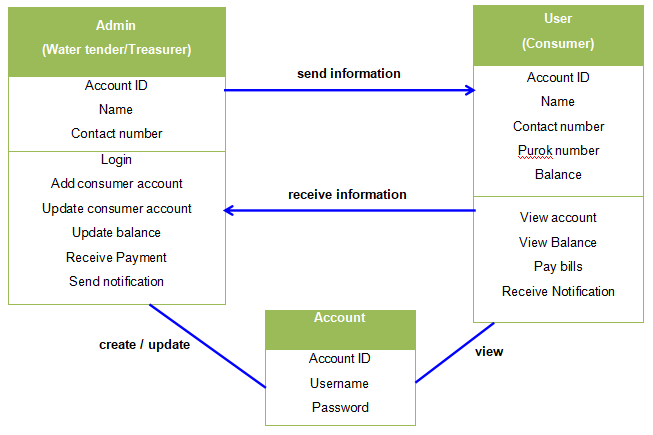
****

Figure 10. Class Diagram

This figure shows the different classes and the methods of the system.

**Sequence Diagram**

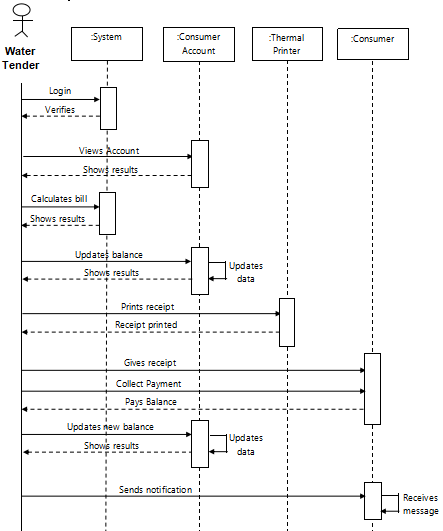
****

Figure 11. Sequence Diagram

This figure shows the sequence of events in using the system.

**Activity Diagram**

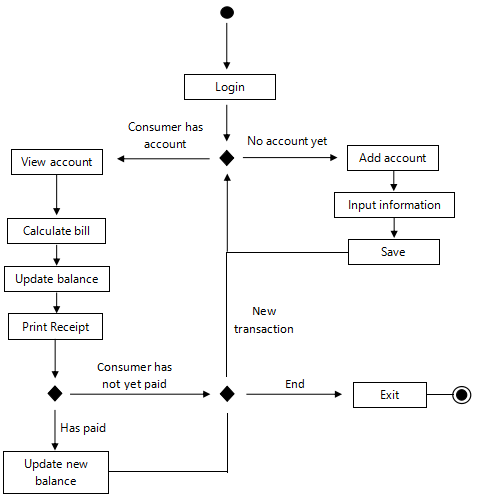
****

Figure 12. Activity Diagram

This figure displays the activity of the user in using the system.

* + 1. **Risks Assessment Analysis**

Table 3: Risk Assessment Analysis

|  |  |  |
| --- | --- | --- |
| **Risks** | **Possible Conflict** | **Mitigation Method** |
| Data Privacy | The possibility of data leakage is high. | Only admins will be allowed to access critical information. |
| Inappropriate data entry | Users wrong inputs might affect the functionality of the system. | Use input restriction functionality in the system. |
| Website Threats | Website might be easily vulnerable to threats. | Admins strengthen the security of the website. |
| Wrong computation of bills | Formula for bill calculation might be subjected to changes from time to time. | Admins will ensure that the system is updated occasionally. |

This table shows the different risks and possible conflicts the system might encounter and the corresponding method or solutions to prevent it.

**4. 3 DESIGN**

* + 1. **Output and User-Interface Design**

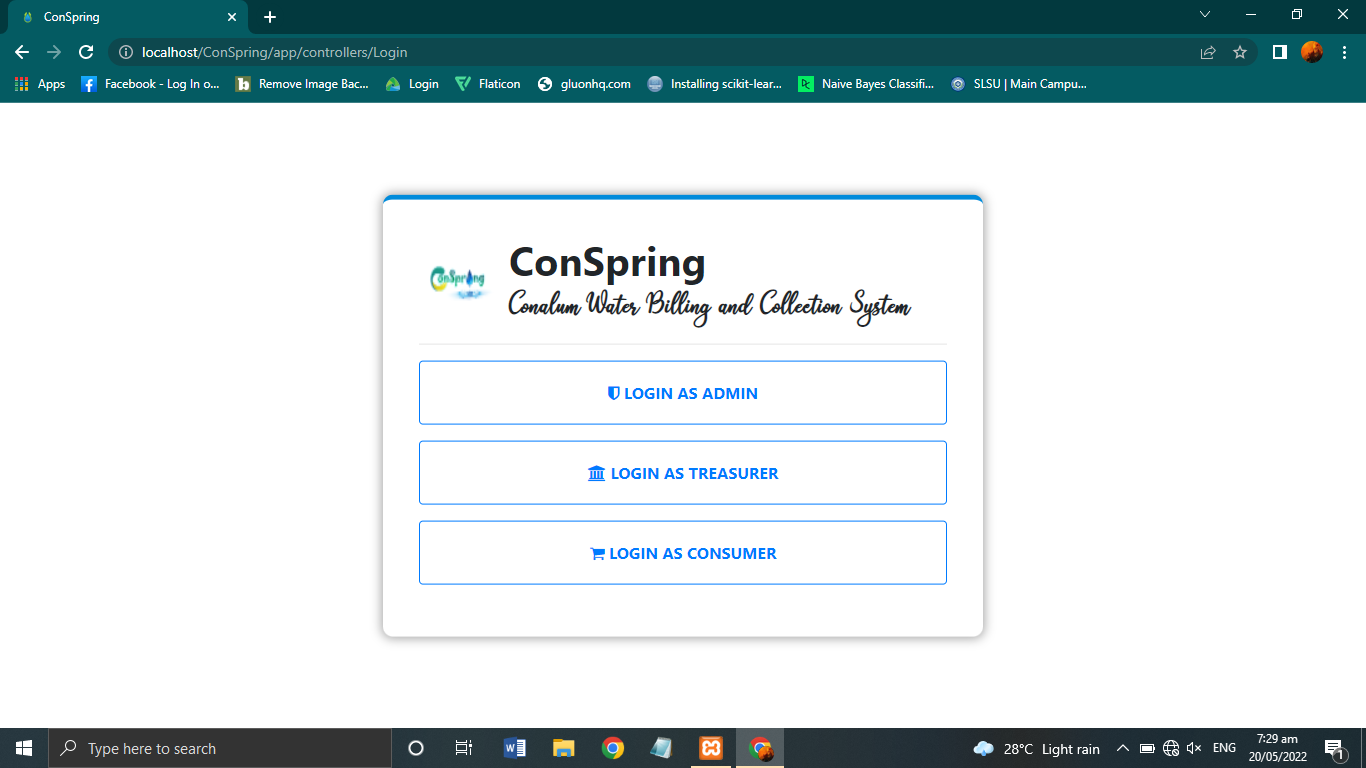
****

Figure 13. Homepage

First, it lets you choose to login either as admin, treasurer, or a customer. Both the Admin and Treasurer have the same capabilities in the system.

**Admin and Treasurer Mode**

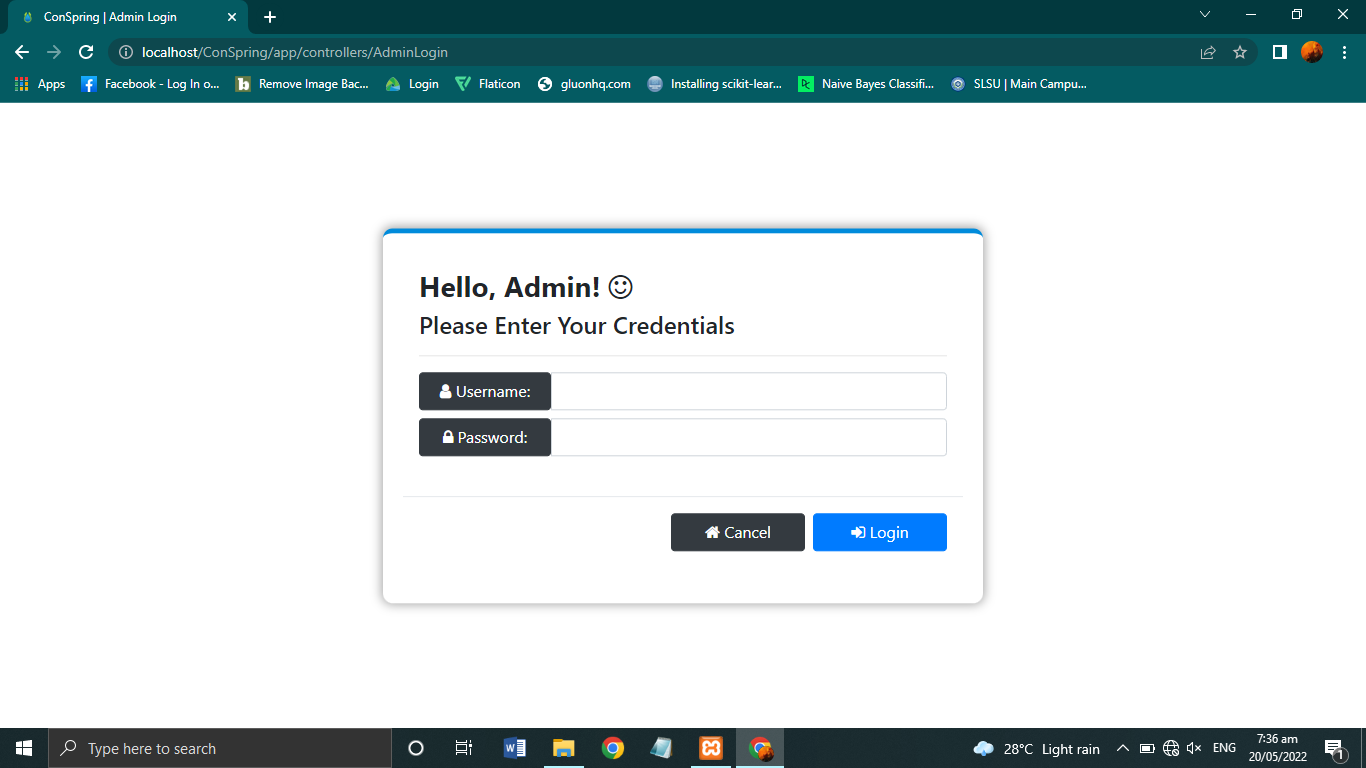
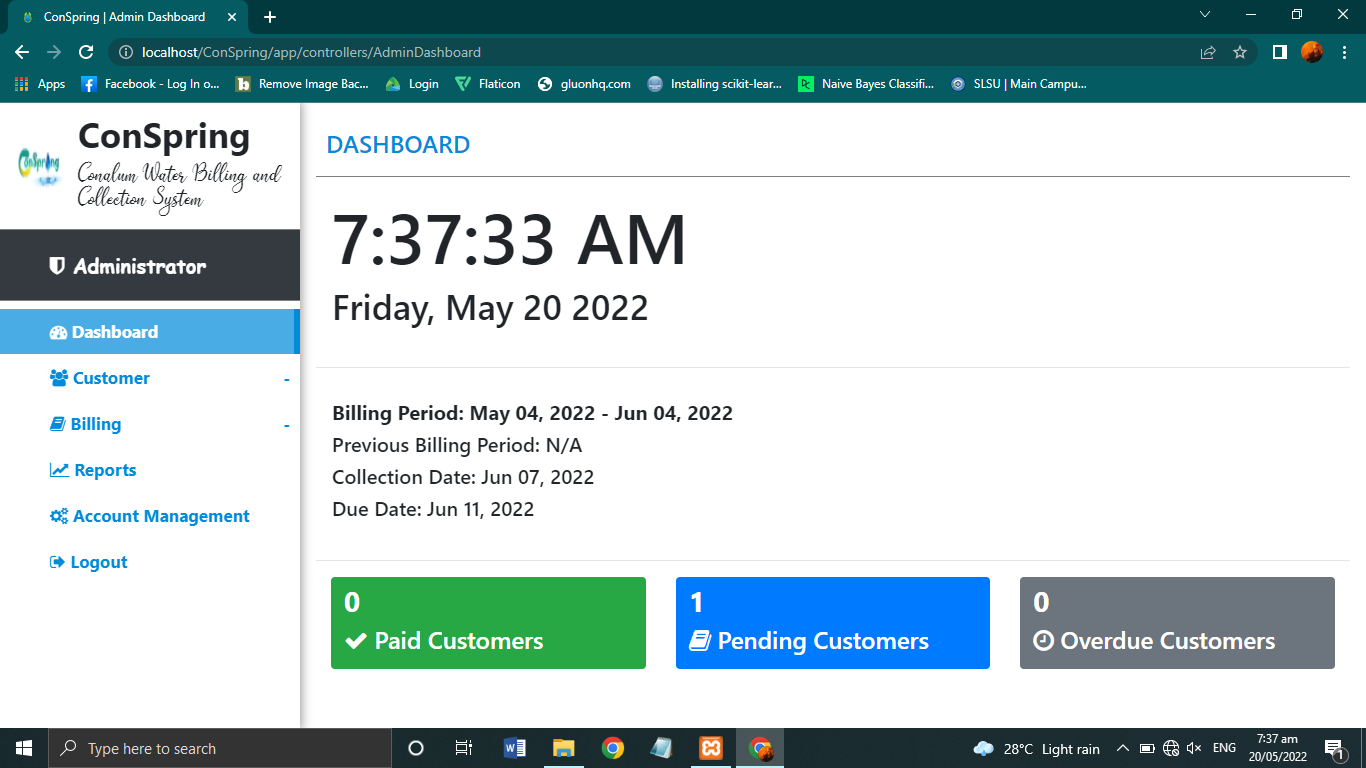


Figure 14. Login Page

By logging in, you are required to enter your Username and Password first.

This figure shows the Admin mode.

Figure 15. Dashboard

After logging in, the Dashboard, Customer, Billing, Reports, Account Management, and Logout tabs can be seen on the Left Side.

The Dashboard displays time, date, and other important information.

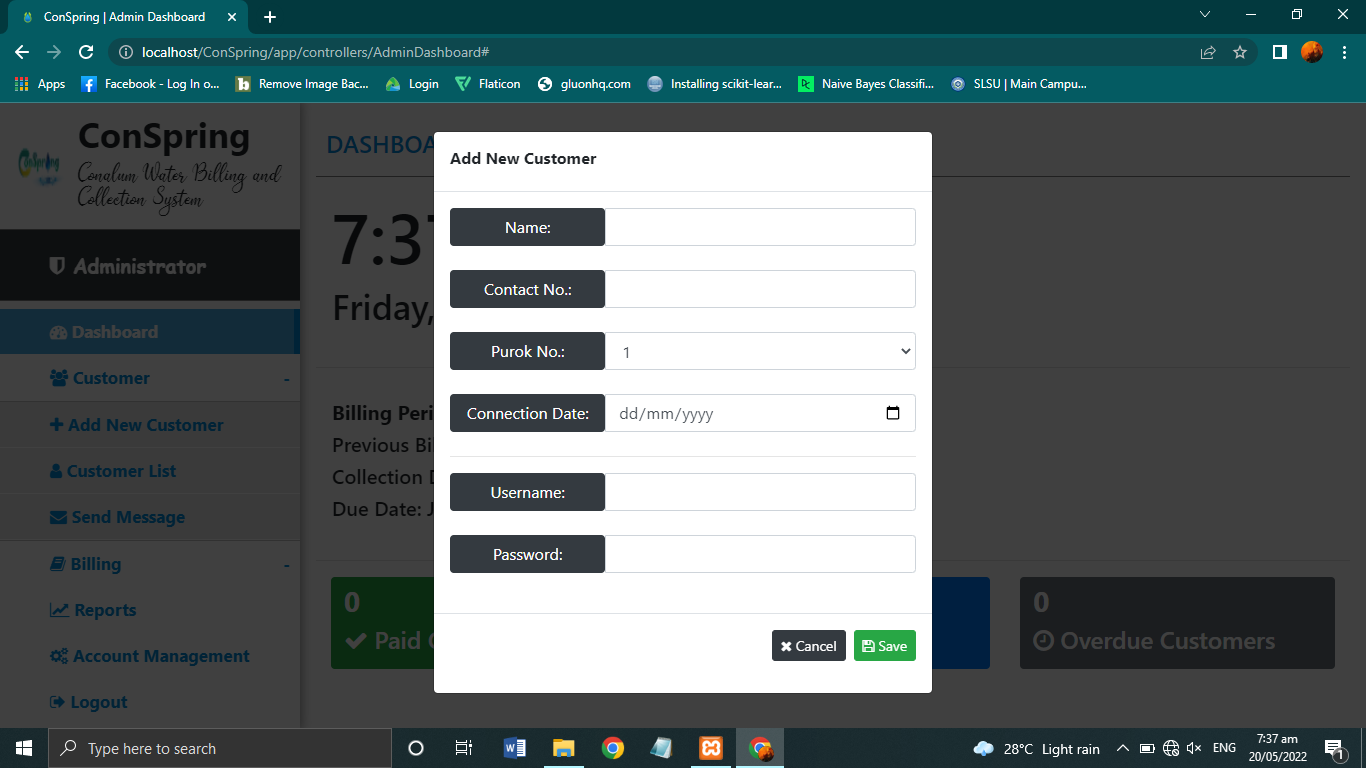


Figure 16. Add New Customer

Under customer, there are 3 sub-options.

Lets the user input details of a new customer and save it.

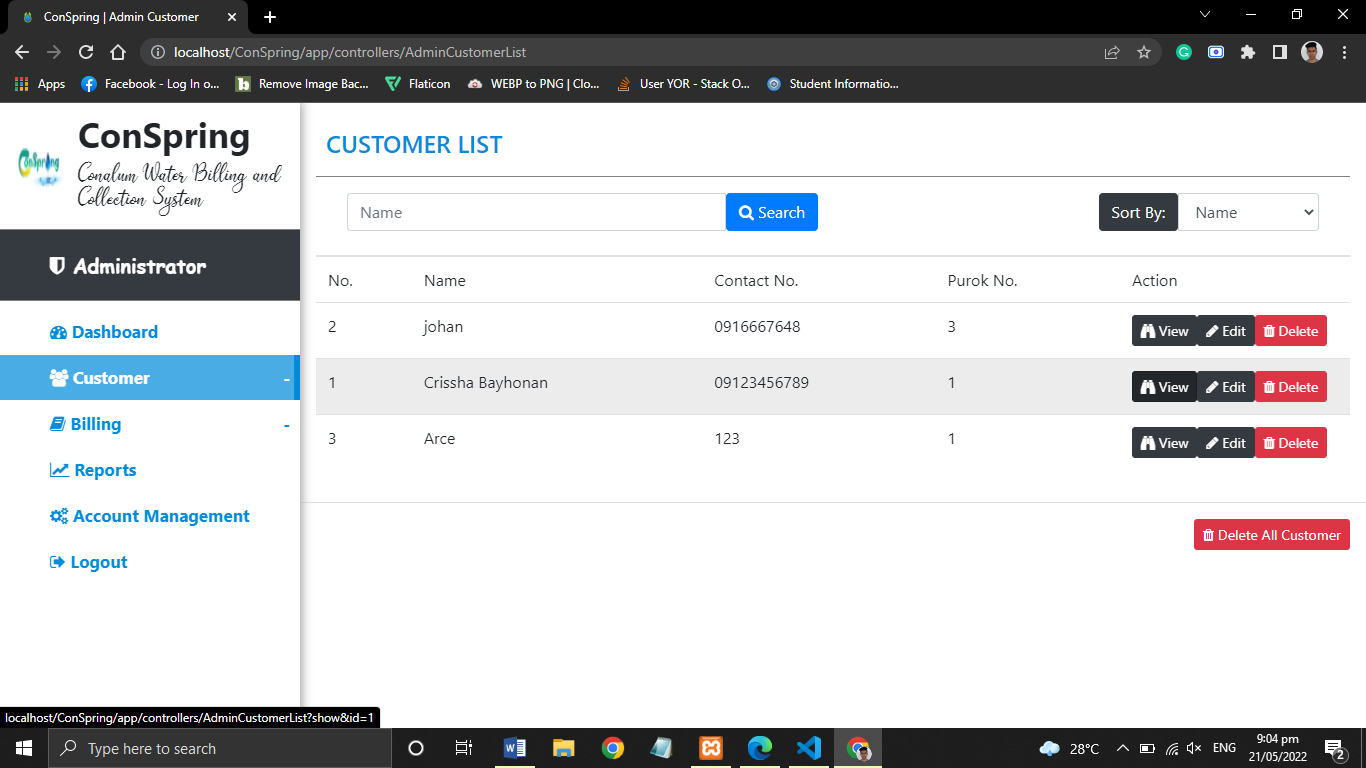
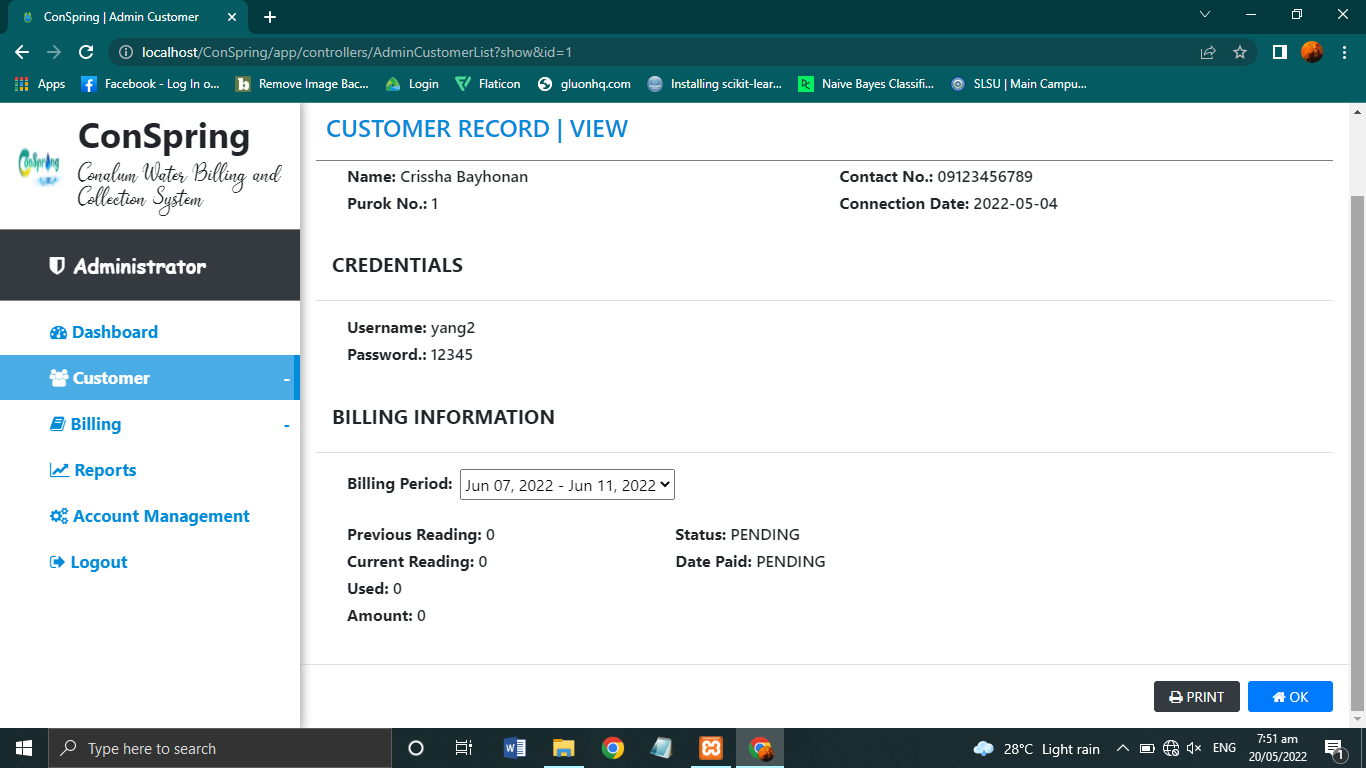


Figure 17. Customer List

Displays information of customers in a table. Also lets user to view, edit and delete customer data.

Figure 18. Customer Record – View

Displays detailed information of customers including their billing information. Also lets user print such information.

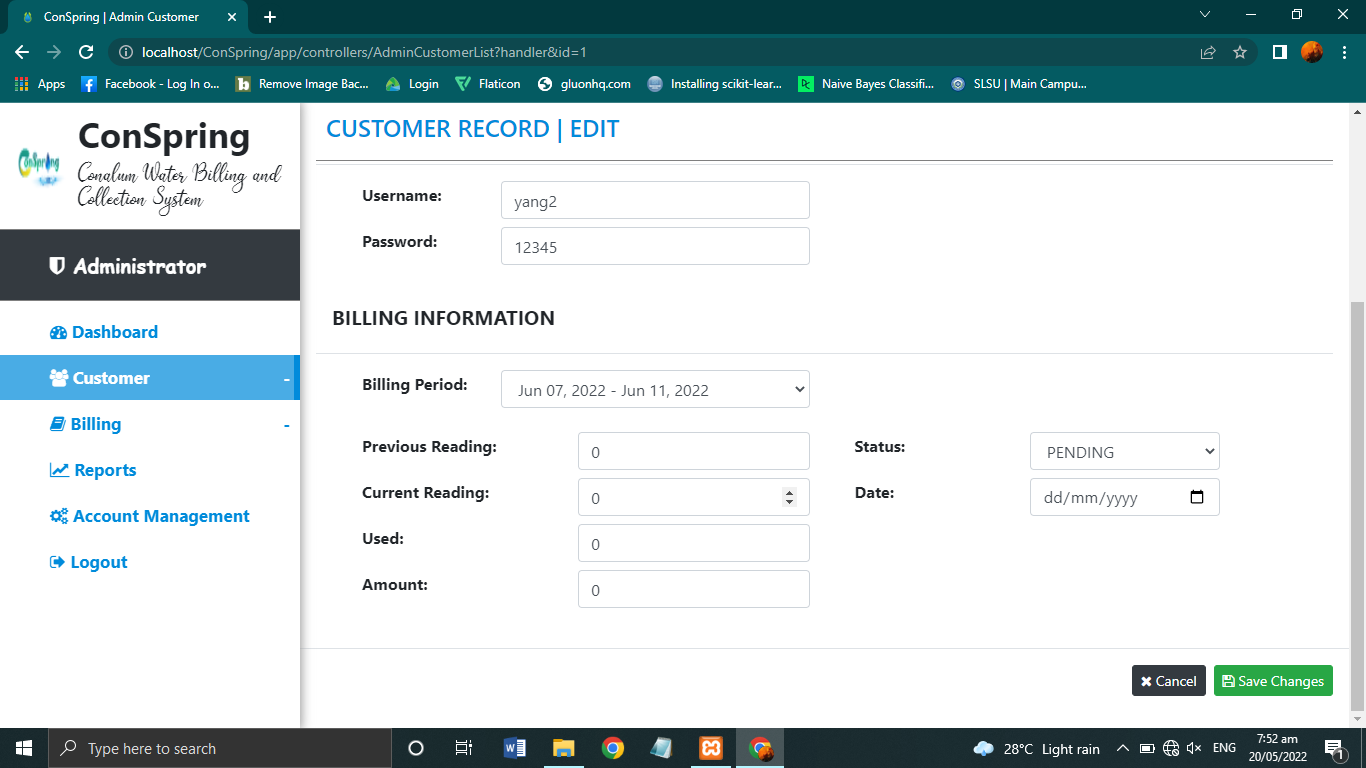


Figure 19. Customer Record - Edit

Displays information of customers and lets user edit their data.

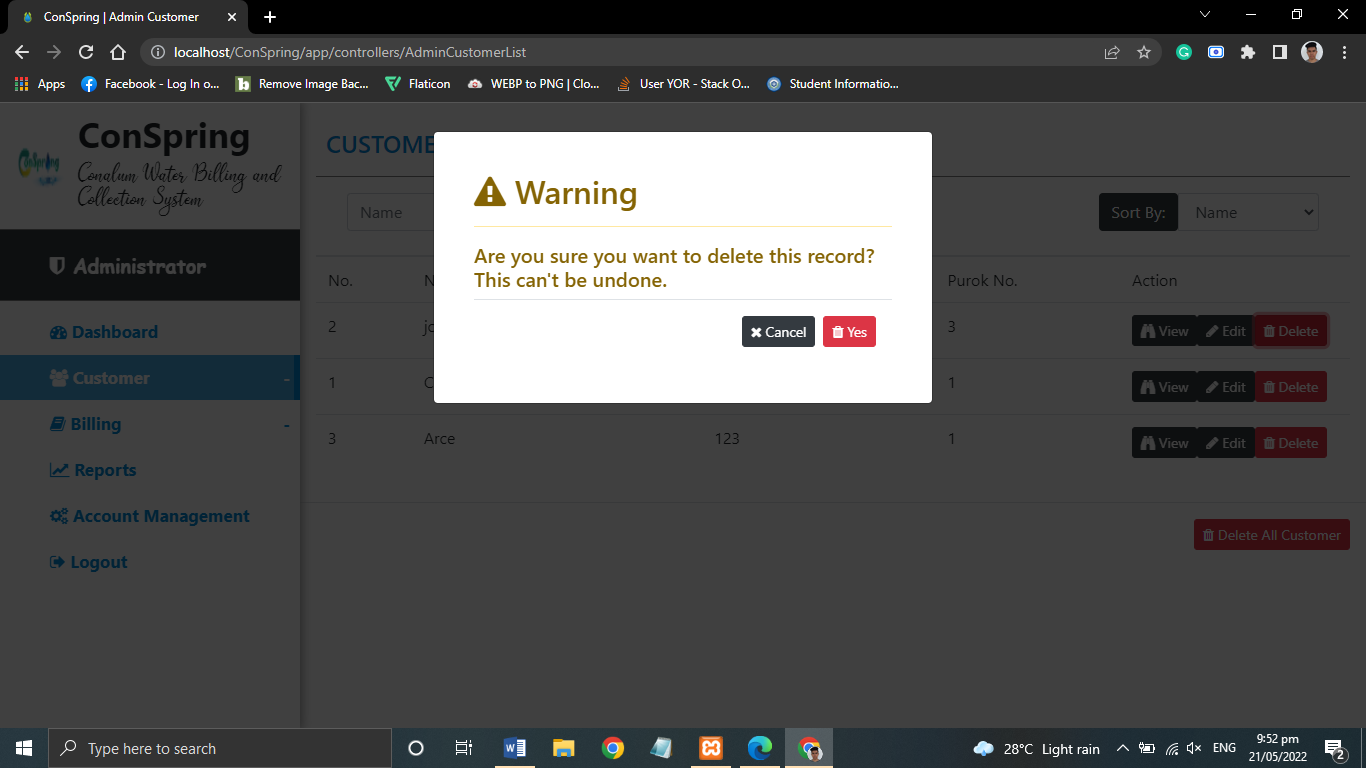


Figure 20. Customer Record – Delete

A warning dialog box will appear to confirm whether the user is sure to delete customer record.

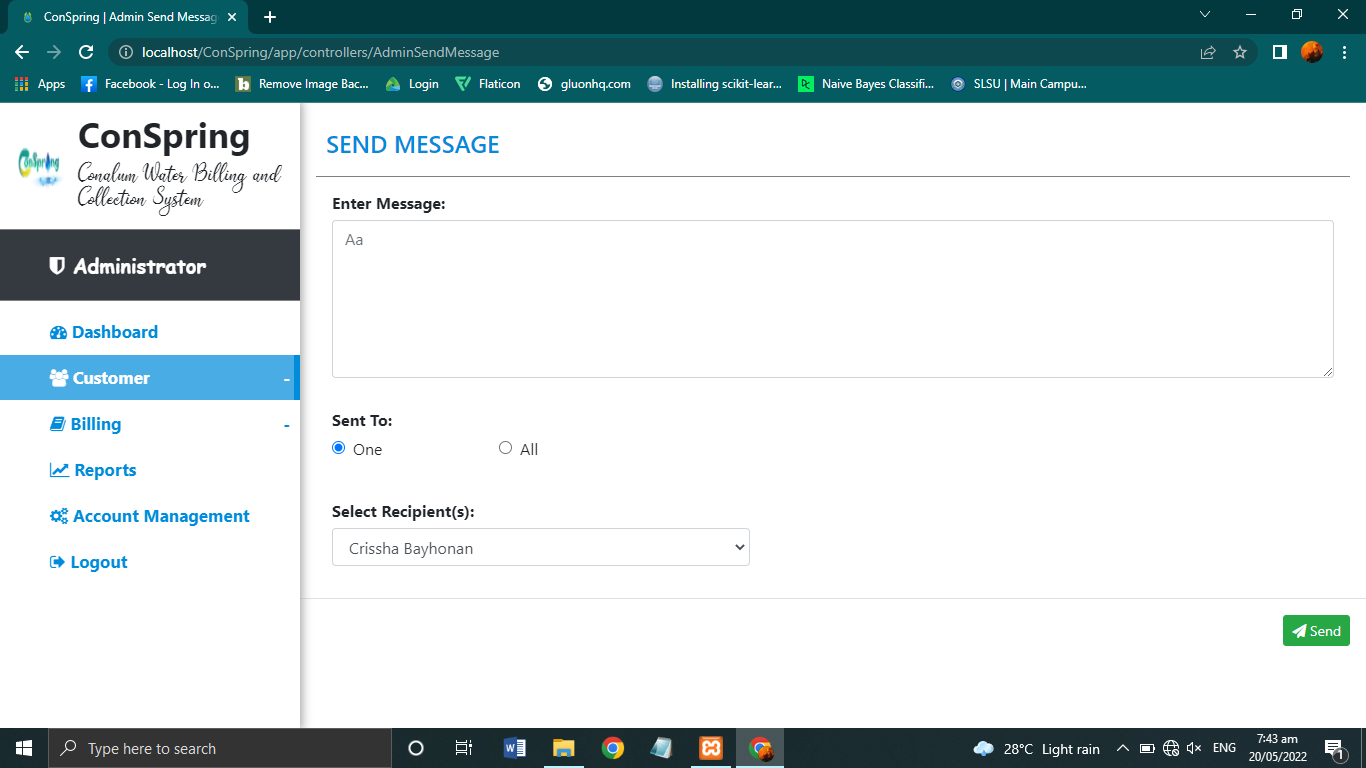


Figure 21. Send Message

Lets user input and send message to desired customer/s.

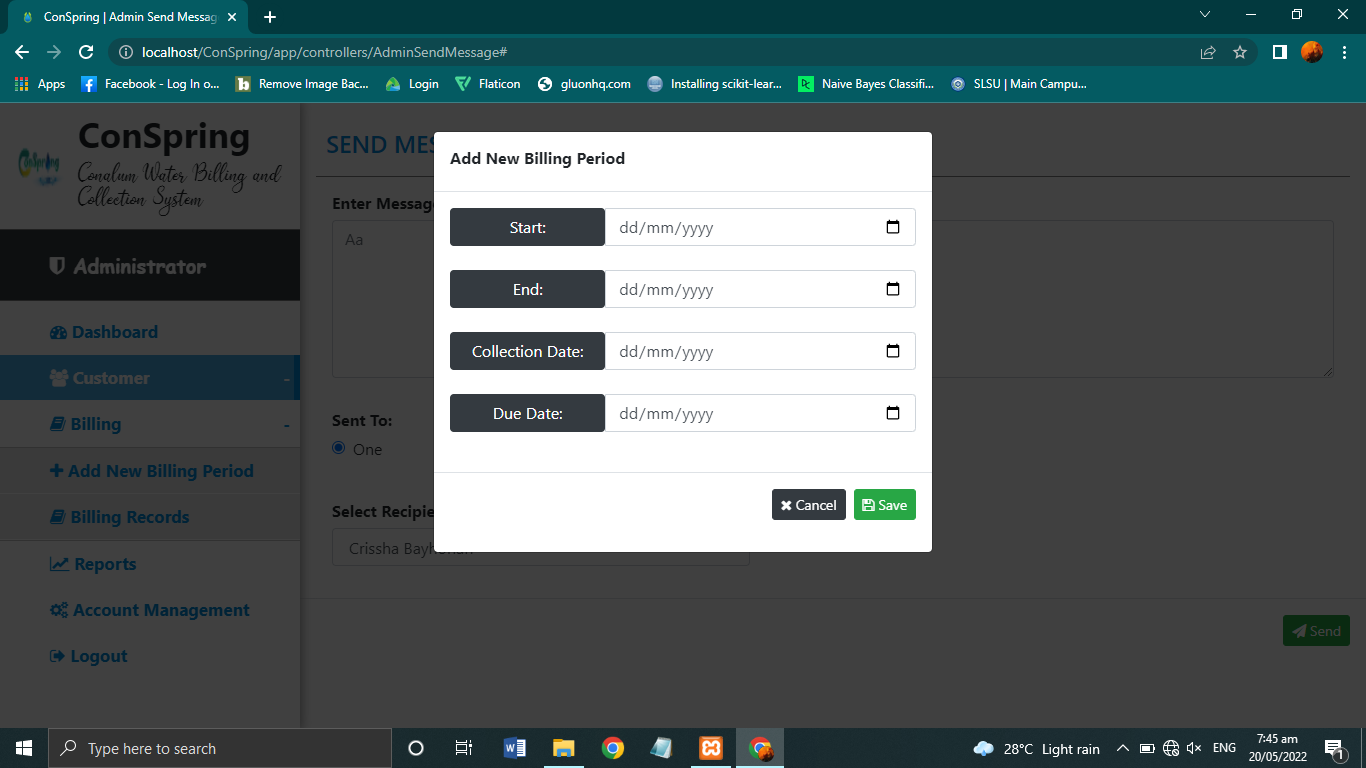


Figure 22. Add New Billing Period

Under Billing are two sub-options.

First is, Add New Billing Period which lets user add new billing period details and saves it.

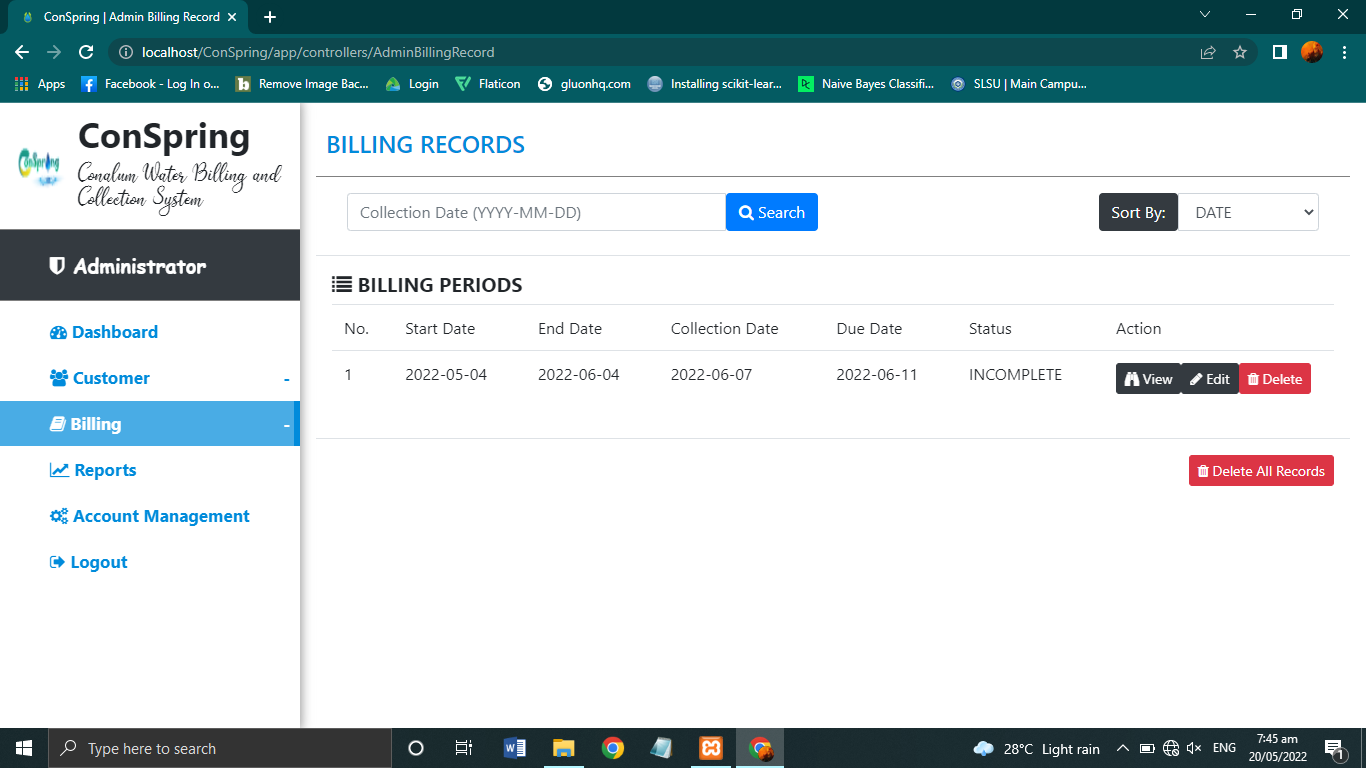
****

Figure 23. Billing Records

Displays all billing period details in a table. Also lets user view, edit, and delete billing period data.

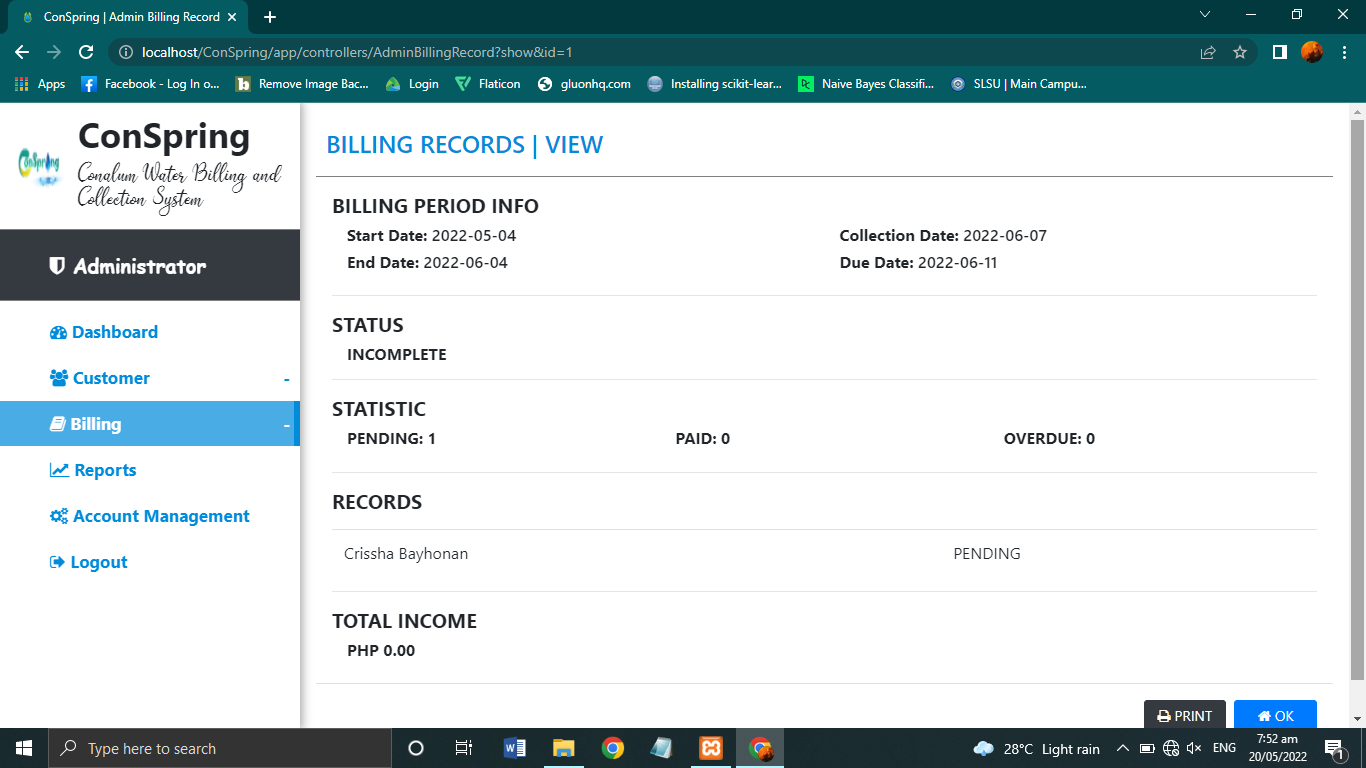


Figure 24. Billing Records – View

Displays detailed information of billing records. Also lets user print such information.

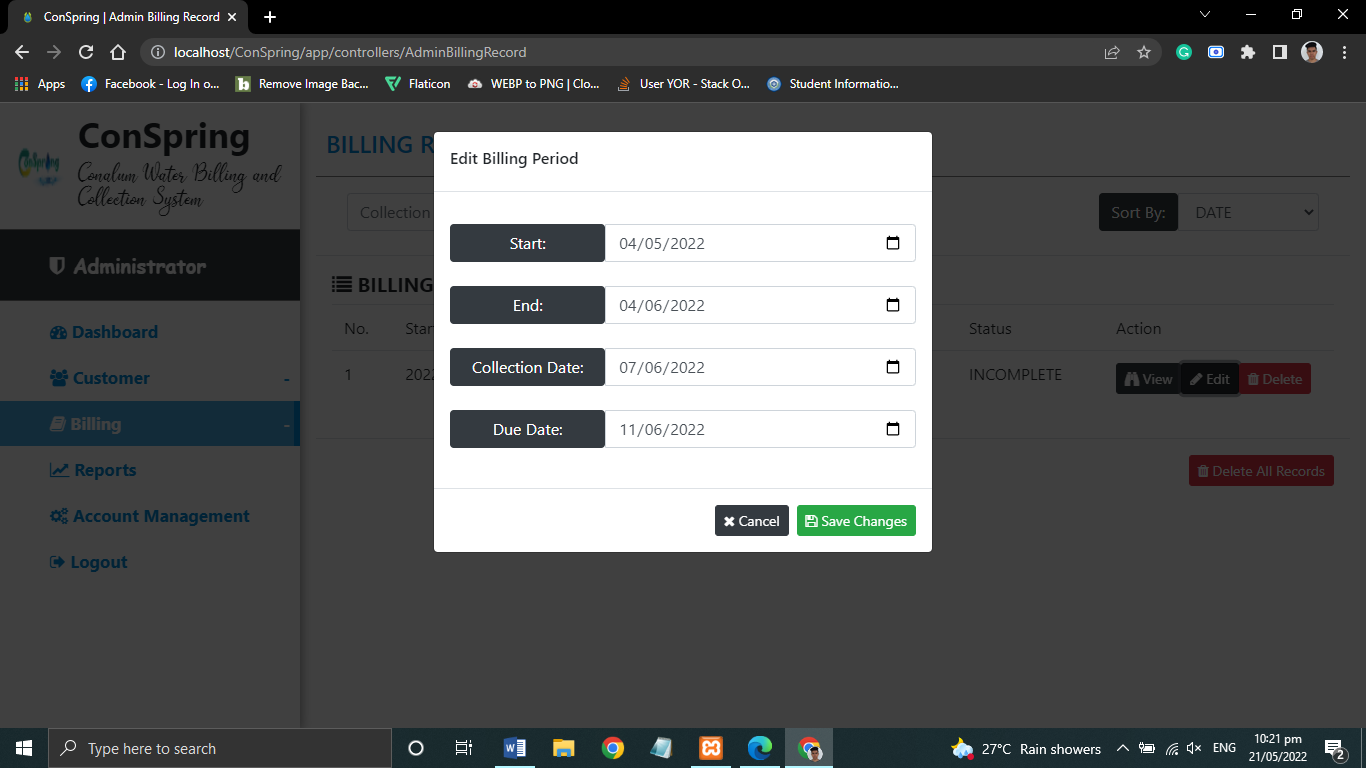


Figure 25. Billing Records – Edit

Displays detailed information of billing records. Also lets user print such information.

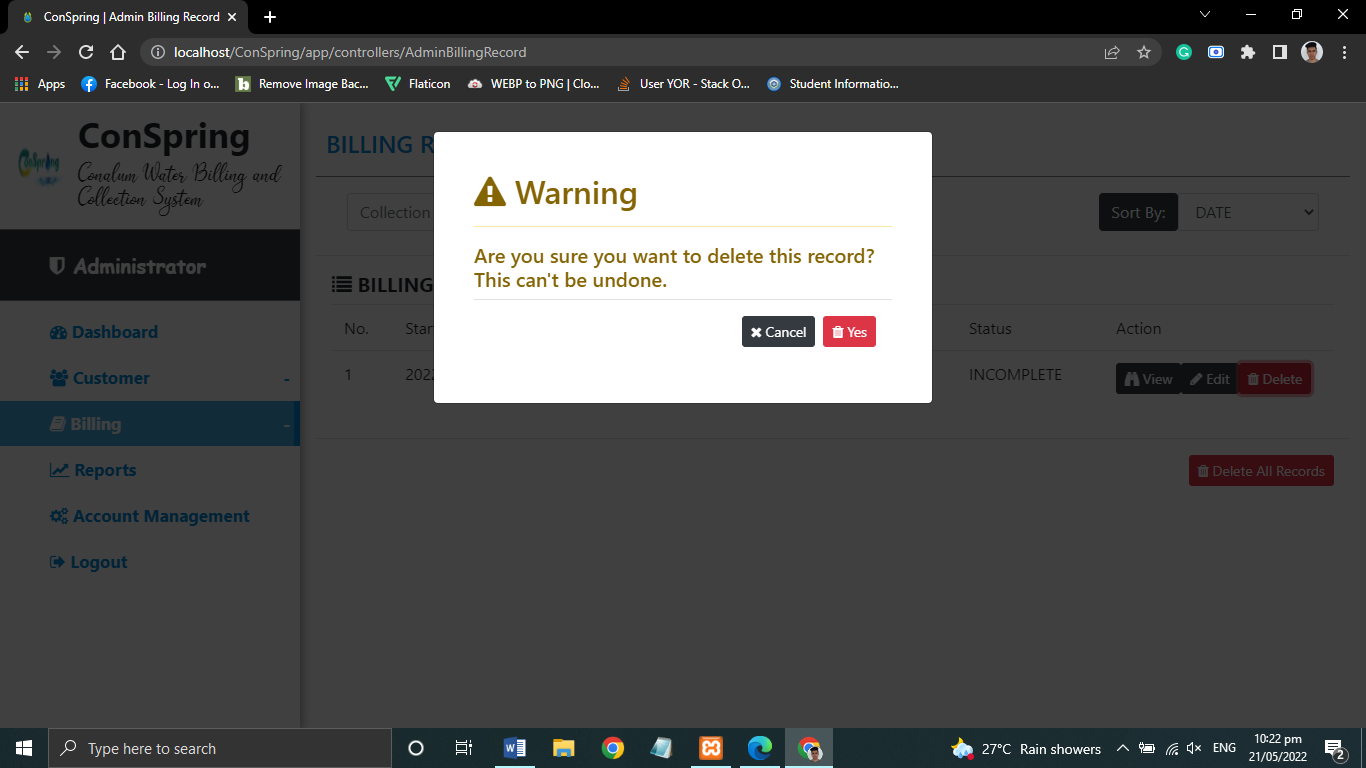
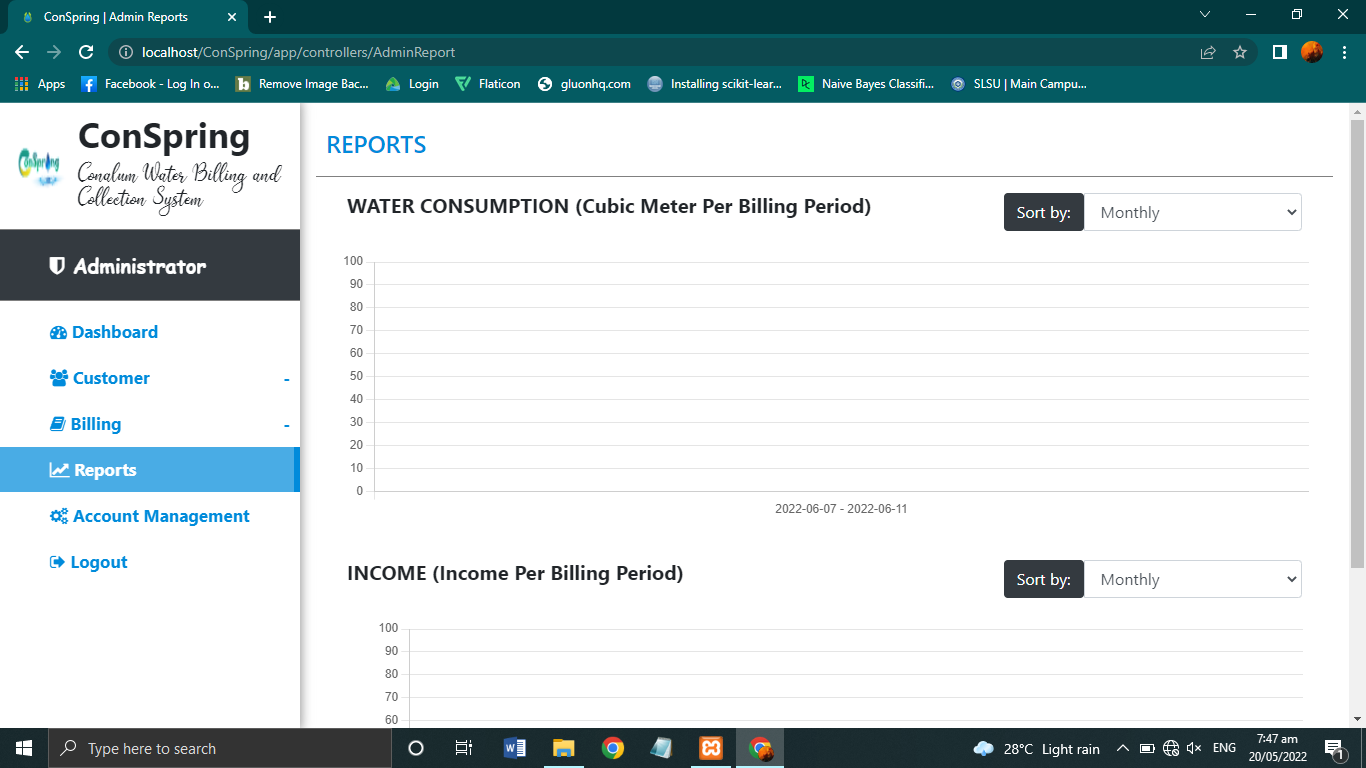


Figure 26. Billing Records – Delete

A warning dialog box will appear to confirm whether the user is sure to delete the billing record.

Figure 27.Reports

Displays Water Consumption and Income details in a graph. User can also sort the data by type.

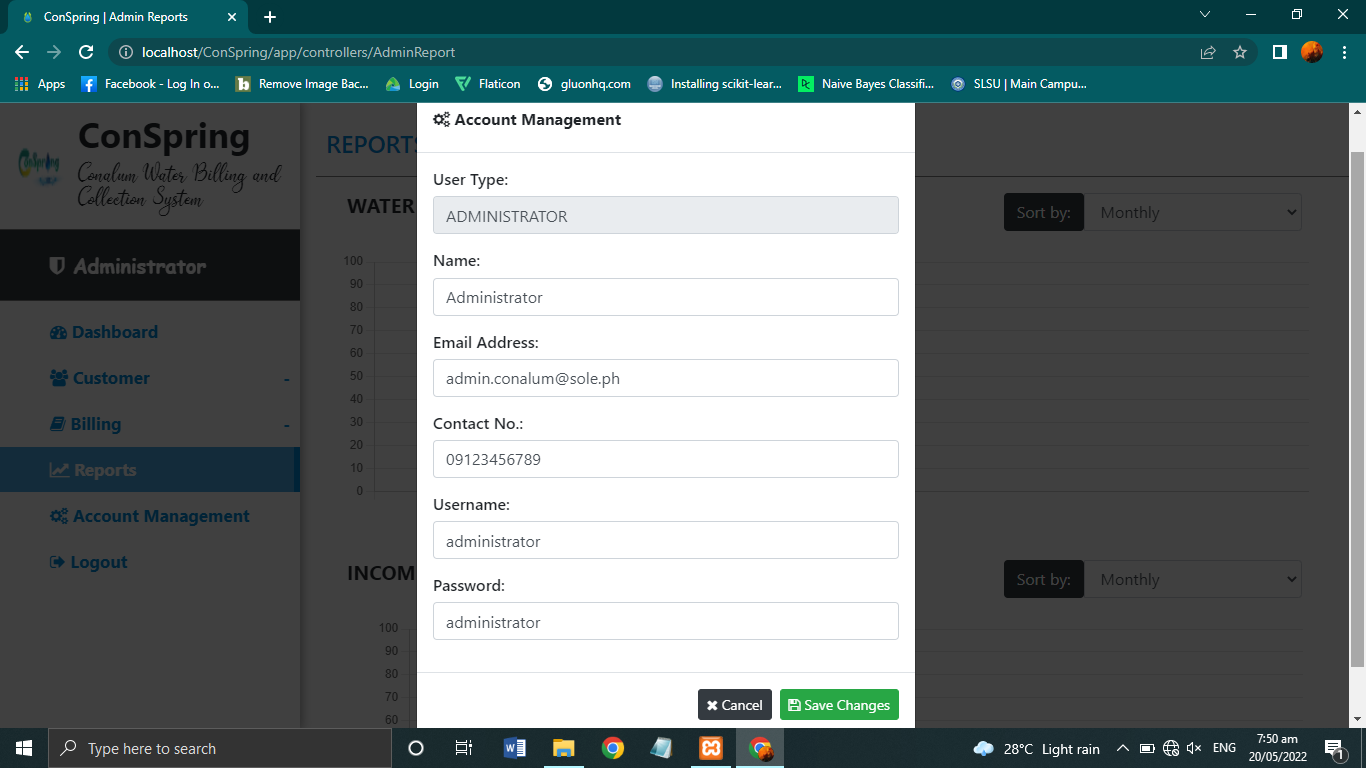


Figure 28. Account Management

Lets user edit User Information and save it.

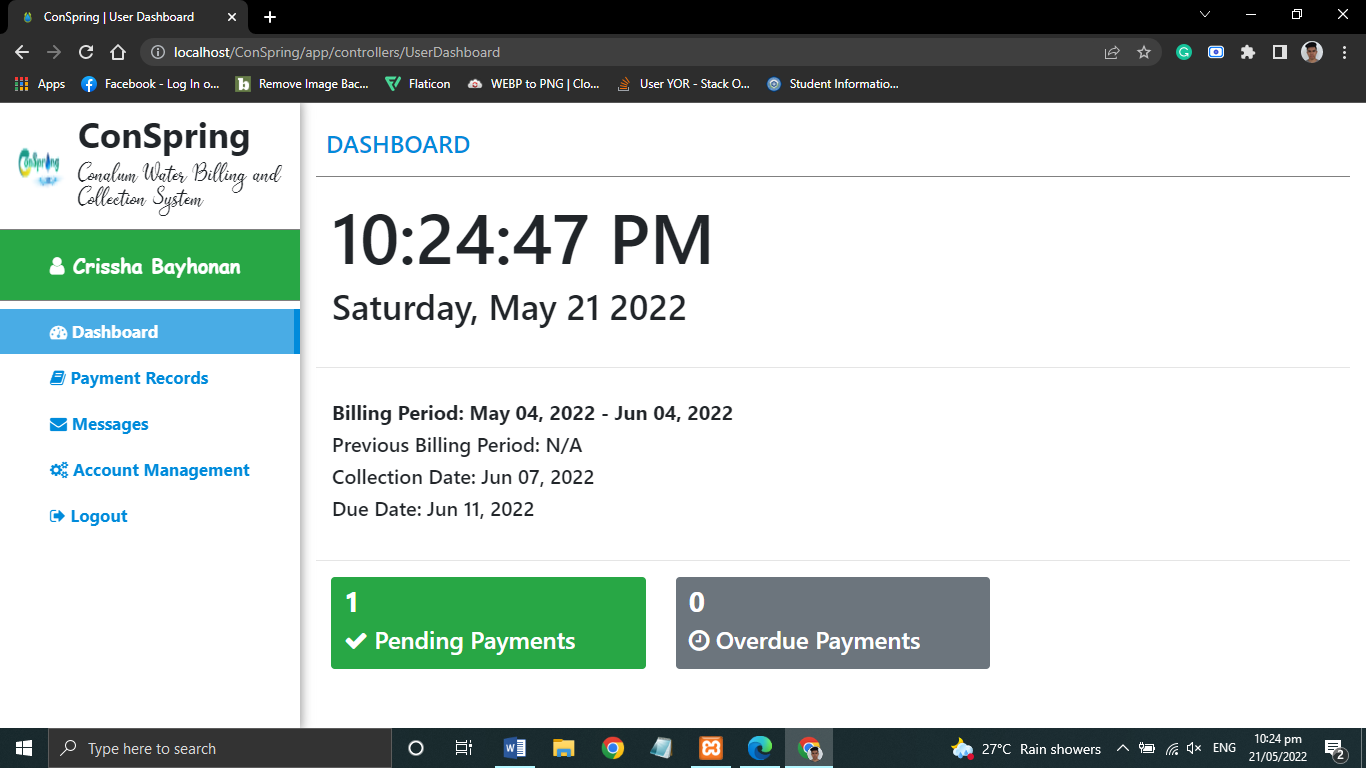
**Customer Mode**

Figure 29. Customer Dashboard

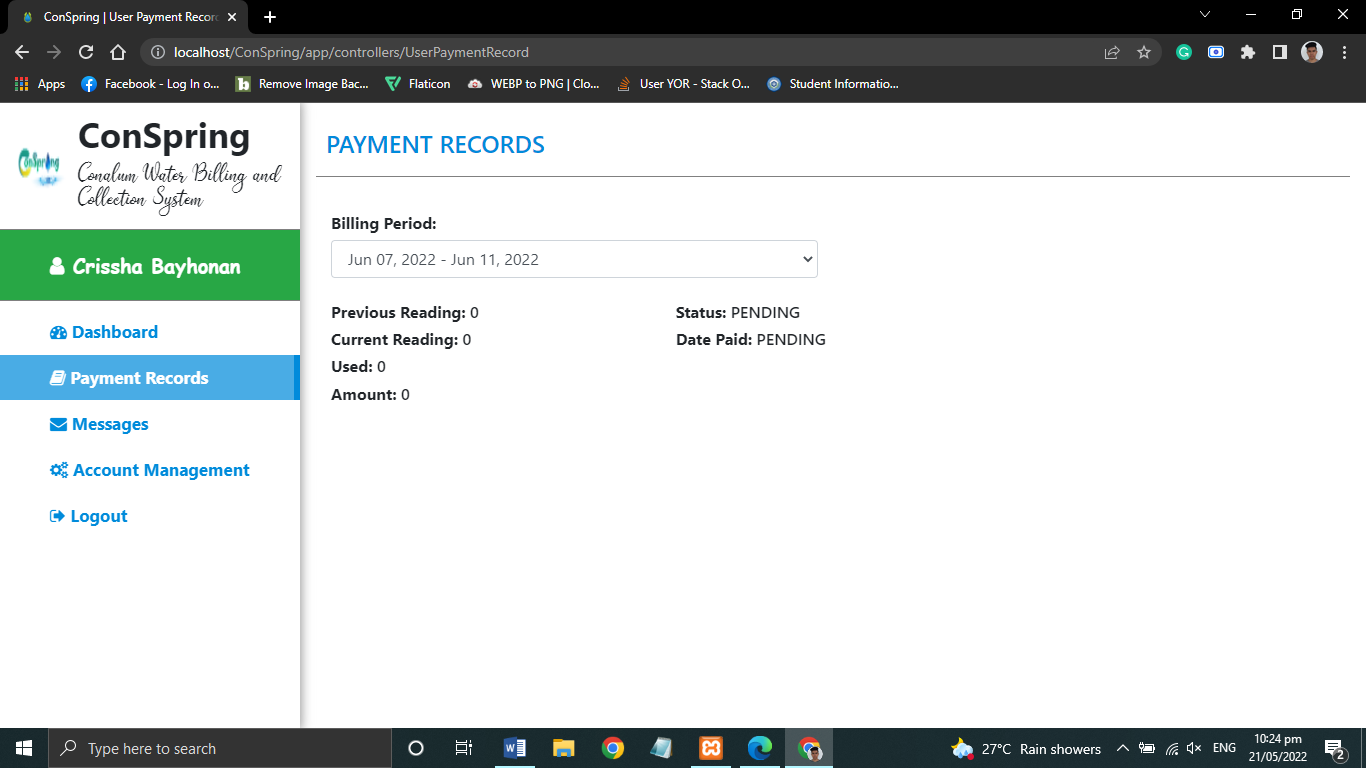
After logging in, the Dashboard, Payment Records, Messages, Account Management, and Logout tabs can be seen on the Left Side. The Dashboard displays time, date, and other important information about the customer.

Figure 30. Customer Payment Records

Displays the billing period and other details like the status of the customer’s payment and meter readings.

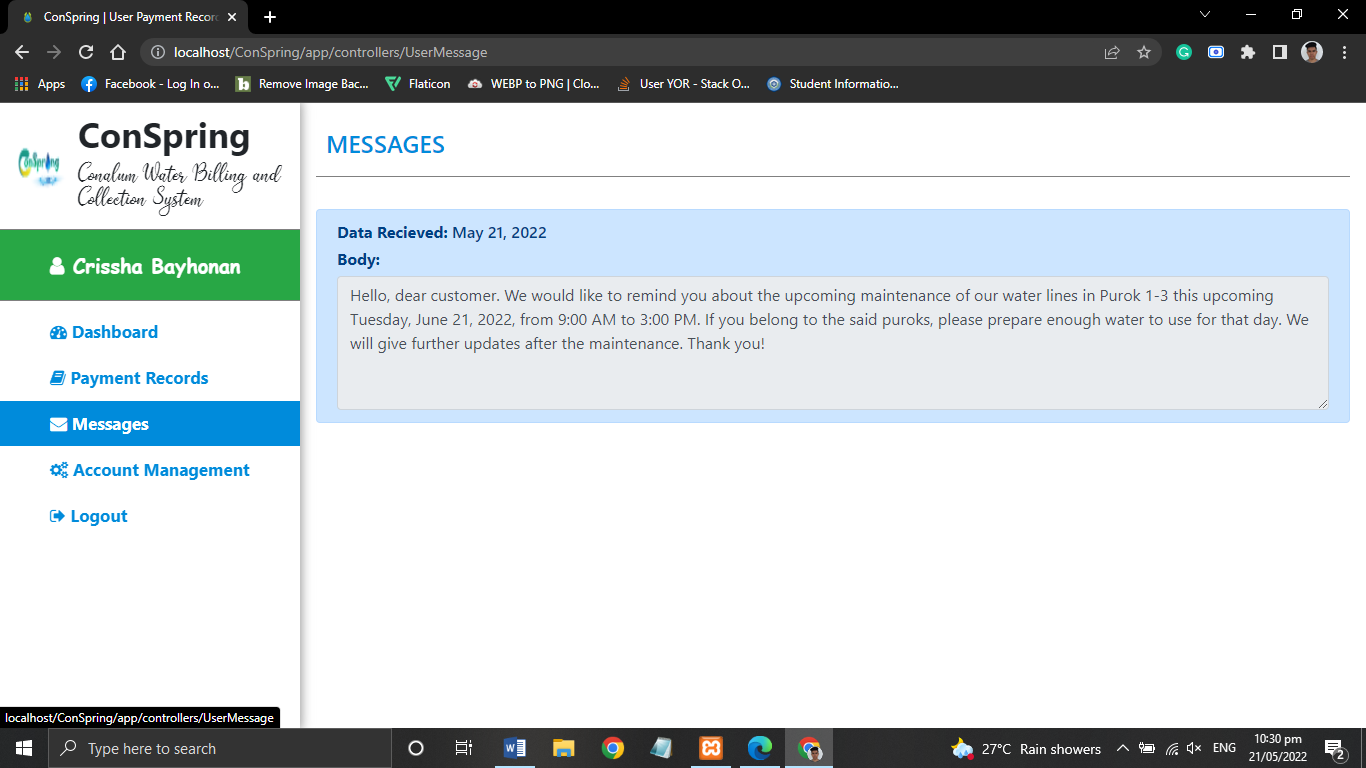


Figure 31. Customer Messages

Displays all the messages sent by the admin or treasurer to the customer.

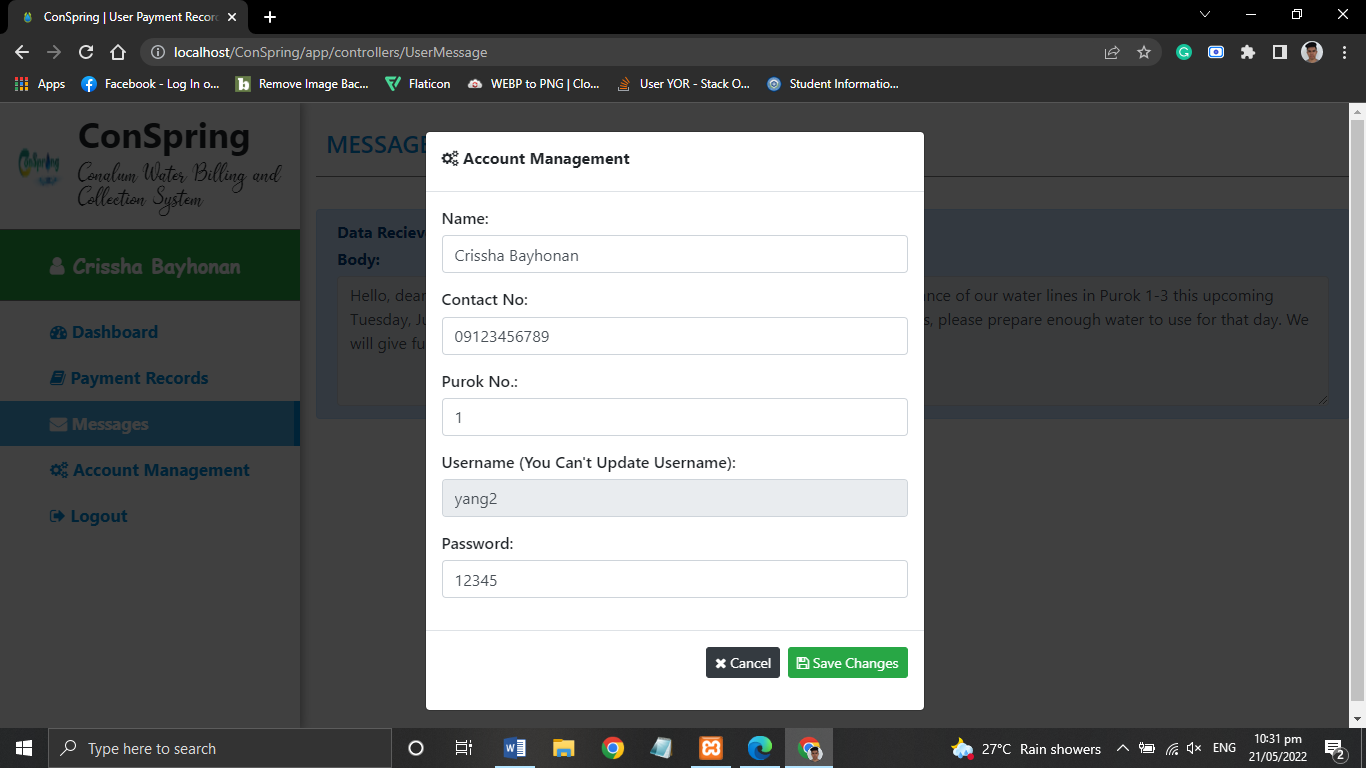


Figure 32. Customer Account Management

Lets the customer change his/her account password.

**4.4 DEVELOPMENT**

**4.4.1 SOFTWARE SPECIFICATION**

* Visual Studio Code
* Notepad
* Xampp
* Google Chrome, Internet Explorer and other search engines

**4.4.2 HARDWARE SPECIFICATION**

Hardware Used:

* PC (Personal Computer) and Laptops With at least intel core i3 up processor

2 GB DDR3 Memory

Windows 8 or 10 versions

With a strong internet connection

* Mobile Phones / Tablets

Android version not lowering from 5.0

Has built in browsers that can connect to the internet

Accepts cookies and caches

**4.4.3 PROGRAM SPECIFICATION**

* This system will be able to keep records of all the households that are connected to the barangay’s water system.
* The admins who manage the database monitors the status of the billing periods and the payment of the customers.
* The system will be utilized by the water tender and the barangay treasurer.
* The admins also control the computation and issuing of water bills to the customers.

**4.4.4 PROGRAMMING ENVIRONMENT**

**FRONT END**

* + Website Homepage
  + HTML with CSS for design.

**BACK END**

* + PHP/Laravel
  + JavaScript

**4.4.5 DEPLOYMENT PLAN**

Table 4. Deployment Plan

|  |  |  |
| --- | --- | --- |
| **SITE READINESS** | **YES** | **NO** |
| Are adequate resources available? | **/** |  |
| Does personnel receive training? | **/** |  |
| Are all agreements and contract signed off? | **/** |  |

This table shows the status of the system’s deployment plan.

**4.4.6 TEST PLAN**

Table 5. Test Plan

|  |  |  |
| --- | --- | --- |
| **TYPE OF TESTING** | **ACTION PLAN** | **DURATION** |
| Load Testing | The website will undergo a speed testing. | Within April 2022 |
| Performance Testing | Test all the capacities of the website. | Within May 2022 |
| Compatibility Testing | The website will be browsed into different platforms and test if the interface will stay the same. | Within May 2022 |
| System Testing | Test everything inside the website. | Within May 2022 |

This table shows the status of the system’s testing plans and the desired schedule.

**CONCLUSION**

In conclusion, based on the functionality of the deliverable components and design of our water billing system, it is possible that this system provides a more efficient, more reliable, faster, and safer way of collecting and storing necessary information about the barangay’s water billing records and the customer’s important details. It is recommended that the system be implemented on Barangay Conalum, Inopacan, Leyte, which gives a more effective transmission of data that will benefit both the water tender, the barangay treasurer, and all the households in the barangay that have water connection. Keeping track of the records will be a lot easier too, avoiding unnecessary data loss because of the limitations of the manual way before. Generating reports can be seen and printed in less than a minute and the overall storing of all records will be much simpler and safer.

**RECOMMENDATIONS**

The advantage of this proposed system is mainly focused on the increased efficiency and safer keeping of records. In future studies, customers should be able to receive more detailed updates about their balances and water consumption, including updates about price changes. Based on the results of the testing, the team recommends the following:

* Admins should confirm existing households that are connected to the barangay’s water connection and inform customers about their account information.
* The water tender and the treasurer should work hand in hand in managing the system to save time, reduce manpower, and effectively monitor the records.

**IMPLEMENTATION PLAN**

**Project Implementation Checklist**

Table 6. Project Implementation Checklist

|  |  |  |  |
| --- | --- | --- | --- |
| **Statement Plan** | **Person Incharge** | **Month** | **Expected Output** |
| 1. Turn over to CCSIT. | Project Manager | May – June 2022 | The system should be given to CCSIT to be used by the department. |
| 2. Deploy the system. | Project Manager | May – June 2022 | The system should run smoothly. |
| 3. Training and testing to the OJT coordinator and trainee. | Programmer, System Analyst | May – June 2022 | OJT student and coordinator will be aware of how the system works. |
| 4. Incubation | Tester | May – June 2022 | Any minor problems should be observed. |
| 5. Feedbacking minor | Writer | May – June 2022 | Minor problems should be observed. |
| 6. Maintenance of the System | Programmer, Tester | May – June 2022 | Update of the system for minor problems. |

This table shows the planned project implementation details.

**Implementation Contingency**

Table 7. Implementation Contingency Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Risks** | **Response** | **Contingency Plan** | **Responsible Person** |
| Web application might have problem | Weekly maintenance of the system. | Fix the errors and other possible problems. | Programmer, System Analyst, Tester |
| Browsers installed in the device is not updated and not compatible. | Use the latest version of the browser. | Update browser in use or use other browsers. | Programmer, Tester |
| Internet connection problem | Make sure there is enough data and there is strong signal. | Buy load or check internet connectivity issues. | System Analyst, Tester |
| Bill calculation might be wrong | Always check for changes of price from the authorities. | Check for sudden updates of prices or check for system errors. | Programmer, System Analyst, Tester |

This table shows the implementation contingency details of the system.

**Infrastructure/Deployment**

Table 8. Infrastructure/Deployment Table

|  |  |  |
| --- | --- | --- |
| **Task** | **Responsible Person** | **Task Description** |
| System Data Backup | Crissha Bayhonan | Saving system data in a flash drive for backup. |
| System Testing | Arce Tafalla | Regular testing of the functionality of the system after minor and major updates. |
| Monitoring of System | Jexter Jay Abrantes | Checking of problems and other issues that may concern the overall condition of the system. |
| Security Maintenance | Johann Von Sarong | Checking of spyware or other security issues like data breach. |

This table shows the tasks of the members of the team to ensure no problems in the system’s deployment.

**BIBLIOGRAPHY**

[1] http://mics.org/advantages-of-automated-billing-software/

[2] Critical Analysis of Billing System. (2016, Sep 13). Retrieved from <https://studymoose.com/critical-analysis-of-billing-system-essay>

[3] https://www.awwa.org/Portals/0/files/publications/documents/M1Ed7LookInside.pdf

[4] Ambre, J (2016). ASIST Automated Water Billing System. International Journal of Research in Management & Business Studies (IJRMBS 2016). Vol. 3 Issue 1 Jan. - Mar. 2016 Retrieved from http://ijrmbs.com > janelyn.pdf

[5] <http://jhelai4b.blogspot.com/2007/11/lan-based-assessment-and-billing-system.html>

[6] Manun-og, M., Manun-og, M., Claridad, N., Tereso, R., & Libarios, J. (2018). AUTOMATED WATER BILLING SYSTEM OF HINUNANGAN MUNICIPALITY. Innovative Technology and Management Journal, 1(1). Retrieved from [https://journal.evsu.edu.ph/index.php/itmj/article/view/41](https://journal.evsu.edu.ph/index.php/itmj/article/view/41?fbclid=IwAR2YK_hPzUX3vSv8w0TtU-coOvhX8gdttCizAQa_NvKhoJIdv6IR2fbseSs)

[7] <https://www.academia.edu/28274021/WATER_BILLING_SYSTEM>

[8] https://studylib.net/doc/9284799/2.1.1-computerized-water-billing-system

[9] Agrawal, P. & Shukla S. (2005 & 2006). Developing Effective Billing and Collection Practices. Retrieved from [http://documents1.worldbank.org](https://l.facebook.com/l.php?u=http://documents1.worldbank.org/?fbclid=IwAR2LvSkM1gvngXFSryaLkvDO_bdVPFXrPopdWdnuKqyeDcYM1xrqjqTTc50&h=AT1iPnw2cEeQJzNcfeC9AAuKC1qV5Z_23bVBWSJU2cG_ojVau6ROQ4Dhz2GjeEY12yjUjri6WxoVQOVRl8w6rUElNnqQCExmnk31Y-6rL0r5LUdqpW7XbRpPo9xnBfx66kLvlw) > 441190WSP0IN0P1ive0billing01PUBLIC1.pdf

[10] Bello, Z. (2014). DESIGN AND IMPLEMENTATION OF SMS BASED WATER BILLING SYSTEM (A CASE STUDY OF BAGUDO LOCAL GOVERNMENT WATER BOARD, KEBBI STATE). Retrieved from https://www.academia.edu/37775337/DESIGN\_AND\_IMPLEMENTATION\_OF\_SMS\_BASED\_WATER\_BILLING\_SYSTEM\_A\_CASE\_STUDY\_OF\_BAGUDO\_LOCAL\_GOVERNMENT\_WATER\_BOARD\_KEBBI\_STATE

[11] <https://www.irjet.net/archives/V5/i8/IRJET-V5I8169.pdf>

[12] "AUTOMATIC WATER BILLING SYSTEM USING ARDUINO", International Journal of Emerging Technologies and Innovative Research (www.jetir.org | UGC and issn Approved), ISSN:2349-5162, Vol.7, Issue 3, page no. pp1146-1149, March-2020, Available at : <http://www.jetir.org/papers/JETIR2003170.pdf>

**APPENDICES**

**Appendix A**

**RELEVANT SOURCE CODE**

**Admin Billing Code**

<?php

    include('../../unite/invoker/invoke.php');

    class AdminBilling{

        public static function index(){

            //code here...

        }

        public static function store(Request $request){

            $billingperiod = new BillingPeriod;

            $billingperiod->start = $request->start;

            $billingperiod->end = $request->end;

            $billingperiod->collection = $request->collection;

            $billingperiod->due = $request->due;

            DB::save($billingperiod);

            if(count(DB::all($billingperiod))){

                $bpid = DB::all($billingperiod,'id','desc')[0]["id"];

                $customeraccount = new CustomerAccount;

                foreach(DB::all($customeraccount) as $ca){

                    $caid = $ca["id"];

                    $billingrecord = new BillingRecord;

                    $billingrecord->caid = $caid;

                    $billingrecord->bpid = $bpid;

                    $billingrecord->preading = 0;

                    $billingrecord->creading = 0;

                    $billingrecord->used = 0;

                    $billingrecord->amount = 0;

                    $billingrecord->status = "PENDING";

                    $billingrecord->datepaid = "PENDING";

                    DB::save($billingrecord);

                }

            }

            Data::load("sole-message",["true","success","Added Successfully"]);

            Route::index(Data::unload("admin-location"));

        }

    }

?>

**Admin Billing Period**

<?php

    include('../../unite/invoker/invoke.api.php');

    class AdminBillingPeriod{

        public static function index(){

            //code here...

        }

        public static function store(){

            //code here...

        }

        public static function show(Request $request){

            $billingrecord = new BillingRecord;

            $billingrecords = DB::find($billingrecord,$request->id);

            if($billingrecords[0]['datepaid'] != "PENDING"){

                $billingrecords[0]['datepaid'] = date('Y-m-d',strtotime($billingrecords[0]['datepaid']));

            }

            Data::json\_response($billingrecords);

        }

    }

?>

**Admin Billing Record**

<?php

    include('../../unite/invoker/invoke.php');

    class AdminBillingRecord{

        public static function index(){

            if(Data::unload("admin-auth")){

                Data::load("admin-location","AdminBillingRecord");

                $billingperiod = new BillingPeriod;

                $billingperiods = DB::all($billingperiod,'id','desc');

                Route::view("administrator.billing-record",compact('billingperiods'));

            }else{

                Route::index("Login");

            }

        }

        public static function show(Request $request){

            $billingperiod = new BillingPeriod;

            $billingperiods = DB::find($billingperiod,$request->id);

            $billingrecord = new BillingRecord;

            $billingrecords = DB::where($billingrecord,'bpid','=',$request->id);

            $customeraccount = new CustomerAccount;

            $customeraccounts = DB::all($customeraccount);

            Route::view("administrator.billing-view",compact('billingperiods','billingrecords','customeraccounts'));

        }

        public static function update(Request $request){

            $billingperiod = new BillingPeriod;

            $billingperiodtemp = DB::prepare($billingperiod,$request->id);

            $billingperiodtemp->start = $request->start;

            $billingperiodtemp->end = $request->end;

            $billingperiodtemp->collection = $request->collection;

            $billingperiodtemp->due = $request->due;

            DB::update($billingperiodtemp);

            Data::load("sole-message",["true","success","Billing Period Updated"]);

            Route::index("AdminBillingRecord");

        }

        public static function destroy(Request $request){

            if($request->type == "solo"){

                $billingrecord = new BillingRecord;

                foreach(DB::where($billingrecord,'bpid','=',$request->id) as $br){

                    DB::delete($billingrecord,$br['id']);

                }

                $billingperiod = new BillingPeriod;

                DB::delete($billingperiod,$request->id);

            }else{

                $billingrecord = new BillingRecord;

                DB::wipe($billingrecord);

                $billingperiod = new BillingPeriod;

                DB::wipe($billingperiod);

            }

            Data::load("sole-message",["true","success","Record Deleted"]);

            Route::index("AdminBillingRecord");

        }

    }

?>

**Admin Billing Record Search**

<?php

    include('../../unite/invoker/invoke.api.php');

    class AdminBillingRecordSearch{

        public static function show(Request $request){

            $billingperiod = new BillingPeriod;

            $billingperiods = DB::where($billingperiod,'collection','like',$request->date,'asc');

            for ($i=0; $i < count($billingperiods); $i++) {

                $billingperiods[$i]["status"] = "COMPLETE";

                $billingrecord = new BillingRecord;

                foreach(DB::where($billingrecord,'bpid','=',$billingperiods[$i]["id"]) as $br){

                    if($br['status'] == 'PENDING' || $br['status'] == 'OVERDUE'){

                        $billingperiods[$i]["status"] = "INCOMPLETE";

                    }

                }

            }

            Data::json\_response($billingperiods);

        }

    }

?>

**Admin Billing Record Sort**

<?php

    include('../../unite/invoker/invoke.api.php');

    class AdminBillingRecordSort{

        public static function show(Request $request){

            $billingperiod = new BillingPeriod;

            if($request->sortby == "start"){

                $billingperiods = DB::all($billingperiod,'id','asc');

                for ($i=0; $i < count($billingperiods); $i++) {

                    $billingperiods[$i]["status"] = "COMPLETE";

                    $billingrecord = new BillingRecord;

                    foreach(DB::where($billingrecord,'bpid','=',$billingperiods[$i]["id"]) as $br){

                        if($br['status'] == 'PENDING' || $br['status'] == 'OVERDUE'){

                            $billingperiods[$i]["status"] = "INCOMPLETE";

                        }

                    }

                }

                Data::json\_response($billingperiods);

            }else{

                $billingperiods = DB::all($billingperiod,'id','asc');

                for ($i=0; $i < count($billingperiods); $i++) {

                    $billingperiods[$i]["status"] = "COMPLETE";

                    $billingrecord = new BillingRecord;

                    foreach(DB::where($billingrecord,'bpid','=',$billingperiods[$i]["id"]) as $br){

                        if($br['status'] == 'PENDING' || $br['status'] == 'OVERDUE'){

                            $billingperiods[$i]["status"] = "INCOMPLETE";

                        }

                    }

                }

                $billingperiodtemp = [];

                foreach($billingperiods as $bp){

                    if($bp['status'] == "COMPLETE"){

                        array\_push($billingperiodtemp,$bp);

                    }else{

                        array\_unshift($billingperiodtemp,$bp);

                    }

                }

                Data::json\_response($billingperiodtemp);

            }

        }

    }

?>

**Admin Customer List**

    <?php

        include('../../unite/invoker/invoke.php');

        class AdminCustomerList{

            public static function index(){

                if(Data::unload("admin-auth")){

                    Data::load("admin-location","AdminCustomerList");

                    $customeraccount = new CustomerAccount;

                    $customeraccounts = DB::all($customeraccount,Data::unload("admin-customer-sort"),"desc");

                    Route::view("administrator.customer-list",compact('customeraccounts'));

                }else{

                    Route::index("Login");

                }

            }

            public static function store(Request $request){

                $customeraccount = new CustomerAccount;

                if(DB::validate($customeraccount,"username",$request->username)){

                    $customeraccount->name = $request->name;

                    $customeraccount->contactno = $request->contactno;

                    $customeraccount->purokno = $request->purokno;

                    $customeraccount->dateconnected = $request->connectiondate;

                    $customeraccount->username = $request->username;

                    $customeraccount->password = $request->password;

                    DB::save($customeraccount);

                    $billingperiod = new BillingPeriod;

                    if(count(DB::all($billingperiod))){

                        $billingrecord = new BillingRecord;

                        $billingrecord->bpid = DB::all($billingperiod,'id','desc')[0]["id"];

                        $billingrecord->caid = DB::where($customeraccount,'username','=',$request->username)[0]['id'];

                        $billingrecord->preading = 0;

                        $billingrecord->creading = 0;

                        $billingrecord->used = 0;

                        $billingrecord->amount = 0;

                        $billingrecord->status = "PENDING";

                        $billingrecord->datepaid = "PENDING";

                        DB::save($billingrecord);

                    }

                    Data::load("sole-message",["true","success","Customer Added"]);

                }else{

                    Data::load("sole-message",["true","warning","Username Already Existed"]);

                }

                Route::index(Data::unload("admin-location"));

            }

            public static function show(Request $request){

                $customeraccount = new CustomerAccount;

                $customeraccounts = DB::find($customeraccount,$request->id);

                $billingperiod = new BillingPeriod;

                $billingperiods = DB::all($billingperiod);

                $billingrecord = new BillingRecord;

                $billingrecords = DB::where($billingrecord,"caid","=",$request->id,'id','desc');

                Route::view("administrator.customer-view",compact('customeraccounts','billingperiods','billingrecords'));

            }

            public static function update(Request $request){

                $update = false;

                $customeraccount = new CustomerAccount;

                if(DB::find($customeraccount,$request->id)[0]['username'] == $request->username){

                    $update = true;

                }else{

                    if(DB::validate($customeraccount,"username",$request->username)){

                        $update = true;

                    }

                }

                if($update){

                    $customeraccounttemp = DB::prepare($customeraccount,$request->id);

                    $customeraccounttemp->name = $request->name;

                    $customeraccounttemp->contactno = $request->contactno;

                    $customeraccounttemp->purokno = $request->purokno;

                    $customeraccounttemp->dateconnected = $request->dateconnected;

                    $customeraccounttemp->username = $request->username;

                    $customeraccounttemp->password = $request->password;

                    DB::update($customeraccounttemp);

                    if($request->billingupdate == "true"){

                        $billingrecord = new BillingRecord;

                        $billingrecordtemp = DB::prepare($billingrecord,$request->brid);

                        $billingrecordtemp->preading = $request->preading;

                        $billingrecordtemp->creading = $request->creading;

                        $billingrecordtemp->used = $request->used;

                        $billingrecordtemp->amount = $request->amount;

                        $billingrecordtemp->status = $request->status;

                        $billingrecordtemp->datepaid = $request->datepaid;

                        DB::update($billingrecordtemp);

                    }

                }else{

                    echo "username exist";

                }

                Data::load("sole-message",["true","success","Record Updated"]);

                header("location: AdminCustomerList?handler&id=".$request->id);

                //Route::index(Data::unload("admin-location"));

            }

            public static function destroy(Request $request){

                if($request->type == "solo"){

                    $billingrecord = new BillingRecord;

                    foreach(DB::where($billingrecord,'caid','=',$request->id) as $br){

                        DB::delete($billingrecord,$br['id']);

                    }

                    $customeraccount = new CustomerAccount;

                    DB::delete($customeraccount,$request->id);

                }else{

                    $customeraccount = new CustomerAccount;

                    DB::wipe($customeraccount);

                    $billingrecord = new BillingRecord;

                    DB::wipe($billingrecord);

                }

                Data::load("sole-message",["true","success","Record Deleted"]);

                Route::index("AdminCustomerList");

            }

            public static function handler(Request $request){

                $customeraccount = new CustomerAccount;

                $customeraccounts = DB::find($customeraccount,$request->id);

                $billingperiod = new BillingPeriod;

                $billingperiods = DB::all($billingperiod);

                $billingrecord = new BillingRecord;

                $billingrecords = DB::where($billingrecord,"caid","=",$request->id,'id','desc');

                Route::view("administrator.customer-edit",compact('customeraccounts','billingperiods','billingrecords'));

            }

        }

    ?>

**Admin Customer Search**

<?php

    include('../../unite/invoker/invoke.api.php');

    class AdminCustomerSearch{

        public static function index(){

            //code here...

        }

        public static function store(){

            //code here...

        }

        public static function show(Request $request){

            $customeraccount = new CustomerAccount;

            $customeraccounts = DB::where($customeraccount,'name','like',$request->name,Data::unload("admin-customer-sort"),"asc");

            Data::json\_response($customeraccounts);

        }

    }

?>

**Admin Customer Sort**

<?php

    include('../../unite/invoker/invoke.api.php');

    class AdminCustomerSort{

        public static function show(Request $request){

            Data::load("admin-customer-sort",$request->sortby);

            $customeraccount = new CustomerAccount;

            $customeraccounts = DB::all($customeraccount,Data::unload("admin-customer-sort"),"desc");

            Data::json\_response($customeraccounts);

        }

    }

?>

**Admin Dashboard**

<?php

    include('../../unite/invoker/invoke.php');

    class AdminDashboard{

        public static function index(){

            if(Data::unload("admin-auth")){

                Data::load("admin-location","AdminDashboard");

                $billingperiod = new BillingPeriod;

                $billingperiods = DB::all($billingperiod,'id','desc');

                $billingrecord = new BillingRecord;

                $paid = count(DB::where($billingrecord,'status','=','PAID'));

                $pending = count(DB::where($billingrecord,'status','=','PENDING'));

                $overdue = count(DB::where($billingrecord,'status','=','OVERDUE'));

                $statistic = array($paid,$pending,$overdue);

                Route::view("administrator.dashboard",compact('billingperiods','statistic'));

            }else{

                Route::index("Login");

            }

        }

    }

?>

**Admin Log-In**

<?php

    include('../../unite/invoker/invoke.php');

    class AdminLogin{

        public static function index(){

            if(!Data::unload("admin-auth")){

                Data::load("admin-location","AdminLogin");

                Route::view("authentication.admin-login");

            }else{

                Route::index(Data::unload("admin-location"));

            }

        }

        public static function store(){

            //code here...

        }

        public static function show(Request $request){

            $adminaccount = new AdminAccount;

            if(Data::unload("user-type") == "administrator"){

                $admin = DB::find($adminaccount,1);

                if($admin[0]['username'] == $request->username && $admin[0]['password'] == $request->password){

                    Data::load("sole-message",["true","info","Welcome Admin!"]);

                    Data::load("admin-auth",true);

                    Data::load("admin",$admin);

                    Route::index("AdminDashboard");

                }else{

                    Data::load("sole-message",["true","warning","Incorrect Username & Password"]);

                    Route::index("AdminLogin");

                }

            }else{

                $admin = DB::find($adminaccount,2);

                if($admin[0]['username'] == $request->username && $admin[0]['password'] == $request->password){

                    Data::load("sole-message",["true","info","Welcome Treasurer!"]);

                    Data::load("admin-auth",true);

                    Data::load("admin",$admin);

                    Route::index("AdminDashboard",compact("admin"));

                }else{

                    Data::load("sole-message",["true","warning","Incorrect Username & Password"]);

                    Route::index("AdminLogin");

                }

            }

        }

        public static function update(Request $request){

            $adminaccount = new AdminAccount;

            $temp = DB::prepare($adminaccount,$request->id);

            $temp->name = $request->name;

            $temp->email = $request->email;

            $temp->contactno = $request->contactno;

            $temp->username = $request->username;

            $temp->password = $request->password;

            DB::update($temp);

            Data::load("admin",DB::find($adminaccount,$request->id));

            Data::load("sole-message",["true","success","Changes Saved!"]);

            Route::index(Data::unload("admin-location"));

        }

        public static function destroy(){

            Data::load("user-type","administrator");

            Data::load("admin-auth",false);

            Route::index("Login");

        }

        public static function handler(Request $request){

            if($request->type){

                Data::load("user-type","administrator");

            }else{

                Data::load("user-type","treasurer");

            }

            Route::index("AdminLogin");

        }

    }

?>

**Admin Report**

<?php

    include('../../unite/invoker/invoke.php');

    class AdminReport{

        public static function index(){

            if(Data::unload("admin-auth")){

                Data::load("admin-location","AdminReport");

                $billingrecord = new BillingRecord;

                $billingrecords = DB::all($billingrecord);

                $billingperiod = new BillingPeriod;

                $billingperiods = DB::all($billingperiod);

                Route::view('administrator.report',compact('billingrecords','billingperiods'));

            }else{

                Route::index("Login");

            }

        }

    }

?>

**Admin Send Message**

<?php

    include('../../unite/invoker/invoke.php');

    class AdminSendMessage{

        public static function index(){

            if(Data::unload("admin-auth")){

                Data::load("admin-location","AdminCustomerList");

                $customeraccount = new CustomerAccount;

                $customeraccounts = DB::all($customeraccount);

                Route::view("administrator.send-message",compact('customeraccounts'));

            }else{

                Route::index("Login");

            }

        }

        public static function store(Request $request){

            $customeraccount = new CustomerAccount;

            if($request->sendto == "all"){

                foreach(DB::all($customeraccount) as $ca){

                    $message = new Message;

                    $message->caid = $ca['id'];

                    $message->message = $request->message;

                    $message->status = 0;

                    DB::save($message);

                }

            }else{

                $message = new Message;

                $message->caid = DB::find($customeraccount,$request->caid)[0]['id'];

                $message->message = $request->message;

                $message->status = 0;

                DB::save($message);

            }

            Data::load("sole-message",["true","success","Message Sent"]);

            Route::index("AdminSendMessage");

        }

    }

?>

**Sole Controller**

<?php

    include('../../unite/invoker/invoke.php');

    class SoleController{

        public static function index(){

            Route::view("sole");

        }

    }

?>

**Login**

<?php

    include('../../unite/invoker/invoke.php');

    class UserDashboard{

        public static function index(){

            if(Data::unload("user-auth")){

                Data::load("user-location","UserDashboard");

                $billingperiod = new BillingPeriod;

                $billingperiods = DB::all($billingperiod,'id','desc');

                $billingrecord = new BillingRecord;

                $records = DB::where($billingrecord,'caid','=',Data::unload("user-id"));

                $pending = 0;

                $overdue = 0;

                foreach($records as $r){

                    if($r['status'] == "PENDING"){

                        $pending++;

                    }

                    if($r['status'] == "OVERDUE"){

                        $overdue++;

                    }

                }

                $statistic = array($pending,$overdue);

                $customeraccount = new CustomerAccount;

                $customer = DB::find($customeraccount,Data::unload("user-id"));

                Data::load("user-data",$customer);

                Route::view("public.dashboard",compact('billingperiods','statistic'));

            }else{

                Route::index("Login");

            }

        }

    }

?>

**User Dashboard**

<?php

    include('../../unite/invoker/invoke.php');

    class UserDashboard{

        public static function index(){

            if(Data::unload("user-auth")){

                Data::load("user-location","UserDashboard");

                $billingperiod = new BillingPeriod;

                $billingperiods = DB::all($billingperiod,'id','desc');

                $billingrecord = new BillingRecord;

                $records = DB::where($billingrecord,'caid','=',Data::unload("user-id"));

                $pending = 0;

                $overdue = 0;

                foreach($records as $r){

                    if($r['status'] == "PENDING"){

                        $pending++;

                    }

                    if($r['status'] == "OVERDUE"){

                        $overdue++;

                    }

                }

                $statistic = array($pending,$overdue);

                $customeraccount = new CustomerAccount;

                $customer = DB::find($customeraccount,Data::unload("user-id"));

                Data::load("user-data",$customer);

                Route::view("public.dashboard",compact('billingperiods','statistic'));

            }else{

                Route::index("Login");

            }

        }

    }

?>

**User Login**

<?php

    include('../../unite/invoker/invoke.php');

    class UserLogin{

        public static function index(){

            if(!Data::unload("user-auth")){

                Data::load("admin-location","UserLogin");

                Route::view("authentication.user-login");

            }else{

                Route::index(Data::unload("user-location"));

            }

        }        public static function show(Request $request){

            $customeraccount = new CustomerAccount;

            $customers = DB::where($customeraccount,"username","=",$request->username);

            if($customers){

                if($customers[0]['password'] == $request->password){

                    Data::load("user-auth",true);

                    Data::load("user-id",$customers[0]['id']);

                    Route::index("UserDashboard");

                }else{

                    Data::load("sole-message",["true","warning","Incorrect Username or Password"]);

                    Route::index("UserLogin");

                }

            }else{

                Data::load("sole-message",["true","warning","Incorrect Username or Password"]);

                Route::index("UserLogin");

            }

        }

        public static function update(Request $request){

            $customeraccount = new CustomerAccount;

            $customer = DB::prepare($customeraccount,$request->id);

            $customer->name = $request->name;

            $customer->contactno = $request->contactno;

            $customer->purokno = $request->purokno;

            $customer->password = $request->password;

            DB::update($customer);

            $customer = DB::find($customeraccount,Data::unload("user-id"));

            Data::load("user-data",$customer);

            Data::load("sole-message",["true","success","Changes Saved!"]);

            Route::index(Data::unload("user-location"));

        }

        public static function destroy(){

            Data::load("user-auth",false);

            Data::load("user-id","");

            Data::load("user-data","");

            Route::index("Login");

        }    }

?>

**User Message**

<?php

    include('../../unite/invoker/invoke.php');

    class UserMessage{

        public static function index(){

            if(Data::unload("user-auth")){

                Data::load("user-location","UserMessage");

                $message = new Message;

                $messages = Data::reverse(DB::where($message,"caid","=",Data::unload("user-id")));

                Route::view("public.message",compact('messages'));

            }else{

                Route::index("Login");

            }

        }

    }

?>

**User Payment Record**

<?php

    include('../../unite/invoker/invoke.php');

    class UserPaymentRecord{

        public static function index(){

            if(Data::unload("user-auth")){

                Data::load("user-location","UserPaymentRecord");

                $customeraccount = new CustomerAccount;

                $customeraccounts = DB::find($customeraccount,Data::unload("user-id"));

                $billingperiod = new BillingPeriod;

                $billingperiods = DB::all($billingperiod);

                $billingrecord = new BillingRecord;

                $billingrecords = DB::where($billingrecord,"caid","=",Data::unload("user-id"),'id','desc');

                Route::view("public.records",compact('customeraccounts','billingperiods','billingrecords'));

            }else{

                Route::index("Login");

            }

        }    }

?>

**Migrations**

<?php

    Migrate::$migration = ["CustomerAccountMigration","BillingPeriodMigration","BillingRecordMigration","MessageMigration","AdminAccountMigration"];

    class CustomerAccountMigration

    {

        public static function index(){

            Migrate::attrib\_table("customeraccount");

            Migrate::attrib\_string(255);

            Migrate::string("name");

            Migrate::string("contactno");

            Migrate::string("purokno");

            Migrate::string("dateconnected");

            Migrate::string("username");

            Migrate::string("password");

        }

    }

    class BillingPeriodMigration

    {

        public static function index(){

            Migrate::attrib\_table("billingperiod");

            Migrate::attrib\_string(255);

            Migrate::string("start");

            Migrate::string("end");

            Migrate::string("collection");

            Migrate::string("due");

        }

    }

    class BillingRecordMigration

    {

        public static function index(){

            Migrate::attrib\_table("billingrecord");

            Migrate::attrib\_string(255);

            Migrate::string("caid");

            Migrate::string("bpid");

            Migrate::string("preading");

            Migrate::string("creading");

            Migrate::string("used");

            Migrate::string("amount");

            Migrate::string("status");

            Migrate::string("datepaid");

        }

    }

    class MessageMigration

    {

        public static function index(){

            Migrate::attrib\_table("message");

            Migrate::attrib\_string(500);

            Migrate::string("caid");

            Migrate::string("message");

            Migrate::string("status");

        }

    }

    class AdminAccountMigration

    {

        public static function index(){

            Migrate::attrib\_table("adminaccount");

            Migrate::attrib\_string(255);

            Migrate::string("type");

            Migrate::string("name");

            Migrate::string("email");

            Migrate::string("contactno");

            Migrate::string("username");

            Migrate::string("password");

        }

    }

?>

**Models**

<?php

    $models = ["CustomerAccount","BillingPeriod","BillingRecord","Message","AdminAccount"];

    class CustomerAccount

    {

        public $table = "customeraccount";

        public $fillable = [

            "name",

            "contactno",

            "purokno",

            "dateconnected",

            "username",

            "password"

        ];

    }

    class BillingPeriod

    {

        public $table = "billingperiod";

        public $fillable = [

            "start",

            "end",

            "collection",

            "due"

        ];

    }

    class BillingRecord

    {

        public $table = "billingrecord";

        public $fillable = [

            "caid",

            "bpid",

            "preading",

            "creading",

            "used",

            "amount",

            "status",

            "datepaid"

        ];

    }

    class Message

    {

        public $table = "message";

        public $fillable = [

            "caid",

            "message",

            "status"

        ];

    }

    class AdminAccount

    {

        public $table = "adminaccount";

        public $fillable = [

            "type", "name", "email","contactno","username","password" ];}

?>

**Appendix B**

**EVALUATION TOOL**

**Instructions:** Please evaluate the “SLSU OJT Monitoring System using Face Recognition” using the scale shown below. Check (/) the appropriate score. Thank You.

**Jexter Jay A. Abrantes Rhoderick D. Malangsa**

**Programmer Adviser**

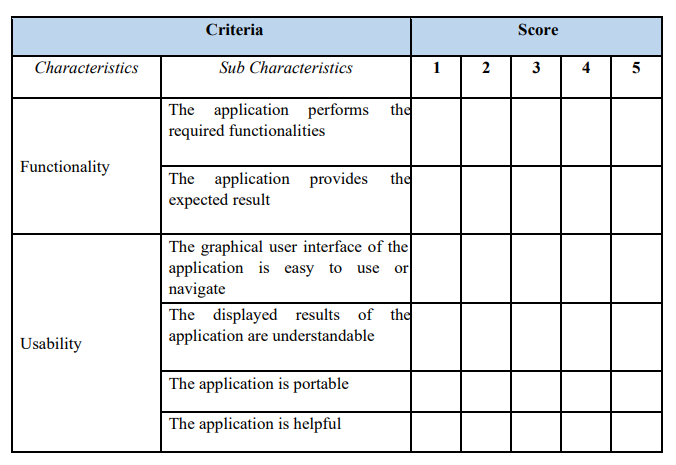


**Qualitative Description per Functionality Indicator**

|  |  |
| --- | --- |
| **Limits of Scale** | **Qualitative Description** |
| 4.21 – 5.00 | Fully Functional |
| 3.21 – 4.20 | Mostly Functional |
| 2.61 – 3.20 | Functional |
| 1.81 – 2.60 | Slightly Functional |
| 1.0 – 1.8 | Not Functional |

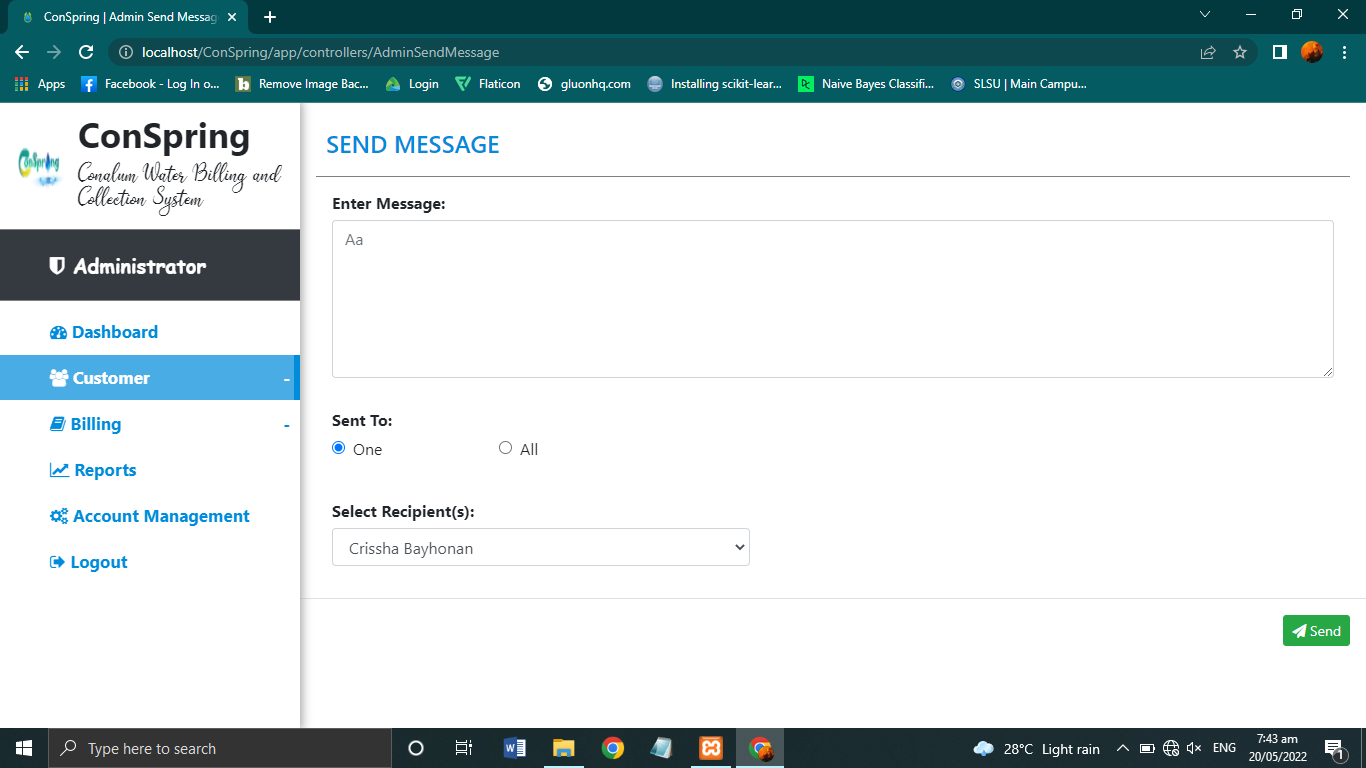
**Qualitative Description per Usability Indicator**

|  |  |
| --- | --- |
| **Limits of Scale** | **Qualitative Description** |
| 4.21 – 5.00 | Fully Usable |
| 3.21 – 4.20 | Mostly Usable |
| 2.61 – 3.20 | Usable |
| 1.81 – 2.60 | Slightly Usable |
| 1.0 – 1.8 | Not Usable |

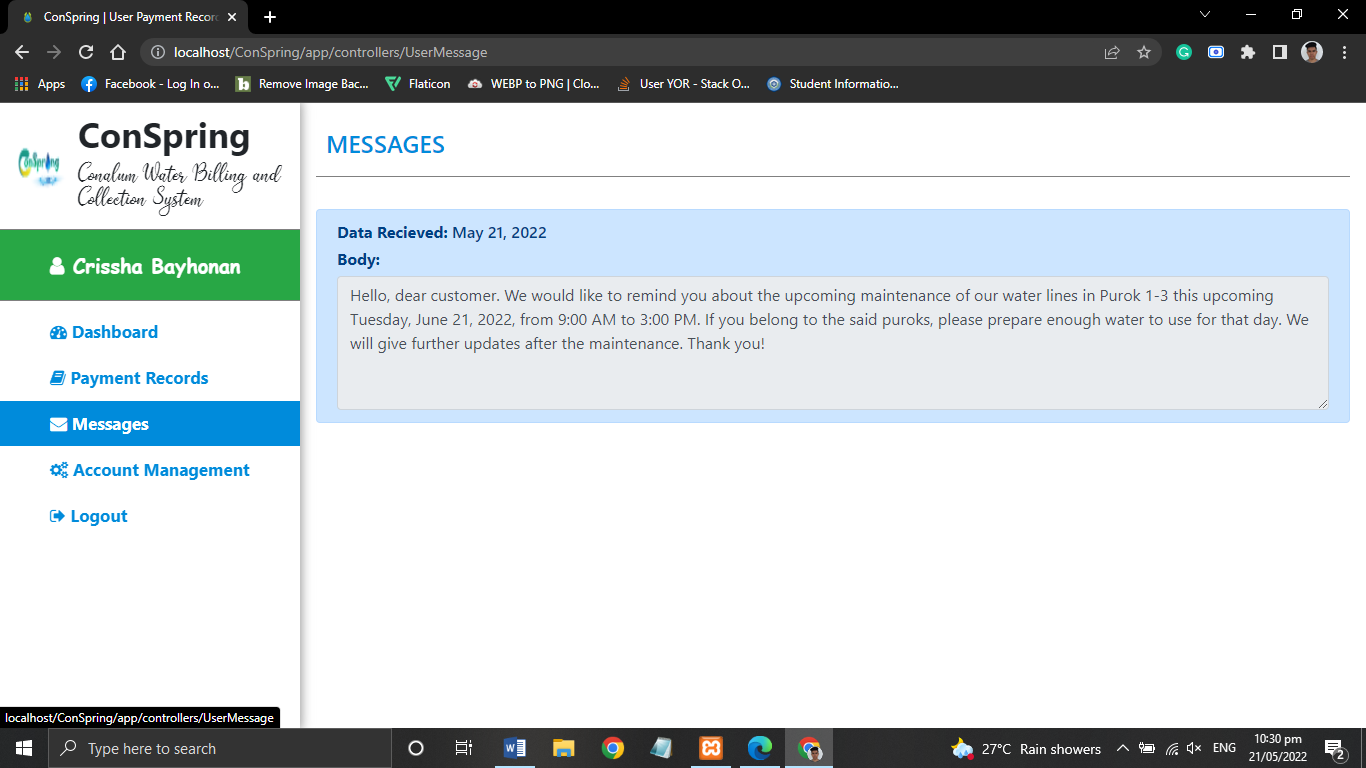


**Appendix C**

**SAMPLE INPUT AND OUTPUT**



**Input for Admin Sending Message to Customer**



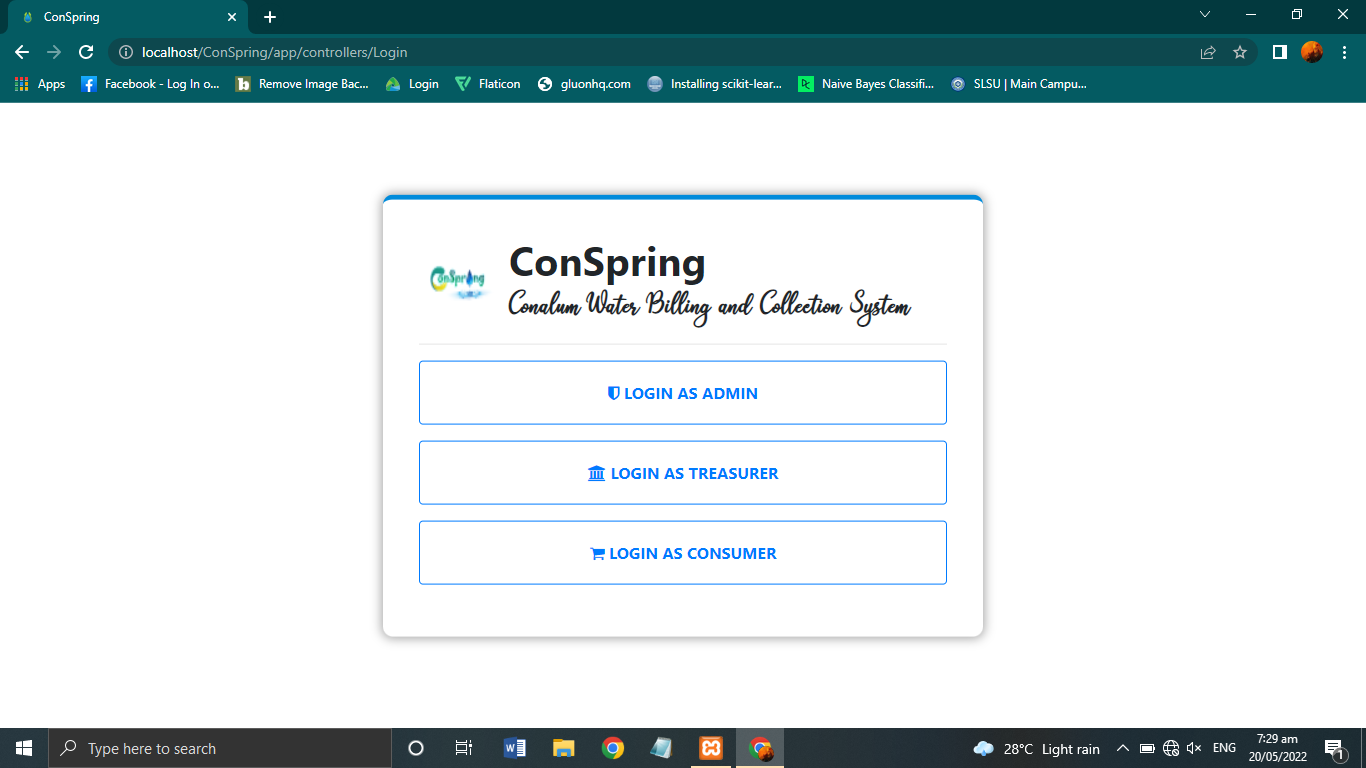
**Output for Admin Sending Message to Customer**

**Appendix D**

**USER’S GUIDE**

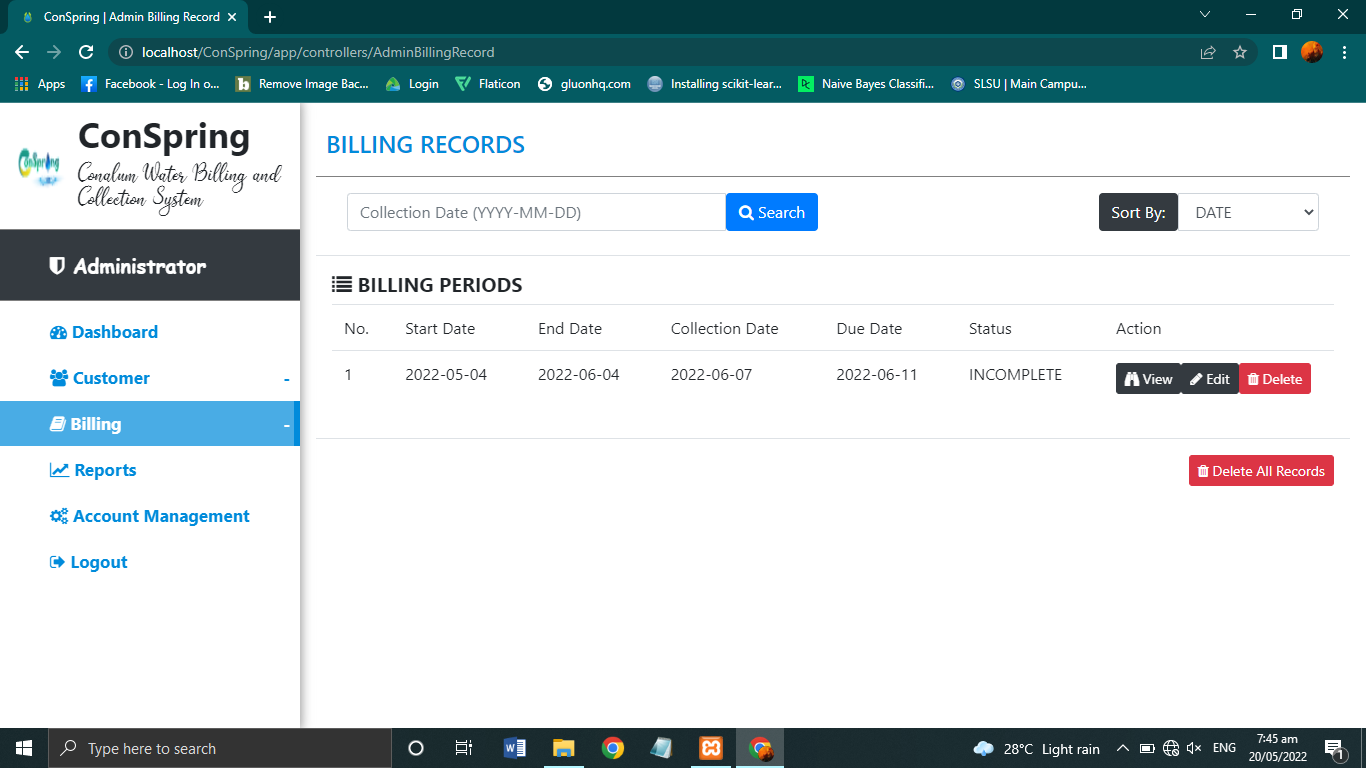
For Admin and Treasurer

**Login**



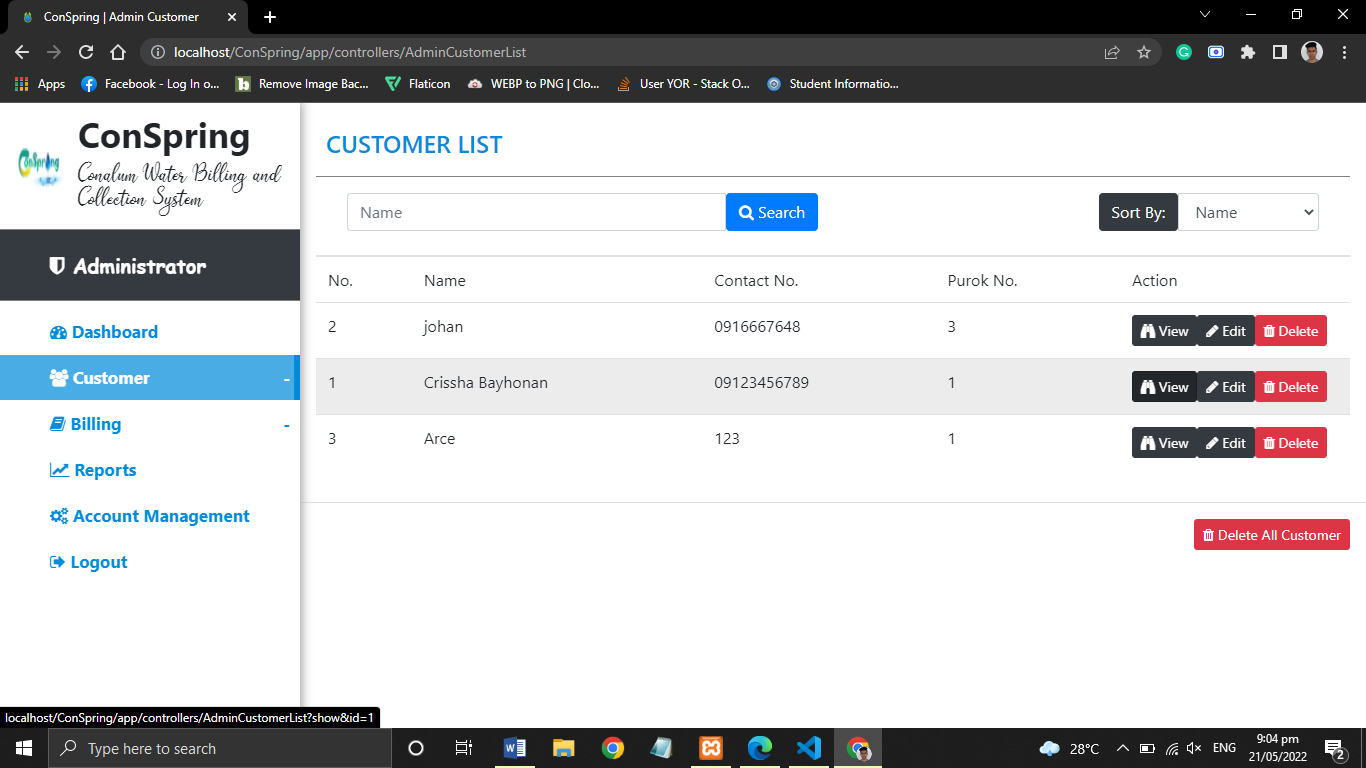
Choose to login either as admin, treasurer, or a customer. Both the Admin and Treasurer have the same capabilities in the system

**View, Edit, and Delete Billing Records**



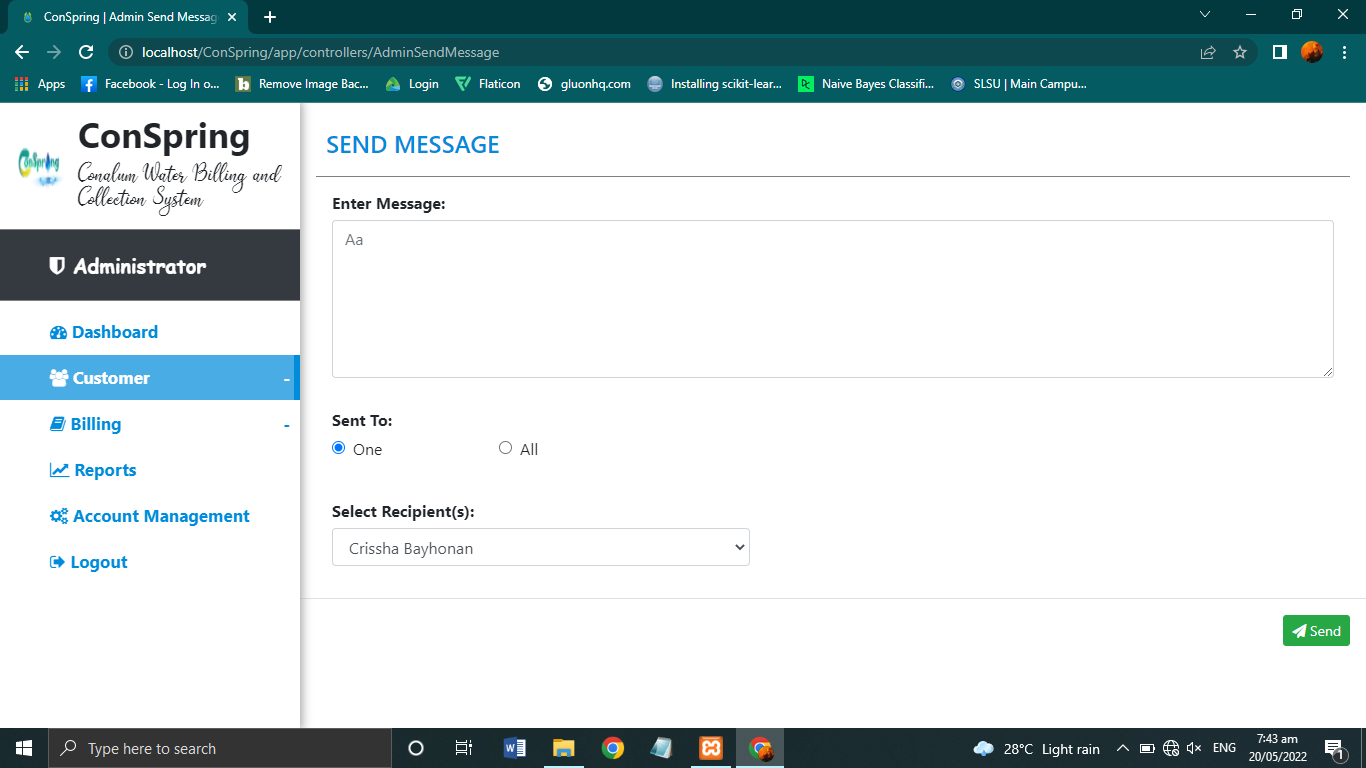
Both Admin and Treasurer can view, edit, and delete billing period data.

**View, Edit, and Delete Customer Records**

****

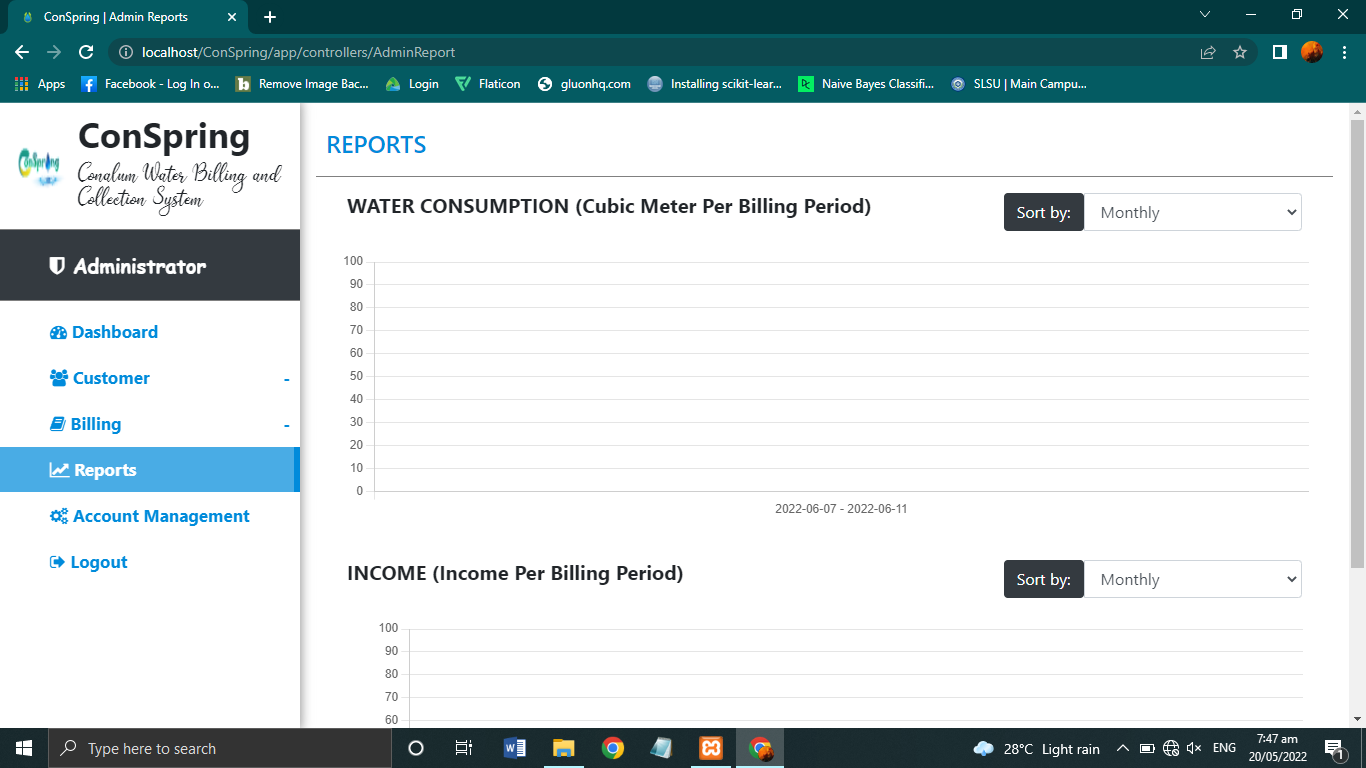
Admins and Treasurer can also view, edit and delete customer data.

**Send Message to Customer**

****

By entering message in the textbox then selecting recipients, Admins and the Treasurer can send message to the customer/s.

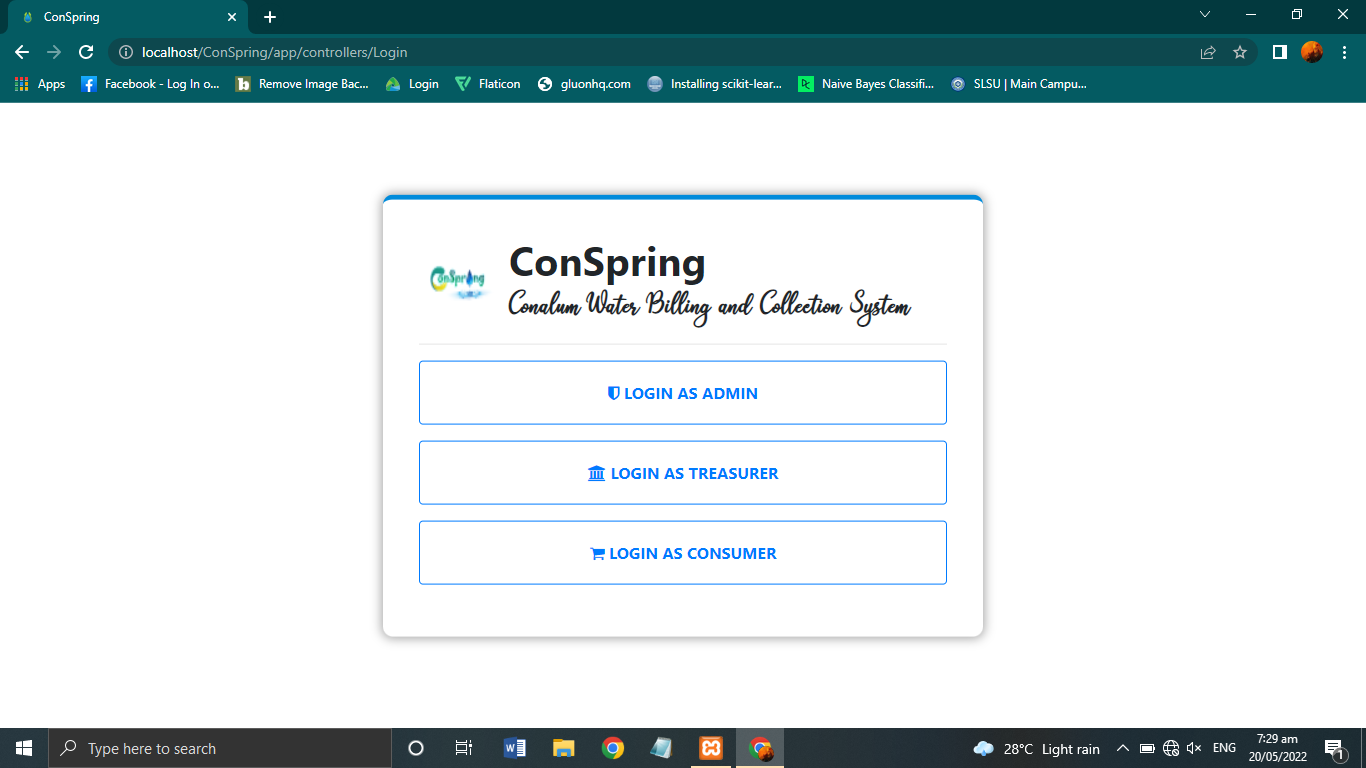
**Generate Reports**



They can also generate reports and sort them by clicking the dropdown menu on the right side. They can print it by clicking the print button below.

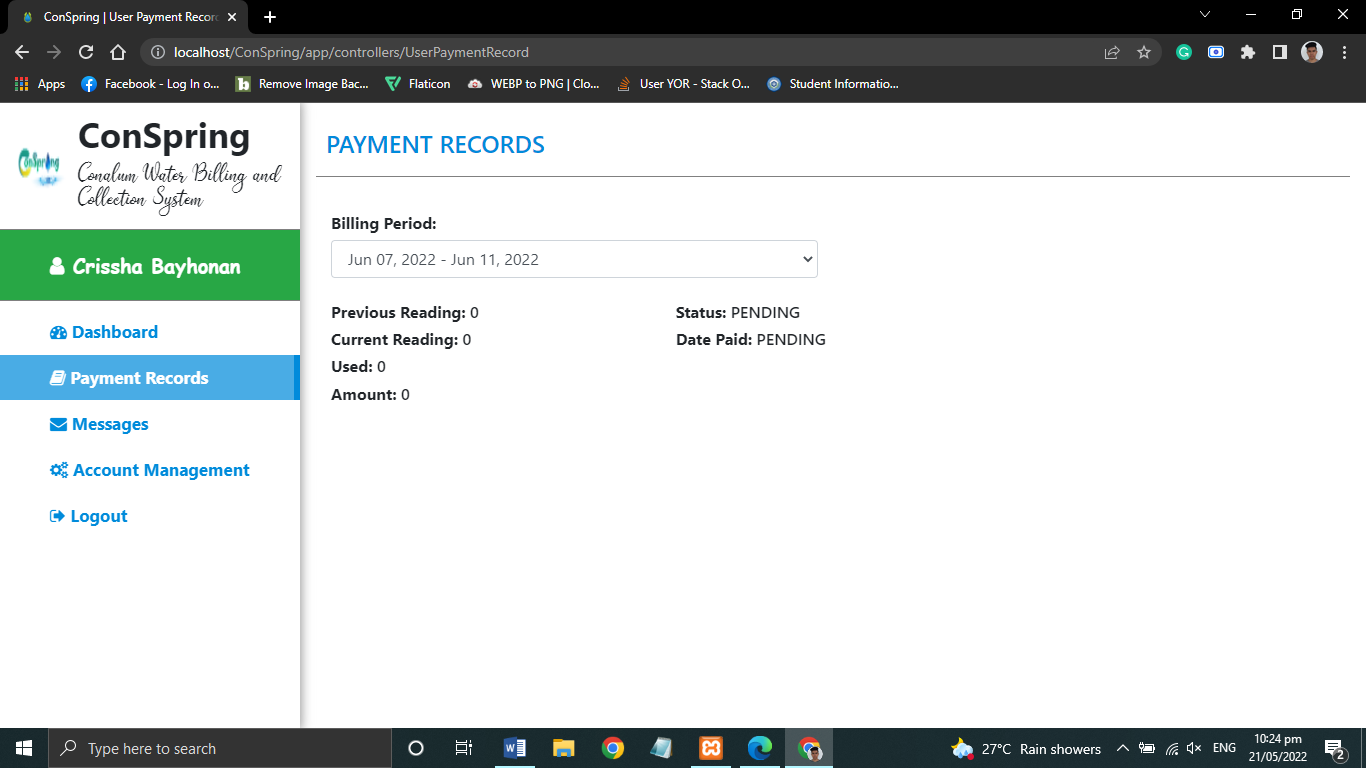
**For Customers**

**Login**



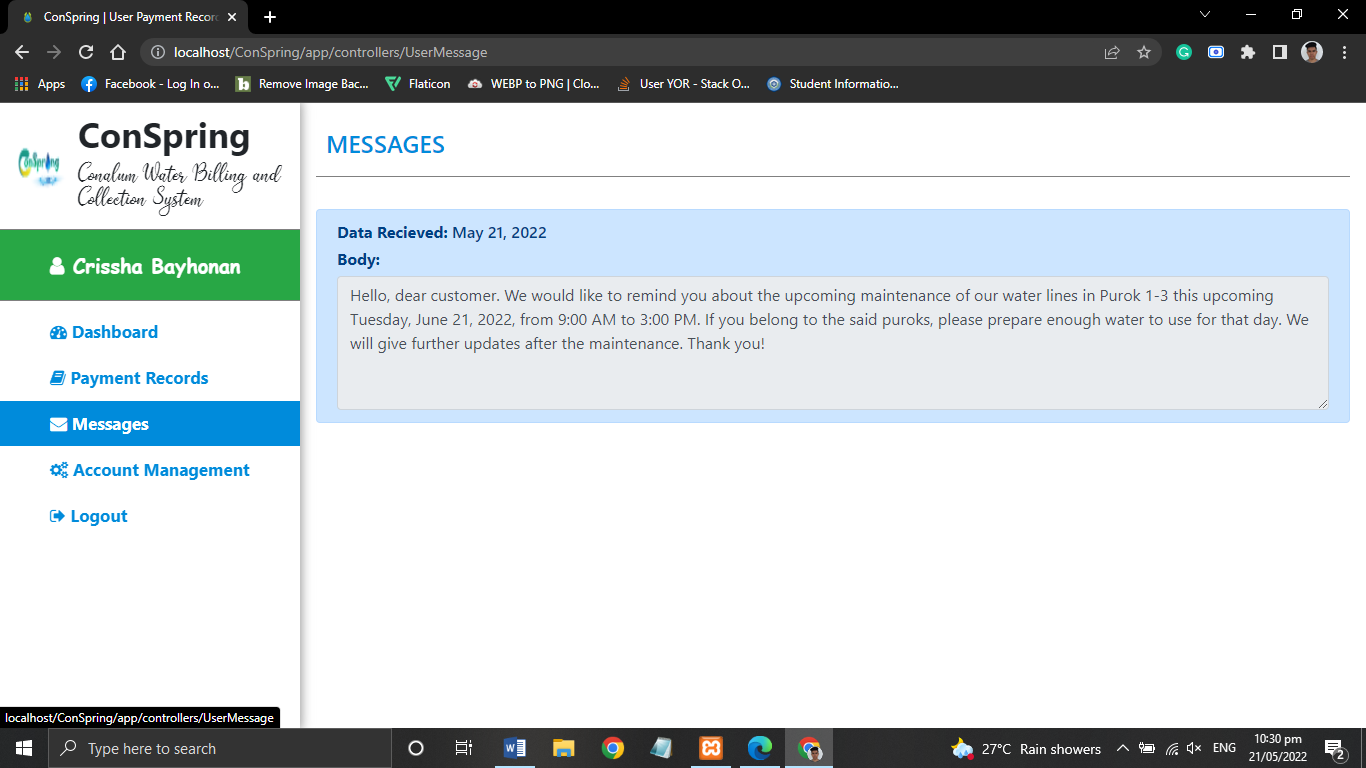
Customers can login by selecting the third option in the Login Page.

**View Payment Record**

****

Customers can view their payment records by clicking the Payment Records Tab.

**View Message**

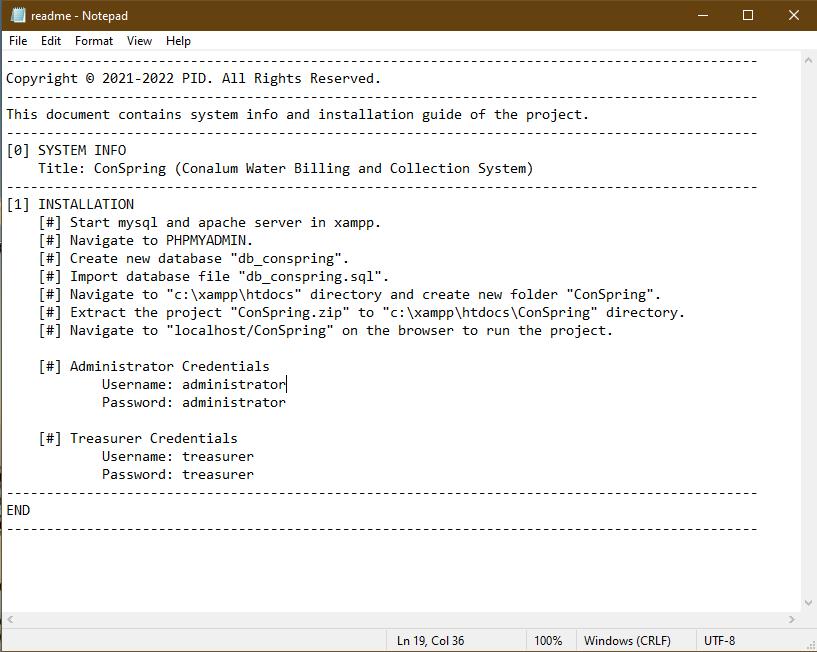


Customers can view their messages in the Messages tab.

**Appendix E**

**OTHER RELEVANT DOCUMENTS**

**Transferring System to a new Laptop/Computer**



**Appendix F**

**WORKING TITLE FORM**

Republic of the Philippines

**SOUTHERN LEYTE STATE UNIVERSITY**

Sogod, Southern Leyte

Website: [www.slsuonline.edu.ph](http://www.slsuonline.edu.ph)

Email: [slsumaincampus@gmail.com/](mailto:slsumaincampus@gmail.com/)

[op@slsuonline.edu.ph](mailto:op@slsuonline.edu.ph)

Telefax: (053) 382-3294

***College of Computer Studies and Information Technology***

**Proponents/Researchers:**

|  |
| --- |
| 1) Abrantes, Jexter Jay A. |
| 2) Bayhonan, Crissha P. |
| 3) Sarong, Johann Von E. |
| 4) Tafalla, Arce L. |

**Proposed Project Title:**

|  |
| --- |
| **ConSpring**: A Water Billing and Collection System  For Barangay Conalum, Inopacan, Leyte |

|  |  |
| --- | --- |
| **Submitted by:**      CRISSHA P. BAYHONAN  (Signature of Project Manager over printed name)    Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | **Noted:**      RHODERICK D. MALANGSA  (Signature of Adviser over printed name)    Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| **Recommending Approval:**      \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (Signature of Patent Searcher over printed name)    Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | **Approved:**      ALEX C. BACALLA  (Signature of the Dean over printed name)    Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

**Appendix G**

**GRAMMARIAN’S CERTIFICATION**

Republic of the Philippines

**SOUTHERN LEYTE STATE UNIVERSITY**

Sogod, Southern Leyte

Website: [www.slsuonline.edu.ph](http://www.slsuonline.edu.ph)

Email: [slsumaincampus@gmail.com/](mailto:slsumaincampus@gmail.com/)

[op@slsuonline.edu.ph](mailto:op@slsuonline.edu.ph)

Telefax: (053) 382-3294

***College of Computer Studies and Information Technology***

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_

**GRAMMARIAN‘S CERTIFICATE**

This is to certify that the undersigned has reviewed and went through all the pages of the proposal project study / research entitled “SLSU OJT Timesheet Monitoring and Journal System using Face Recognition” as against the set of structural rules that governs the composition of sentences, phrases and words in the English language.

Signed:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Grammarian

Conforme:

Crissha P. Bayhonan

Project Manager

**Appendix H**

**CURRICULUM VITAE**



### Jexter Jay Abrantes

Home Address: Brgy. Tinago, Inopacan Leyte

Mobile: 09166676482

Email: jexterabrantes@gmail.com

**SKILLS**

**OBJECTIVES**

**OEDUCATIONAL BACKGROUND BJECTIVES**

▪ To apply On-the-Job training in the field of Information Technology to gain more knowledge and to develop my skills in the field of my specialization. **SKILLS**

## SKILLS

* Microsoft Office Literate
* Graphic Design
* Programming Skills /C#, PHP, HTML/CSS

## PERSONAL INFORMATION

|  |  |
| --- | --- |
| Date of Birth | : Jan. 16, 2000 |
| Place of Birth | : Pasig, Manila |
| Civil Status | : Single |
| Citizenship | : Filipino |
| Height | : 5’5” |
| Weight | : 65 kg |
| Age | : 22 |
| Gender | : Male |

Religion : Roman Catholic

Languages or dialects spoken and written: Tagalog, Bisaya, English

|  |  |  |
| --- | --- | --- |
| **EDUCATIONAL BACKGROUND** | | |
| College: | Southern Leyte State University (SLSU)  *Bachelor of Science in Information*  *Technology*  *Major in Programming* | 2018 – present |
| Secondary: | Tinago National High School  *Accountancy, Business and Management*  *(ABM)* | 2017 – 2018 |
| Elementary: | Inopacan Parish Academy 2011 – 2012  Brgy. Poblacion, Inopacan, Leyte | |
| **SEMINARS, TRAINING ATTENDED, CERTIFICATES RECEIVED AND EXPERIENCES** | | |

**ERTIFICATES RECEIVED AND EXPERIENCES**

### Microsoft Office Specialist Training and Examination

26th of January 2019

MOS Certificate-Office Word 2016

Southern Leyte State University Sogod, Southern Leyte

### Image Processing and Artificial Intelligence

2nd of May 2019

USSC BUILDING PLENARY HALL, SLSU - MC Sogod, Southern Leyte.

## CHARACTER REFERENCES

**NCES**

### ▪ Dr. Rhoderick D. Malangsa, DIT

Dean, College of Computer Studies in Information Technology

SLSU, Sogod, Southern Leyte

▪ **James Brian Flores, Ph. D.**

Dept. Head, Bachelor of Science in Information Technology

**I HEREBY CERTIFY** that the above information is true and correct to the best of my knowledge and belief.



### Crissha P. Bayhonan

Home Address: Brgy. Conalum, Inopacan Leyte

Mobile: 09272858014

Email: crisshabayhonan@gmail.com

**SKILLS**

**OBJECTIVES**

**OEDUCATIONAL BACKGROUND BJECTIVES**

▪ To apply On-the-Job training in the field of Information Technology to gain more knowledge and to develop my skills in the field of my specialization. **SKILLS**

## SKILLS

* Microsoft Office Literate
* Graphic Design
* Programming Skills /C#, PHP, HTML/CSS

## PERSONAL INFORMATION

|  |  |
| --- | --- |
| Date of Birth | : Nov. 19, 1999 |
| Place of Birth | : Conalum, Inopacan Leyte |
| Civil Status | : Single |
| Citizenship | : Filipino |
| Height | : 5’2” |
| Weight | : 65 kg |
| Age | : 22 |
| Gender | : Female |

Religion : Roman Catholic

Languages or dialects spoken and written: Tagalog, Bisaya, English

|  |  |  |
| --- | --- | --- |
| **EDUCATIONAL BACKGROUND** | | |
| College: | Southern Leyte State University (SLSU)  *Bachelor of Science in Information*  *Technology*  *Major in Programming* | 2018 – present |
| Secondary: | Tinago National High School  *Accountancy, Business and Management*  *(ABM)* | 2017 – 2018 |
| Elementary: | Conalum, Inopacan, Leyte 2011 – 2012  Brgy.Conalum, Inopacan, Leyte | |
| **SEMINARS, TRAINING ATTENDED, CERTIFICATES RECEIVED AND EXPERIENCES** | | |

**ERTIFICATES RECEIVED AND EXPERIENCES**

### Microsoft Office Specialist Training and Examination

26th of January 2019

MOS Certificate-Office Word 2016

Southern Leyte State University Sogod, Southern Leyte

### Image Processing and Artificial Intelligence

2nd of May 2019

USSC BUILDING PLENARY HALL, SLSU - MC Sogod, Southern Leyte.

## CHARACTER REFERENCES

**NCES**

### ▪ Dr. Rhoderick D. Malangsa, DIT

Dean, College of Computer Studies in Information Technology

SLSU, Sogod, Southern Leyte

▪ **James Brian Flores, Ph. D.**

Dept. Head, Bachelor of Science in Information Technology

**I HEREBY CERTIFY** that the above information is true and correct to the best of my knowledge and belief.



### Johann Von E. Sarong

Home Address: Brgy. Tinago, Inopacan Leyte

Mobile: 09166676486

Email: sarongjohan@gmail.com

**SKILLS**

**OBJECTIVES**

**OEDUCATIONAL BACKGROUND BJECTIVES**

▪ To apply On-the-Job training in the field of Information Technology to gain more knowledge and to develop my skills in the field of my specialization. **SKILLS**

## SKILLS

* Microsoft Office Literate
* Graphic Design
* Programming Skills /C#, PHP, HTML/CSS

## PERSONAL INFORMATION

|  |  |
| --- | --- |
| Date of Birth | :July 7, 1998 |
| Place of Birth | : Paranaque city, Manila |
| Civil Status | : Single |
| Citizenship | : Filipino |
| Height | : 5’4” |
| Weight | : 55 kg |
| Age | : 23 |
| Gender | : Male |

Religion : Roman Catholic

Languages or dialects spoken and written: Tagalog, Bisaya, English

|  |  |  |
| --- | --- | --- |
| **EDUCATIONAL BACKGROUND** | | |
| College: | Southern Leyte State University (SLSU)  *Bachelor of Science in Information*  *Technology*  *Major in Programming* | 2018 – present |
| Secondary: | Tinago National High School  *Accountancy, Business and Management*  *(ABM)* | 2017 – 2018 |
| Elementary: | Sampaloc Site II Elementary School  BF Homes Paranaque city 2009 – 2010  Brgy. Poblacion, Inopacan, Leyte | |
| **SEMINARS, TRAINING ATTENDED, CERTIFICATES RECEIVED AND EXPERIENCES** | | |

**ERTIFICATES RECEIVED AND EXPERIENCES**

### Microsoft Office Specialist Training and Examination

26th of January 2019

MOS Certificate-Office Word 2016

Southern Leyte State University Sogod, Southern Leyte

### Image Processing and Artificial Intelligence

2nd of May 2019

USSC BUILDING PLENARY HALL, SLSU - MC Sogod, Southern Leyte.

## CHARACTER REFERENCES

**NCES**

### ▪ Dr. Rhoderick D. Malangsa, DIT

Dean, College of Computer Studies in Information Technology

SLSU, Sogod, Southern Leyte

▪ **James Brian Flores, Ph. D.**

Dept. Head, Bachelor of Science in Information Technology

**I HEREBY CERTIFY** that the above information is true and correct to the best of my knowledge and belief.



### Arce L. Tafalla

Home Address: Brgy. 57, White Lane, Tacloban City

Mobile: 09168591182

Email: arcetafalla13@gmail.com

**SKILLS**

**OBJECTIVES**

**OEDUCATIONAL BACKGROUND BJECTIVES**

▪ To apply On-the-Job training in the field of Information Technology to gain more knowledge and to develop my skills in the field of my specialization. **SKILLS**

## SKILLS

* Microsoft Office Literate
* Graphic Design
* Programming Skills /C#, PHP, HTML/CSS

## PERSONAL INFORMATION

|  |  |
| --- | --- |
| Date of Birth | : Oct. 16, 1994 |
| Place of Birth | : Daraupay, Capoocan Leyte |
| Civil Status | : Single |
| Citizenship | : Filipino |
| Height | : 5’0” |
| Weight | : 58 kg |
| Age | : 27 |
| Gender | : Male |

Religion : Seventh-Day Adventist

Languages or dialects spoken and written: Tagalog, Bisaya, English

|  |  |  |
| --- | --- | --- |
| **EDUCATIONAL BACKGROUND** | | |
| College: | Southern Leyte State University (SLSU)  *Bachelor of Science in Information*  *Technology*  *Major in Programming* | 2018 – present |
| Secondary: | Sagkahan National High School | 2008 - 2012 |
| Elementary: | Tacloban City Adventist Elem. School 2001 - 2007  Tacloban City, Leyte | |
| **SEMINARS, TRAINING ATTENDED, CERTIFICATES RECEIVED AND EXPERIENCES** | | |

**ERTIFICATES RECEIVED AND EXPERIENCES**

### Microsoft Office Specialist Training and Examination

26th of January 2019

MOS Certificate-Office Word 2016

Southern Leyte State University Sogod, Southern Leyte

### Image Processing and Artificial Intelligence

2nd of May 2019

USSC BUILDING PLENARY HALL, SLSU - MC Sogod, Southern Leyte.

## CHARACTER REFERENCES

**NCES**

### ▪ Dr. Rhoderick D. Malangsa, DIT

Dean, College of Computer Studies in Information Technology

SLSU, Sogod, Southern Leyte

▪ **James Brian Flores, Ph. D.**

Dept. Head, Bachelor of Science in Information Technology

**I HEREBY CERTIFY** that the above information is true and correct to the best of my knowledge and belief.

**GLOSSARY**

**Water Billing System -** is an automated system that was based on paying water bills. This system can manage transaction such as creating an invoice and paying the bills of customers. It will track all the records of the customer if they have paid on the due date or not.

**water billing and collection -** The system generates billing statement that indicates the water consumed and its corresponding cost. Collection covers the recording of consumers' payments and necessary for the report generation as required by treasurers' department of the said unit.

**manual technique -** It is most literally work done with the hands (the word "manual" comes from the Latin word for hand) and, by figurative extension, it is work done with any of the muscles and bones of the body.

**tracking of records -** Records tracking is the component of a records management system that ensures that you can locate records when you need to use them. Accurate recording and knowledge of the whereabouts of all paper records is essential if the information they contain is to be located quickly and efficiently.

**important customer information -** Customer data is information held on file about customers by a store or other business, usually including names, contact details, and buying habits. Customer data are the firsthand responses that are obtained from customers through investigation or by asking direct questions.