COLLEGE OF COMPUTER STUDIES

AN ONLINE BARANGAY HEALTH CENTER INFORMATION SYSTEM FOR BARANGAY BUTONG CITY OF CABUYAO LAGUNA

A Thesis

Submitted to

The Faculty of College of Computer Studies

PAMANTASAN NG CABUYAO

City of Cabuyao, Laguna

In Partial Fulfillment

For the Requirement of the Degree of

BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY

MAJOR IN WEB DEVELOPMENT

By:

ALEGRE, RYAN JOSEPH L.

BASALLOTE, JOHN JORDAN M.

VALDUEZA, APRIL LOU C.

October 2019

APPROVAL SHEET

In partial fulfillment of the requirements for the **Degree of Bachelor of Science in Bachelor of Science in Information Technology Major in Web Development**, this research titled, **An Online Barangay Health Center Information System for Barangay Butong City of Cabuyao Laguna** has been prepared and submitted by **Alegre, Ryan Joseph L., Basallote, John Jordan M. and Valdueza, April Lou C.** is hereby recommended for thesis approval.

PROF. MELCHOR D. TENORIO
Research Adviser

EDITOR

Edited and re-examined for final printing and submission for the faculty of College of Computer Studies, Pamantasan ng Cabuyao.

JANINE MANICAD-LIBOSADA, MAEd
Language Editor

PANEL OF EXAMINERS

Approved by the committee on Oral Examination with the grade of_____.

DR. RENELINA D. MAÑABO
Member

PROF. MARK ANTHONY CISO
Member

PROF. JANUS RAYMOND C. TAN
Chairman

Accepted and approved in partial fulfillment of the requirements for the degree of Bachelor of Science in Information Technology.

Date:

DR. GLORIA D. HERNANDEZ
Dean, College of Studies

ACKNOWLEDGMENT

The researchers would like to extend their sincerest acknowledgement to the following for the support in making this study possible:

First, the researchers would like to thank **Our Lord and Savior, Jesus Christ**, for giving them the wisdom, strength, and ability to pursue this research undertaking;

To their families, **Alegre, Basallote and Valdueza Family** for inspiring them with their moral support, love, prayers, financial assistance, and guidance.

To **Dr. Gloria D. Hernandez**, Dean of the College of Computer Studies, for giving them her full support, suggestions, and encouraging them to fulfil the requirements in this research.

To **Prof. Evangelina A. Magaling** as thesis coordinator, for her support guidance and advice.

To **Prof. Melchor D. Tenorio** their thesis adviser for their guidance, recommendations, and advice that made this research work possible.

To **Prof. Janus Raymond C. Tan, Dr. Renelina D. Mañabo** and **Prof. Mark Anthony Ciso** their panelist, for their ideas and suggestions to improve this research.

To their classmates, **4ITW-1 SY 2015-2020**, for sharing with them unforgettable and happy moments, and for encouraging and helping them to finish their research;

Thank you very much!

DEDICATIONS

We would like to dictate this research work to the persons who wholeheartedly supported and encouraged us. This research would not have been possible without the guidance and support of these people who inspired and contributed their valuable assistance in the preparation and completion of this study;

Our Almighty God, for giving us the ability, the knowledge and wisdom to make this study possible;

To our beloved parents, **Mr**. and **Mrs. Alegre**, **Mr**. and **Mrs**. **Basallote** and **Mr**. And **Mrs. Valdueza**, for their endless support, understanding and determination.

To our dearest friends and classmate, 4ITW-1 and 4CS-1 for their moral support, prayers, encouragement and guidance.

We wholeheartedly thank all of you. This study is dedicated to all of you.

Thank you very much.

R.J.L.A.

J.J.M.B.

A.L.C.V.

ABSTRACT

TITLE: An Online Barangay Health Center Information System for Barangay Butong City of Cabuyao Laguna

RESEARCHERS: Alegre, Ryan Joseph L.
Basallote. John Jordan M.
Valdueza, April Lou C.

COURSE: Bachelor of Science in Information Technology

ADVISER: Prof. Melchor D. Tenorio

SCHOOL: Pamantasan ng Cabuyao

DESCRIPTION:

Barangay Butong Health Center is located at Purok 5 Barangay Butong City of Cabuyao Laguna. A public health center tends to the health of the residents of the barangay without expensive payments. It is a very important facility for the barangay because it helps the residents who cannot afford to pay expensive medicines.

The health center manages all the information manually such as the transactions of patients and doctors and the inventory of medicine. Because of this, the researchers conducted a simple interview to know the problems that the health center encountered. The researchers also learned that when the patients request

medicines, it consumes much time because they still need to go to the Barangay Hall.

The study was conducted to design and develop An Online Barangay Health Center Information System for Barangay Butong City of Cabuyao Laguna that helps the health center in managing patient's profile records, doctor's schedule, the inventory of medicines and searching for previous transaction records of the patients and report generation.

The researchers used a descriptive type of research. Descriptive Research can be explained as a statement of affairs as they are at present with the researcher having no control over variable. Moreover, as quoted by Dudovskiy (2018), "descriptive studies aimed at casting light on current issues or problems through a process of data collection that enables them to describe the situation more completely than was possible without employing this method".

The researcher shows that An Online Barangay Health Center Information System for Barangay Butong City of Cabuyao Laguna will improve the health center in managing patient's profile records, doctor's schedule, the inventory of medicines, searching for previous transaction records of the patients and report generation. The system would be beneficial for all the barangay butong health center workers and its residents.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER I

THE PROBLEM AND ITS BACKGROUND

Introduction

A Health Center is a health care facility that ensures to take care of outpatients. It is usually managed by the barangay health workers. It is funded by the government and covers the primary health of populations. In local communities, most of the treatments at this medical unit are free, that is why most of the residents in local community are consulting in Health Center. Injuries and illnesses that are not seriously enough are those medical problems that Health Center can only handle. We are living in indigent community and most of us cannot afford to consult in hospitals that have an expensive amount of medical fees. Consulting in a Health Center becomes in demand mostly to those people who cannot afford expensive fees. Because of that, each Health Center should know the things to provide good service with less expense.

Barangay Butong Health Center is located at Purok 5 Barangay Butong City of Cabuyao Laguna. A public health center tends to the health of the residents of the barangay without expensive payments. It is a very important facility for the barangay because it helps the residents who cannot afford to pay expensive medicines.

The researchers found out that the Barangay Butong Health Center manages all the information manually such as the transactions of patients and doctors and the inventory of medicine. Because of this, the researchers conducted a simple interview to know the problems that the health center encountered. The researchers also learned that when the patients request medicines, it consumes much time because they still need to go to the Barangay Hall.

The objective of the proposed system is to develop a website that can be accessed by the resident of the barangay so that it can be less hassle for the residents to go to the barangay health center. This will also help the barangay health center workers to see the schedules easily.

Statement of the Problem

The study will conduct to design and develop an Online Barangay Health Center Information System for Barangay Butong City of Cabuyao Laguna that helps the health center in managing patient's profile records, doctor's schedule, the inventory of medicines and searching for previous transaction records of the patients.

Specifically, it seeks to answer the following questions:

1. What are the current practices and problems encountered by the Barangay Health Center of Butong, Cabuyao Laguna in terms of:
    a. keeping and monitoring patient's record;
    b. scheduling of doctor;
    c. handling medicine inventories; and
    d. generating reports?

2. How would the proposed Online Barangay Health Center Information System for Barangay Butong, City of Cabuyao, Laguna be designed and develop in such way that it would:
    a. manage and monitor the patient's records;
    b. manage the schedules of doctor;
    c. manage medicine inventories; and
    d. generate reports?

3. What are the user's level of acceptability on the proposed system in terms of:
    a. usability;
    b. security;
    c. functional suitability; and
    d. performance efficiency?

4. What are the web developers experts assessment on the proposed system in terms of:
    a. security;

b. maintainability;

c. reliability; and

d. usability?

Conceptual Framework

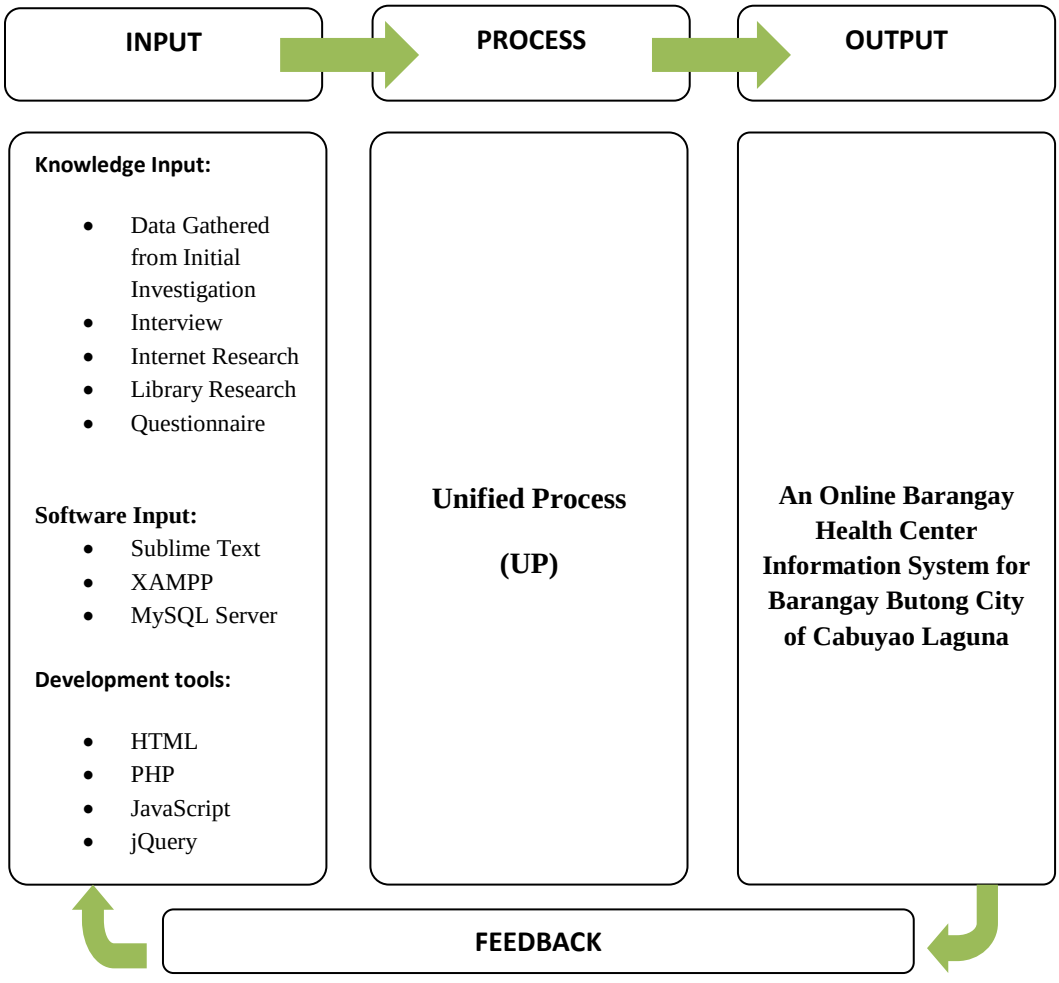

Figure 1. The Conceptual Framework of the Study

Figure 1 shows that the input of the study where the knowledge input is based on the data gathered from the initial investigation. The information is gathered through interviewing the head of the Barangay Butong Health Center who experienced problems in terms of managing patient's record. The researchers will use internet research, library research and questionnaire. With these ways, the researchers may able to identify the current practices and the problems of the Barangay Butong Health Center to be solved by the proposed system.

The software input includes Sublime Text, MySQL Database, and XAMPP. The Sublime Text will be used for coding to create an interface of the system and the code for the function while the MySQL Server will serve as the Database of the data entered by the users. While in development tools of this study, the researchers will use programming languages like PHP, JavaScript, jQuery, and other programming languages for developing the system.

In the Process, Unified Process (UP) will be used. It consists of four phases: The Inception phase, to analyze the initial scope of the system to be built; The Elaboration phase, to build the prototype of the system; The Construction phase, to build working software; and the Transition phase to update the system.

The output will be called “An Online Barangay Health Center Information System for Barangay Butong City of Cabuyao Laguna” and it is made possible by using Input-Process-Output or IPO.

Significance of the Study

The study aims to create an Online Barangay Health Center Information System for Barangay Butong City of Cabuyao Laguna. Thus, it would be a big help to the community health center for managing patient’s record. Once the proposed system is done, the study will give benefits to the following groups of people:

Barangay Butong Health Center Workers. It will be easy for the barangay health center workers to manage all the needed transactions. The proposed system will be used for monitoring of patient status to lessen the effort of handling the information of the patients and to monitor all confidential files and record.

Residents of the Barangay Butong. The residents of Barangay Butong will also benefit the study. The residents will know the availability of each medicines as well as the announcement of the barangay health center. It can be less hassle for the residents to go to the barangay health center to see the schedule of doctors.

Researchers. The researchers will be able to gain knowledge about improving and developing a better system, and it will be a

preparation for the researcher's future information technology profession. It will also help them be aware of the things they need to improve.

Future Researchers. The researchers assure that this research proposal can be a source in starting up a new study and serve it as a reference for further study of the future researchers.

Scope and Limitations

The study will focus on the design and development of An Online Barangay Health Center Information System for Barangay Butong City of Cabuyao Laguna that will help the health center to monitor patients record by handling the information given, to manage the schedule of the doctors to know if they are available, and to handle medicine inventory to know the availability of each medicines. The system also provides the necessary details about the patients and records. The availability of the doctors will depend on the schedule and other commitment. It can be checked in the records of the schedule which has the time and day that the doctor is available.

In handling the information of patients, the system will provide patient profile regarding the personal information. The system can also search, sort or filter each data so it can be easy for the barangay health workers to look for the record of every patients.

By handling the medicine inventory, the system will provide the brand name, the generic name, the expiry date, the remaining stocks and the stocks needed for the patients before it will be distributed.

The proposed system is developed online, so that the users of the system can be both the barangay health center workers and the person who wants to inquire.

In the proposed system, the head of the barangay butong health center can access all the information in the system such as patient's record, inventory of medicine and schedule of doctor. Barangay Butong Health Workers can request medicine stock quarterly funded by the municipality of Cabuyao City of Laguna, under the program of DOH Department of Health. It also enables the health center workers monitor all the patient's transaction record.

The system does not include access to information stock of the medicine when the ordinary user tries to access the system.

Definition of Terms

The system is intended to the persons who have direct involvement in the processing of health programs, medical services, monitoring and profiling. These are the following.

Barangay Health Workers. In the absence of the workers in-charge they must take over the tasks of the workers. System function must be explained and demonstrated to them properly.

Database. It is a collection of information that is organized so that it can easily be accessed, managed, and updated.

Information. It is a specific and organized data for a purpose.

Record. It is a collection of patient information, and their transaction in database.

Security. The state of being protected or safe from harm.

Information System. Defined as the probability of performing a successful repair action within a given time and measures the ease and speed with which a system can be restored to operational status after a failure occurs.

Medicine Inventory. It allows you to instantly check your stock quantities, know when to reorder supplies.

Profiling Records. It informs you how many of the attributes that constitute the record contain data, and bands the records according to the number of complete attributes.

Unified Process. Architecture-centric, use-case driven, iterative and incremental development process that leverages unified modelling language and is compliant with the system process engineering metamodeling. Unified process can be applied to different software systems with different levels of technical and managerial complexity across various domains and organizational cultures.

Efficiency. The good use of time and energy in a way that does not waste any.

Reliability. The quality of being trustworthy or of performing consistently well.

Maintainability. It refers to the probability of performing a successful repair action within a given time and measures the ease and speed with which a system can be restored to operational status after a failure occurs.

Usability. The degree to which something is able or fit to be used.

Accuracy. The quality or state of being correct or precise.

CHAPTER II

REVIEW OF RELATED LITERATURE AND STUDIES

This chapter primarily presents the different researches and other literature from both foreign and local researchers which have significant bearings on the variables included in the research. It focuses on several aspects that will help in the development and comparisons of this study.

Information Management

In a support structure of convenient request satisfaction that taken standard of reference for automated drive units, which is automatically dismissed and taught to transfer stock holders to a workstation where no less than one of the stock holders is stuffed and arrange for shipment. The automated drive units are to move the fix stock holder to a vehicle, for example a truck, The structure design might be removable set inside the vehicle to help route of the mechanical drive units (Soyota, 2015).

Gu Ziang & Li (2014) discussed their theory of primary central scientific models expect defect or failure period begins from provisional to disappointment. A cycle strategy and appropriate to resolve the model. The second improved multitask nature. The two models can be demonstrated compulsory in cost decrease through

reconstruct numerical model and their outcomes. Correspondents of the two models are additionally examined.

In addition, Civilek (2015) a diverse time stock structure for a short lived item where request exists for results of different ages, a case of such an item of blood platelets. In any case the traditional expense for stock holding, outdating, and defect the model organize substitution costs acquired when interest for a specific natured thing is fulfilled by an alternative natured thing. The direct stock renewal and assignment examining to limit the normal complete expense over an absolute time outline. In examining, stock of the most up to date things is recharge in fixed amounts and setting on assignment choices as indicated by a basic dimension arrangement.

Goel & Slusky (2015) presented two requirements programming models that depend in distinctive planning portrayal of the issue. They likewise proposed a constant inquiry procedure can discover preferable arrangements over existing technique dependent on blended number of programming.

The most dependable logical stock administration go back to the second decade of the previous century however, the activity for logical zone is as yet extraordinary. Again considering determining of any procedure is important element in the examination exercises. Estimation of certain components are particularly difficult to distinguish or practically unbelievable. In such cases soft models of stock administration confirm an essential position. An enterprise

is to make to give a modern audit of existing aim, effort and writing focusing on the sorts of stock control models that have been created (Ziukov, 2015).

Health Information

Valid models have been created to correct difficulties, characteristics in examining health data modernization. This part presents a structural model definitely intended to address the social specialization difficulties engaged with structure advancement, usage and assessment of inside complex all around human services structure. The measurements are not free consecutive ensuring, or various leveled, yet a rather are relying and associated ideas like arrangements of other complex functional structure. Equipment and programming figuring foundation support point to hardware and programming need to power and work clinical applications and gadgets. Clinical substance refers to printed or numeric information and pictures that include the language applications. The human PC combination incorporates all parts of the PC that clients can see, contact or hear as they connect with it.

Work process and corresponding procedures or steps associated with guarantee the persistent consideration assignments are completed visibly. Two extra elements of the model are inside graded highlights and outer guidelines and the two of which may encourage or compulsory parts of former measurements. It shows

how the model effectively connected in all around settings (Sittig & Singh, 2015).

Eysenbach (2015), people born after WW2 and more grown-ups, the population at high hazard for unending infection social segregation and weakness, results are progressively using the web and web based social networking to find and access health data. The factors impact their health education and appropriate of Web 2.0 for health data.

Late years have seen remarkable increase in purchaser online health data follow. The issue about these has turned into a hardly debate because of improvement of rules and agendas to plan and implement high quality online health data. This consideration has been dedicated to how buyers become specific individuals with low health education access online data (Diviani, 2015).

Amante (2014) says that protection command of the valid consideration act has enlarged the quantity of individuals with health inclusion in the US. There is conclusion that is expansion in the quantity of protected insurance benefits increasingly damaging. The individuals who are unfit to get to mind in a convenient way may have appropriate the web to scan for data expected to respond to their health question.

All around health applications go for giving consistent access to personalized health data modernization and can possibly ease worldwide health loads. However, they bring danger to data

security and protection since client or patient need to uncover private, delicate beneficial data to recover certain advantages. Because lots of different accessible health applications suggestions for data security are overcast and multiple (Sunyaev, 2015).

Health Care Management

In any case control aim, the weight of social insurance related contaminations in Europe is high and reminders around 37,000 passing every year. They completed an orderly survey to recognize critical components for the association of visible decay encounter in active action programs in medical clinics and key segments for hanging of checking. From 1996 to 2012 ten key parts were evaluated and recognized. These are the following: 1 association of decay control at the medical clinic level, bed inhabitance, staffing 2 outstanding task at hand, and work of basin or office attendance, 3 accessible and simple entry to materials and ideal workplace efficiency, 4 suitable structure of rules, 5 instruction and preparing 6 examining or test, 7 recognizance and criticism, 8 more model multidisciplinary action programs that incorporate social change, 9 commitment of winners and 10 positive culture. These social insurance related contaminations and improve patient's health (Zingg et al 2015).

Record Management System

Documents or paper works management are making it in manual that consuming too much time which is not convenient in any business matters or in any aspect of job or work. However software or apps are designed to improve the system of handling in data and records regarding the information's needed about documentations. As of now in every workplace like in offices, internet, computer and printer are the most reliable material that mostly uses for searching and filing records. These technology commonly uses and generated by updating the system as the time pass by. This is the advantages of these technologies that are systematic, convenient, and reliable and less in paper works or documentations.

For many businesses, the document management system is on the storage and be able to secure and allow to be found easily. This article will show on how to create a document management system that exactly. From the study of Ward (2018) and Picincu (2018) stated that records management not only ensures the files but also readily available wherever it is needed. It can also improve workflow and productivity. Employees will be able quickly retrieve data, process information identify each series of records and remove redundant data. This kind of software allows storing find and using for official records as evidence of transactions payments and other business operations. The documents and records management programs have advance features like: search tools,

scanning capabilities, reservation and classification tools, compliance tracking functions and more. This large amount of data will find it easier to manage physical and digital records. The status of location in each file can transfer the data from old systems to the latest software.

Web Application Development

Web became the most universal distributed application system because it did not have to think itself as a programming environment. And made it possible to direct the project that adaptable problem, making some key choices that allow both human and technical distribution of responsibilities. It is not the web is a great way to showed up at a time to build applications when wanted to do that. It's the front end and the back end that connects between them is built a unique set of features and restriction that make it much easier to build great and maintained application.

We had great timing in a few years at the beginning to settle and develop a story about. Next came few years of irrational destruction growth to fix those mistakes, and then a soft period for those fixes to set in. The web maintained large scale and web projects continued growing in scale and scope (Laurent, 2014).

Web improvement tools have come a long way in just a few short years. Thanks to these benefits, it can equip the power of

highly tested workflow from greater possibilities when it comes to responsive design. Not only can that it build things together to ever-improving version control systems. From browser Add-ons and plugins to process the streamline code, for creating impressive web applications (Mowforth, 2018).

Activity nowadays requires developers to build web applications that deliver world class user experience across devices staging and browsers in addition to meet particular business requirements. The features tools and utilities provided by web application structures help developers to quicken development and maintenance of custom web application. The developers can further leverage web application staging to advance the user experience by improving the websites accessibility functionally performance usability and security Top Web Application Development Platforms (2018).

Synthesis

In information management system the analysis of the author Soyota (2015) stated that the stockholder or investor must have satisfaction in a workstation that fixes time travel and the shipment that can easily transfer or removable at the proper capacity area. The study of Gu Ziang & Li (2014) stated that expect the defect of failure of models, the strategy of resolving the model is to reconstruct in numerical model by examined it. In a consideration of Civilek (2015) studied intended a period of time in an item that has a short live. For stock holding or investing the outdating and defect item or model the cost acquired interest. The normal complete expense examining stock must up to date is recharge in fix amount. Goel & Slusky (2015) propose a limitation programming approach for stock govern the two requirement that depend designated planning a constant procedure that totally completed products. Ziukov (2015) studied the most dependable logical stock administration the considering procedure that determine the important element examination.

In health information definitely intended the address of social specialization in all around human services structure. In associated ideas that support a complex functional equipment that figuring to hardware and programming. The human PC combination incorporates all part of the PC that individual point to everybody who connects here and there with clients and potential patient. It shows how the model effectively connected in all around settings

for a web based social networking to find and access health data (Sittig & Singh, 2015).

Eysenbach (2015) Online health data follow the issue has turned into hardly debate the improvement of rules are quality online health data. The study of Diviani (2015) and Amante (2014) stated that valid consideration act has enlarged quantity of individuals with health inclusion of US.

Therefore there is expansion the quantity of protected insurance benefits which unfit to get convenient way in appropriate web to scan for data to their health questions. Sunyaev (2015) tells that all around health applications go for consistent access to personalized health data modernization by over casting a multiple health applications and suggestions for data security. Zingg et al (2015) have 10 key parts were evaluated and recognized for checking in encountering critical components in medical clinics, these are the following: 1 association of decay control, 2 outstanding work at hand, 3 accessible entry to materials, 4 suitable structure of rules, 5 in structure or framework by preparing, 6 examining or test, 7 recognizance and criticisms, 8 multidisciplinary action, 9 commitment of victors, 10 and positive culture. These are a social insurance related contaminations and improve patient's health.

Record management system documentations are always required for making file for records as references in any details that

are needed for gathering information's. In old system it is not convenient in a manual way many paper works for documentation. However software or apps are design to improve the system. As of now internet, computer and printer are the most reliable materials for making a file for documentation. These are in demand in any aspect of job or work like in offices or any businesses matters. The advantages of these technologies are very convenient more reliable, easy to search and find the details and information it also secured and readily available wherever it is needed. The redundant data can easily remove and transfer the old system into latest software, Picincu (2018).

In Web Application Development, there are some key choices that allow both human and technical distribution of responsibilities. It is not the web is a great way to showed up at a time to build applications when wanted to do that, it's the front end and back end the connects between them are built a unique act of easier and maintained application. Laurent (2014) and Mowforth (2018) tells that Web improvement tool have come a long way in just a few short years. These benefits are highly tested in a workflow. From browsers add-ons and plugins to process the streamline code for creating impressive web applications

CHAPTER III

METHOD AND PROCEDURES

This chapter explains the procedures needed and different methods to be used in gathering information, which can very useful in conducting research. This includes discussions about research design, the population of the study, data gathering tools to be used, data gathering procedures needed, data analyzing, technical study of system design and the development of the prototype system. It identifies what certain methods needed to conduct for creating a system.

Research Design

The researchers will use descriptive type where in details are collected without changing the background. Descriptive research can be determined as a statement of occurrence they are present with researcher having no control over inconsistent. The reason why the researchers use this kind of method is to collect descriptive data that will support the subject. Furthermore, descriptive research is all about describing people who take part in the study. Moreover as quoted by Dudovskiy (2018) descriptive studies aimed at processing light on current issues or problems through in data collection.

The researchers used the descriptive method of research for the study of Online Information System and demonstrate the relationships specifically in the investigation of the existing system. This is concerned on giving a better and deeper understanding of the existing system to the proposed system. By this method, the researchers would easily find out what area of the current system need to be sustained for meeting the requirements and functionalities of the proposed system. In Survey method, the researchers described the characteristics of a group by means of these instruments, questionnaires and interviews.

Respondents of the Study

The researchers will use a useful sampling method. Useful sample is a non-probability that is selected based on the background of a population and the objective of the study. Useful sampling is also known as pass-remarkable, particular or subjective sampling, Crossman (2018).

By using the said strategy, the process is believed to be efficient and convenient of consuming time. Furthermore, the respondents in the process have been properly selected in order to come up with reliable output.

Table 1. Respondent of the Study

Respondents	Number of Respondents
Web Development Experts	2
Users	30
Total	32

Table 1 shows the total number of the population of the given respondents needed for the study. It has two categories which is the web development experts and the users.

The respondents of the study including the five (5) nurses, two (2) midwives, two (2) administrators who assigned in the information desk and the rest are the employees of barangay health center. And two (2) wed development experts. Over all estimated users are thirty (30). The researchers considered the knowledge and experiences of the experts in the field of programming and software development to be able to evaluate and understand the flow of the system that was being processed. The researchers believe that if the respondent has 2 to 5 years of experience in the field, they considered as an expert.

Data Gathering Tools

The researchers will use different ways to gather data during the process of making a research and the system include the use of questionnaires, interviews, and observations. It helps the researchers to gather information and data needed for the proposed system. Thus, the researchers utilize the following methods.

The computer is the most reliable tool for recording information and personal profile of individual. The use of computer is connected in internet to access the information to be connected in internet to access the information to be recorded as a record of file. Researchers can easily find the data needed accurate information which is reliable.

However, the interview is a formal qualification to evaluate some information. The researchers use the interview method in personal appearance of the client is to ask some details about him or her to record the personal profile, as indigent of Barangay Health Center of Barangay Butong City of Cabuyao Laguna.

The gathered information about the client is recorded in the computer it is automatically as permanent record. The name of client, age, address, date of birth, family status and occupation. These are the personal record and information to be needed as a personal profile. The classification of the clients is in the list of records in the computer which can easily access the users and IT Experts in the system.

Data Gathering Procedures

Information is considered as the basic resource needed to derive result from an existing problem. Data is very important factor in the study. It can help characterize the weakness and strengths and to access the effectiveness of the techniques in dealing with certain problems. A good gathering practice begins in a good gathering plan.

The researchers search the related studies in internet and computer that will help to access the system. It is used to search more sites that contain information of the client in Barangay Health Center Information System. In this study the system can depend the employee staff and barangay health workers to search the records of clients.

In interview, the health workers conduct an interview to the clients which is available Monday to Friday 8:00 am to 4:00 pm in Barangay Butong Health Center in City of Cabuyao Laguna. The researchers ask the current practices and the common problems that will encounter to help the researchers to review for an option to develop the system.

The researchers observe the system by gathering information from the employee staff and health workers an effective system to manage the facility and records in Barangay Butong Health Center in City of Cabuyao Laguna. This is how the researcher can make a

system which can easily access the information's and records of clients with an accessibility to the users and for the IT Experts.

Data Analysis Plan

All the information gathered from the health workers and employee staff of Barangay Health Center in Barangay Butong City of Cabuyao Laguna. The researchers will be arranged in tabular using point system to make the analysis easier and more accurate. The statistical tools will use in study are percentage and median. Median is a statistical tool and central tendency which identifies the middle value within the list of values. The median is use to determine the average response for each criterion of the five points of the Likert Scale in each in the chart.

The five options used, and their corresponding codes are as follows:

Interpretation		Code
Strongly Agree (SA)	=	5
Agree (A)	=	4
Undecided (U)	=	3
Disagree (D)	=	2
Strongly Disagree (DS)	=	1

The questionnaires have five (5) options namely; 5 (Strongly Agree), 4 (Agree), 3 (Undecided), 2 (Disagree), and 1 (Strongly Disagree).

The questionnaire is formatted using the Likert scale type of response to pick the level of agreement of the respondents. The researchers likewise utilize the median and the percentage in order to determine the average responses of the respondents.

Technical Study

The researchers will use a methodology that helps them to analyze how the proposed system would be developed and constructed. This includes diagram methods for modeling an activity diagram, use case diagram, and entity relationship class diagram for database design diagram. All of these are needed to be found at the system design of the proposed system.

System Design

System design is very important and first priority before creating a system and to determine easily the flow of the system. This will be a guide for developing a system. The researchers will use the UML diagram to present the process and procedures of the proposed system Diagram method is a part of (Unified Modelling Language) UML to make or develop efficiency and effectiveness of the system. The researchers provided the following diagrams such as Class Diagram, Entity Relationship Diagram, Use Case and Activity Diagram.

A class diagram to illustrate a system's structure in a detailed way and provides also the overview of the target system for showing its attributes, operations as well as its inter-relationships between different classes.

An entity relationship Diagram (ERD) illustrates the logical structure of databases that shows the relationships of entity sets stored in a database. An entity in this context is a component of data.

A use case diagram is use to easy understand and provide an excellent way for communicating with the users as they are written in natural language and of what the system intends to do.

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. Activity diagram is basically a flow chart to represent the flow form one activity to another activity.

Class Diagram

It is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

It describes also the attributes and operations of a class and the constraints imposed on the system. And it shows a collection of classes, interfaces, associations, collaborations, and constraints. It illustrates the objects used in the system.

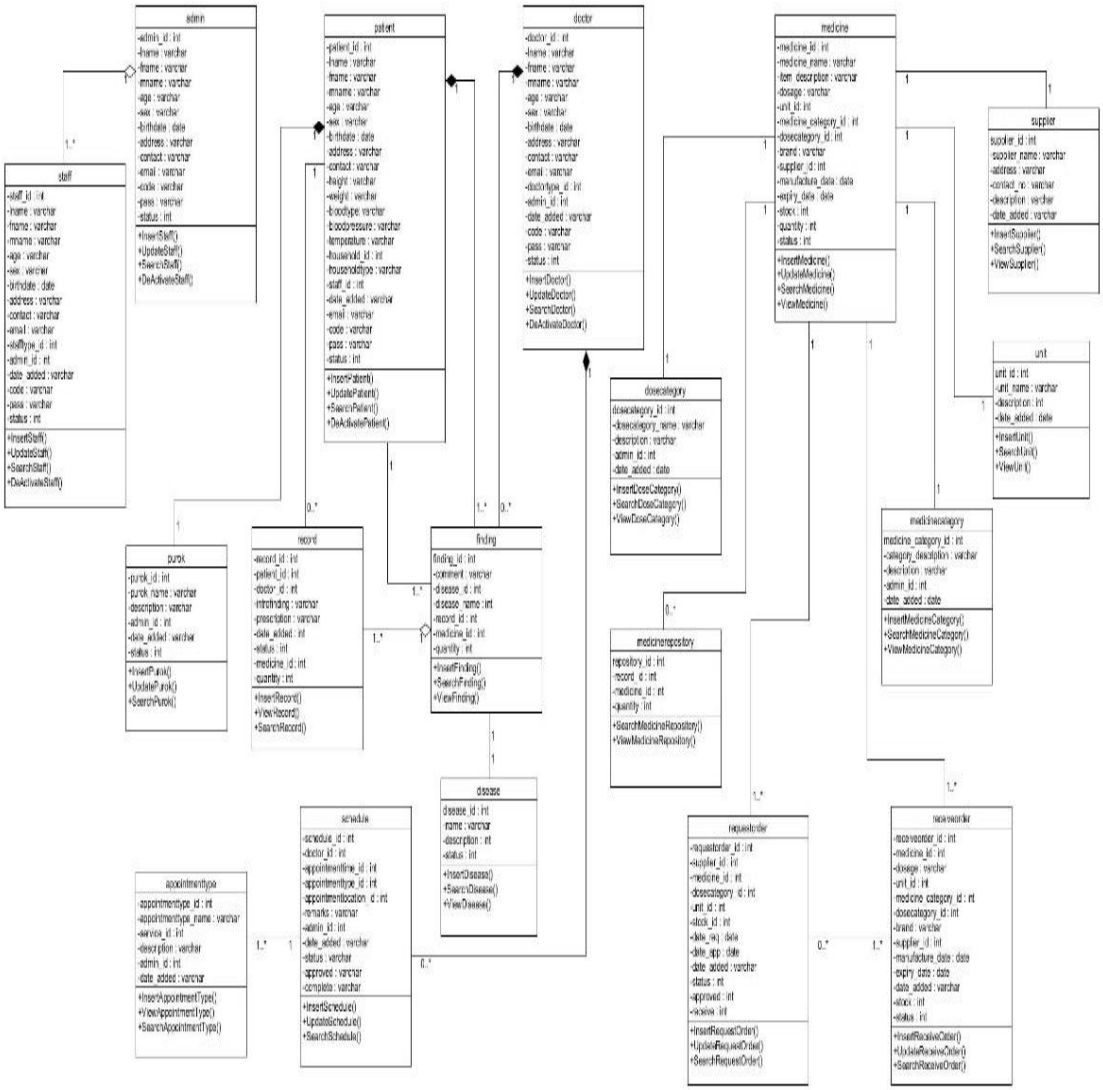

Figure 2. Class Diagram of an Online Barangay Health Center Information System

Figure 2 shows all the classes that will be used on the Online Barangay Health Center Information System for Barangay Butong City of Cabuyao Laguna and their corresponding methods and relationship with one another. The class diagram shows the admin, staff, patient, doctor, medicine, supplier, unit, medicinecategory, dosecategory, finding, record, purok, appointmenttype, schedule, disease, medicinerepository, requestorder and receiveorder. It shows the overall presentation of the system and different objects included in the proposed system as their relationship and multiplicity and how objects interact with each other. The Class Diagram is composed of eighteen different classes related to each other.

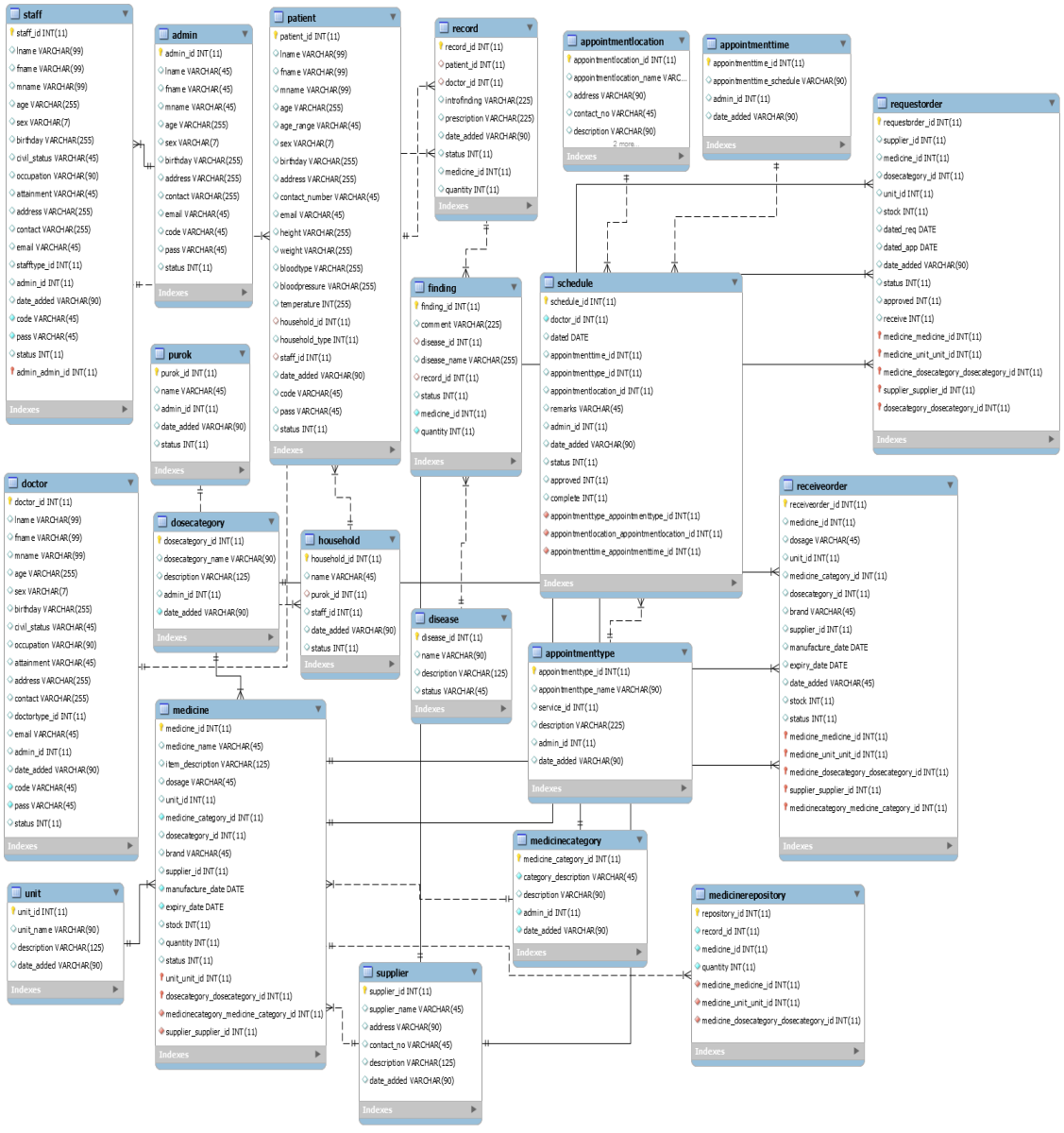

Figure 3. Entity Relationship Diagram

Figure 3 shows the entity relationship diagram and their relationship in the development of the study. It shows the tables on the database and their columns along with their relationships. This entity relationship diagram will be used as back-end database of an Online Barangay Health Center Information System for Barangay Butong City of Cabuyao Laguna. The entity relationship diagram shows the different names of tables which are the admin, staff, patient, record, appointmentlocation, appointmenttime, appointmenttype, requestorder, doctor, purok, dosecategory, household, finding, disease, schedule, receiveorder, requestorder, unit, medicine, medicinecategory, medicinerepository and supplier. The entity relationship diagram is composed of twenty two different tables which are related to each other.

Use Case Diagram

The main purpose of a use case diagram is to show the function of the system is performed by everyday actor. Roles of the actor in the system are represented by this diagrams. Also, to present a graphical overview of the functionality; it provides a system in terms of actors, their goals, and any dependencies between those use cases will be displayed.

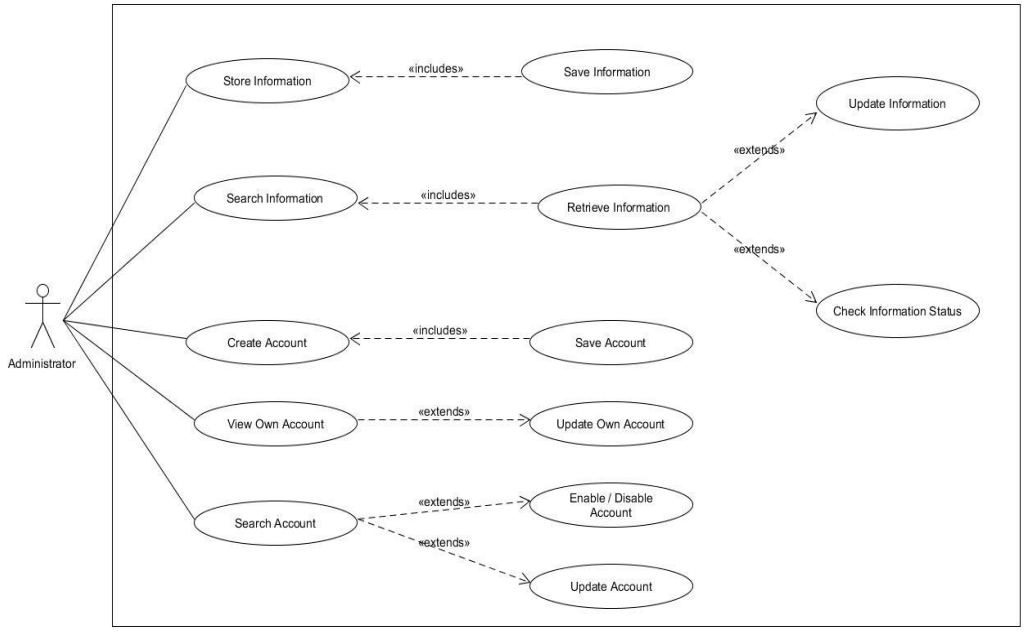

Figure 4. Use Case Diagram for Administrator Management Subsystem

Figure 4 shows use case diagram of Administrator Management Subsystem. The administrator can do store information which include save information, search information and extends update information, check information status then it includes retrieve information, create account which includes save account, view account which extends update own account and search account which extends enable/disable account and update account.

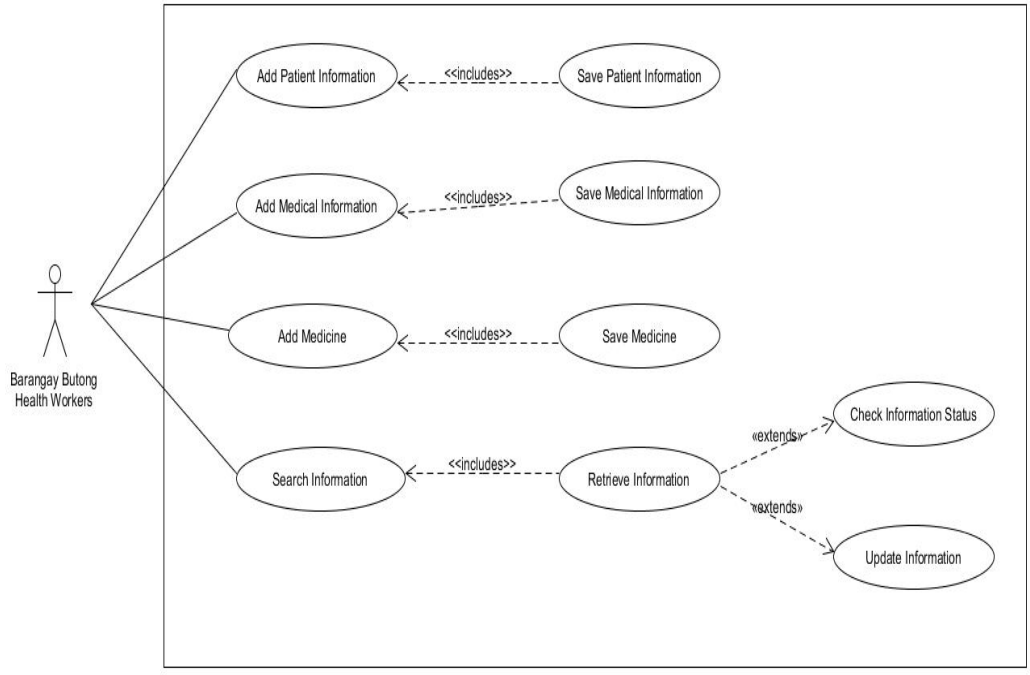

Figure 5. Use Case Diagram for Barangay Butong Health Workers Subsystem

Figure 5 shows use case diagram for Barangay Butong Health Workers Subsystem. The Barangay Butong Health Workers can add patient information which includes save patient information, add medical information which includes save medical information, add medicine it includes save medicine and search information which extends update information and check information status.

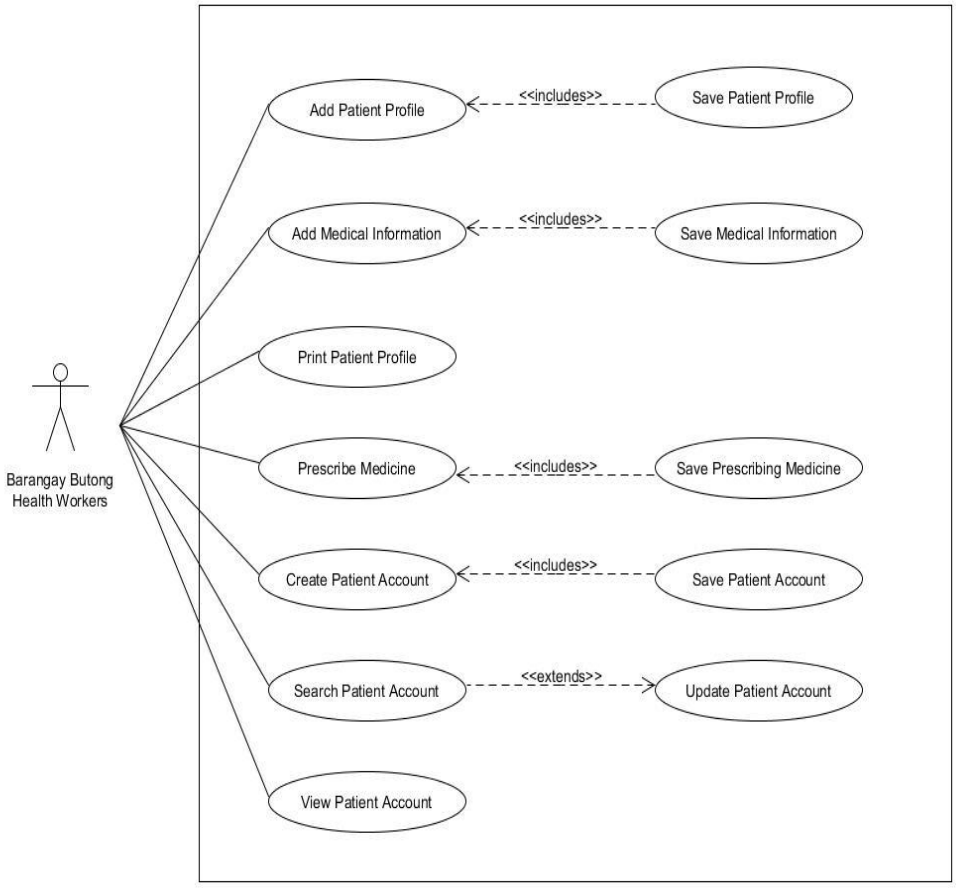

Figure 6. Use Case Diagram for Patient Management

Figure 6 shows use case diagram for Patient Management. The Barangay Butong Health Workers can add patient profile which includes save patient profile, add medical information which includes save medical information, print patient profile, prescribe medicine which includes save prescribing medicine, create patient account which includes save patient account, and search patient account which extends update patient account and view patient account which includes also display patient account.

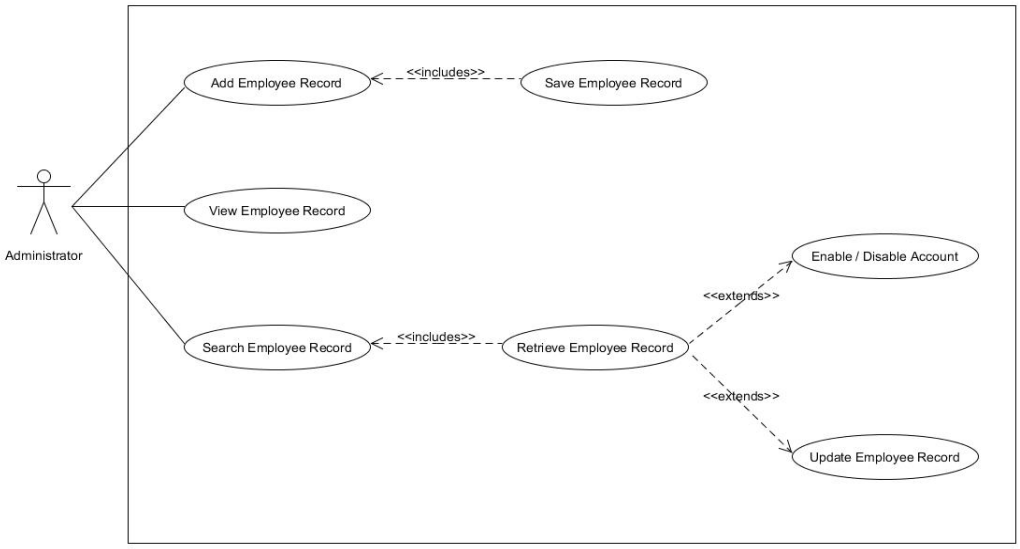

Figure 7. Use Case Diagram for Employee Record Management

Figure 7 shows use case diagram of Employee Record Management. The administrator can add employee record, which includes save employee records, view employee record and search

employee record which includes retrieve employee record and it extends update employee record and enable/disable account.

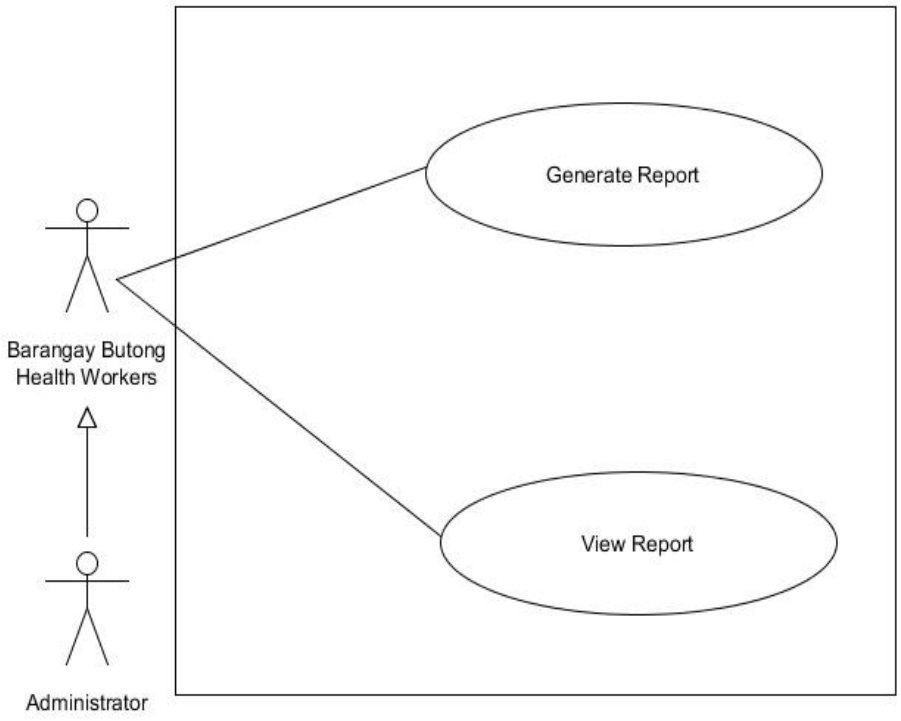

Figure 8. Use Case Diagram for Generating Reports

Figure 8 shows use case diagram for Generating Reports. As you can see Barangay Butong Health Workers and Admin can do generate report and viewing of reports.

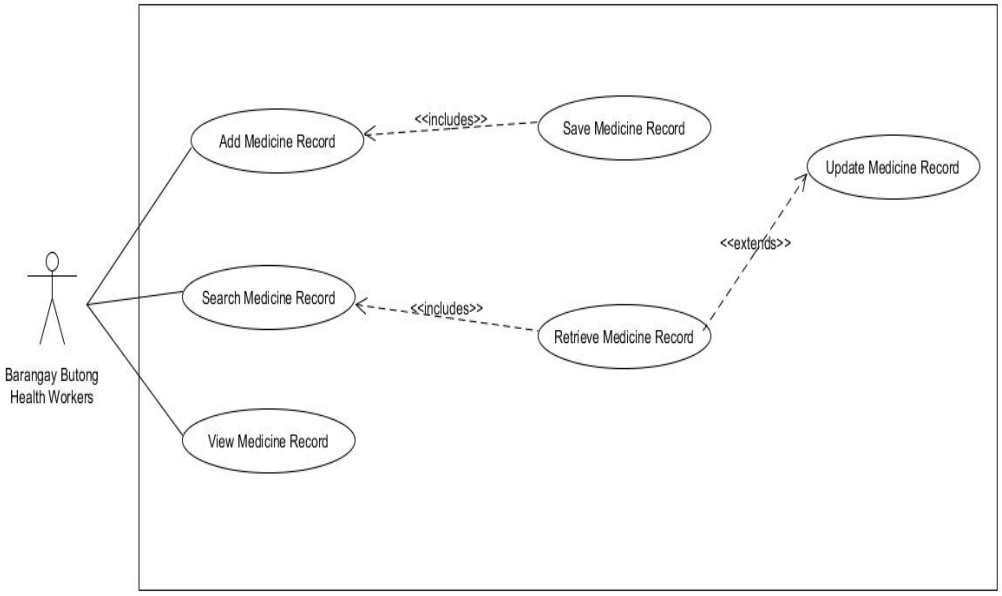

Figure 9. Use Case Diagram for Inventory of Medicine

Figure 9 shows use case diagram for Inventory of Medicine. The Barangay Butong Health Workers can add medicine record which includes save medicine record, search medicine record which extends update medicine record and retrieve medicine record, view medicine record which includes display medicine record.

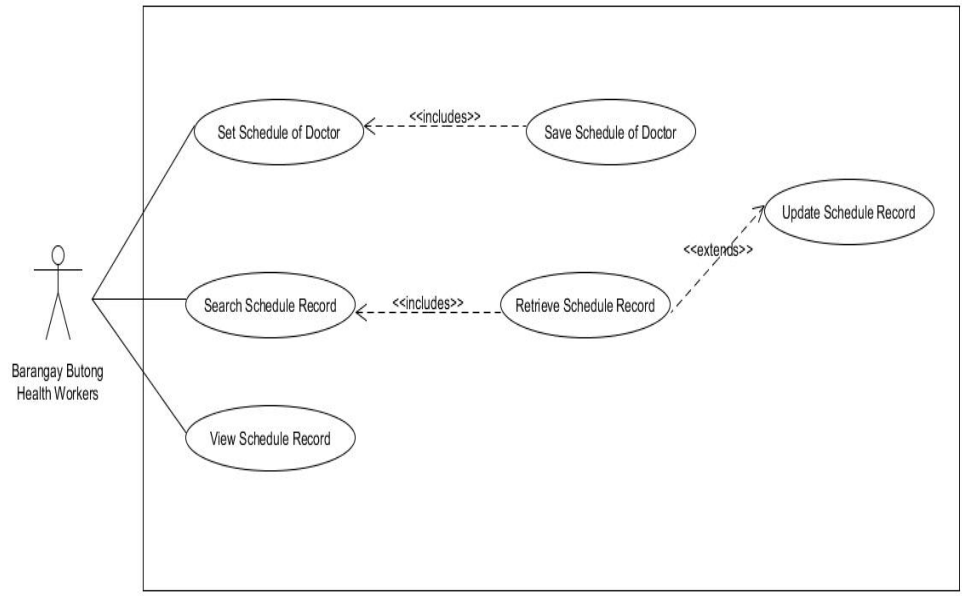

Figure 10. Use Diagram for Doctor Scheduling

Figure 10 shows use case diagram for Doctor Scheduling. The Barangay Butong Health Workers can set the schedule of doctor which includes save schedule of doctor, search schedule record which extends update schedule record and includes retrieve schedule record, view schedule record and includes display schedule record.

Activity Diagram

Activity diagrams is graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

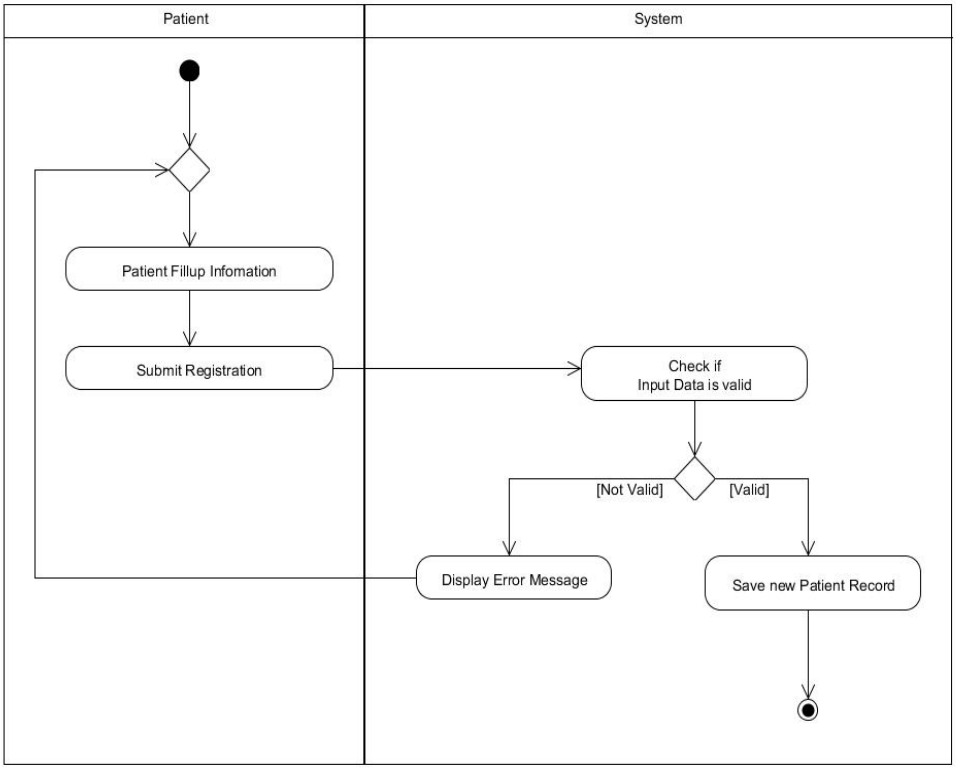

Figure 11. Activity Diagram for Patient Registration Form

Figure 11 shows an activity diagram for Patient Registration Form. The patient will do fill up information after the patient fill up the information the patient will submit the registration then the system will validate the input data of the patient if it is valid it will saved new patient record, if it is not valid the system will display error message.

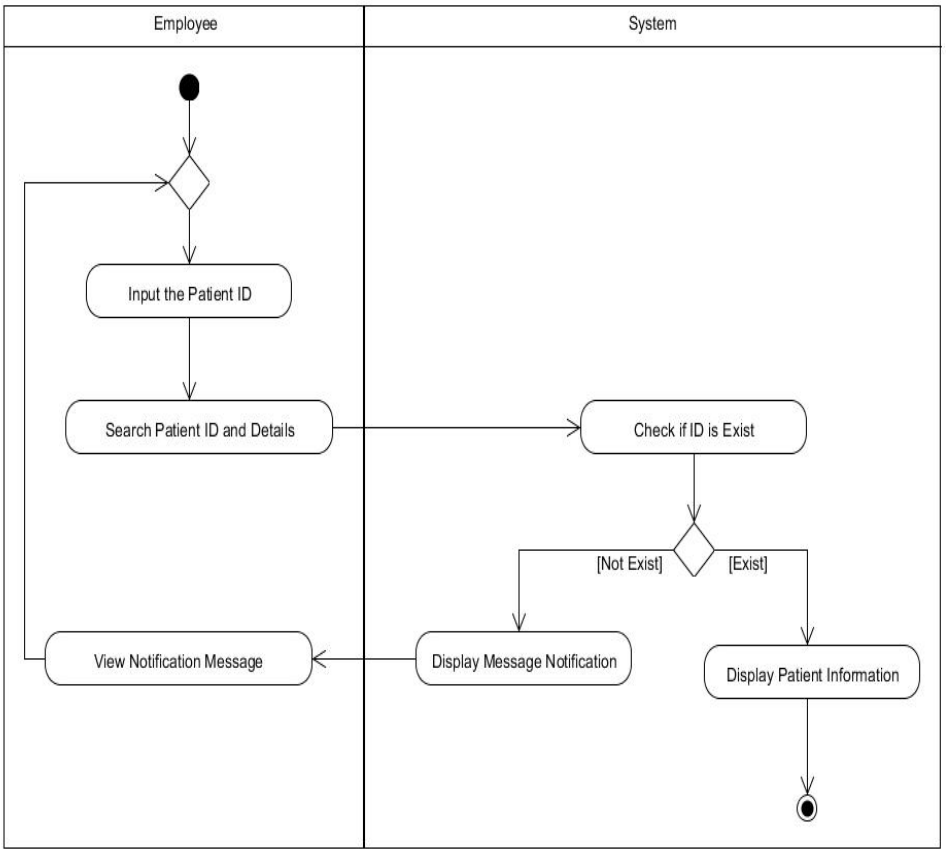

Figure 12. Activity Diagram for Searching Patient Profile

Figure 12 Shows an Activity Diagram of Searching Patient Profile. The user or staff input patient ID search the patient id details. The system will check if the patient ID exist if it is exist it display patient Information and the user or staff will able to view the patient information and if the search patient profile does not exist the system will display message notification

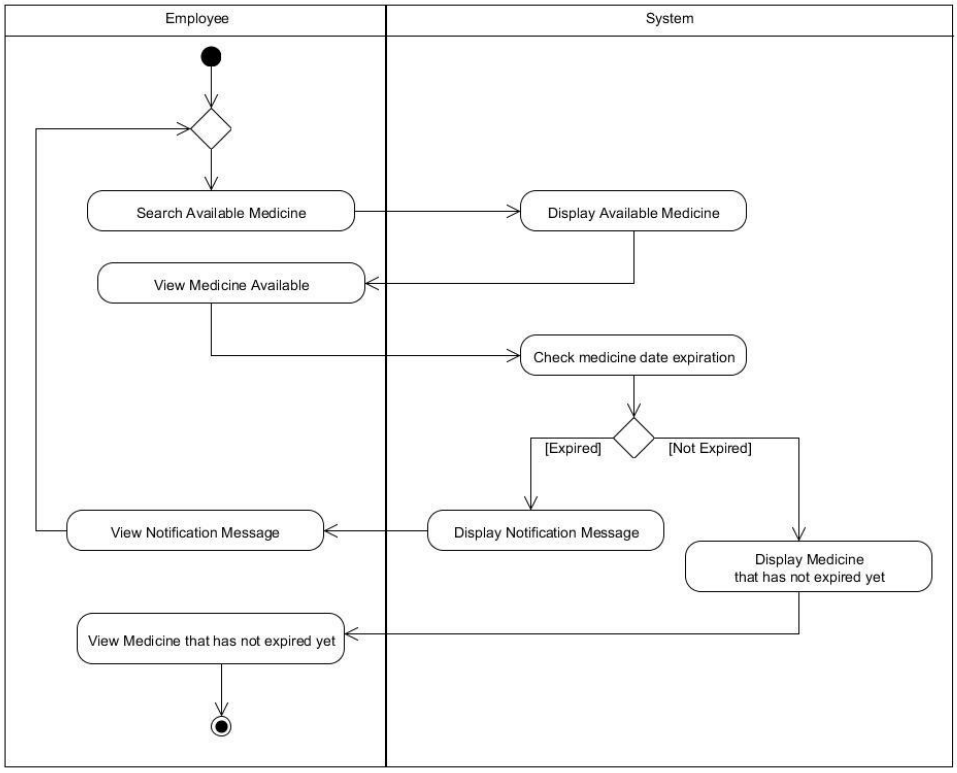

Figure 13. Activity Diagram for Searching Medicine

Figure 13 shows an activity diagram for Searching Medicine. The user or employee will search available medicine then the system will display available medicine and the user or staff can able to see the available medicine, next the system will check the medicine date expiration if it is not expired the system will display the medicine that is not expired yet and if the medicine is already expired the system will display notification message that the search medicine is expired.

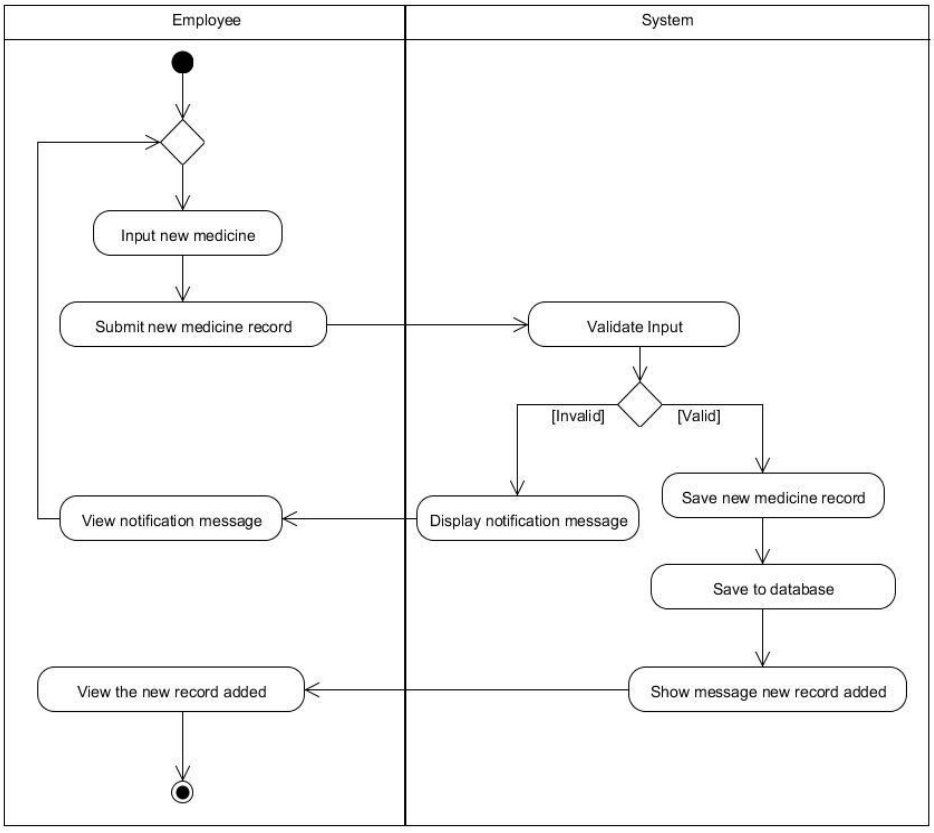

Figure 14. Activity Diagram for Adding Medicine Record

Figure 14 shows an activity diagram of Adding Medicine Record first the user will input new medicine record, submit new medicine, then save the new medicine record to the system then it automatically save to database. After that the system will display message successfully that the new record is added.

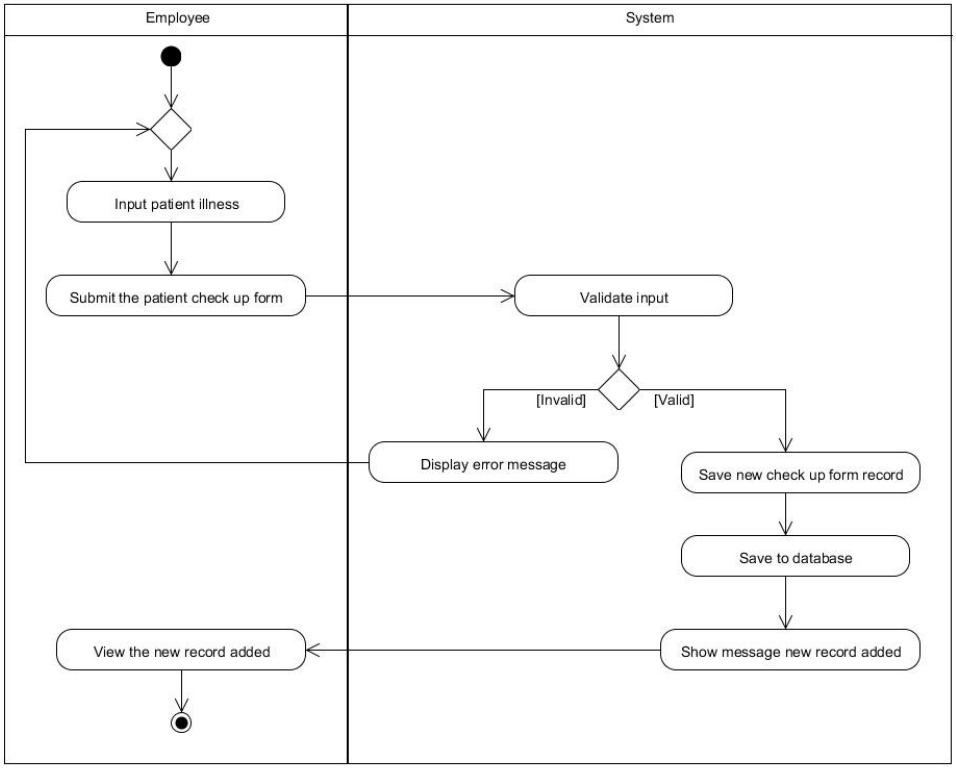

Figure 15. Activity Diagram for Creating Checkup Form

Figure 15 shows an Activity Diagram of Creating Checkup Form first the employee will input patient illness, submit the

patient checkup form, When the user submit the patient checkup form the it will save to our system save to database and display message notification that the new record is successfully added.

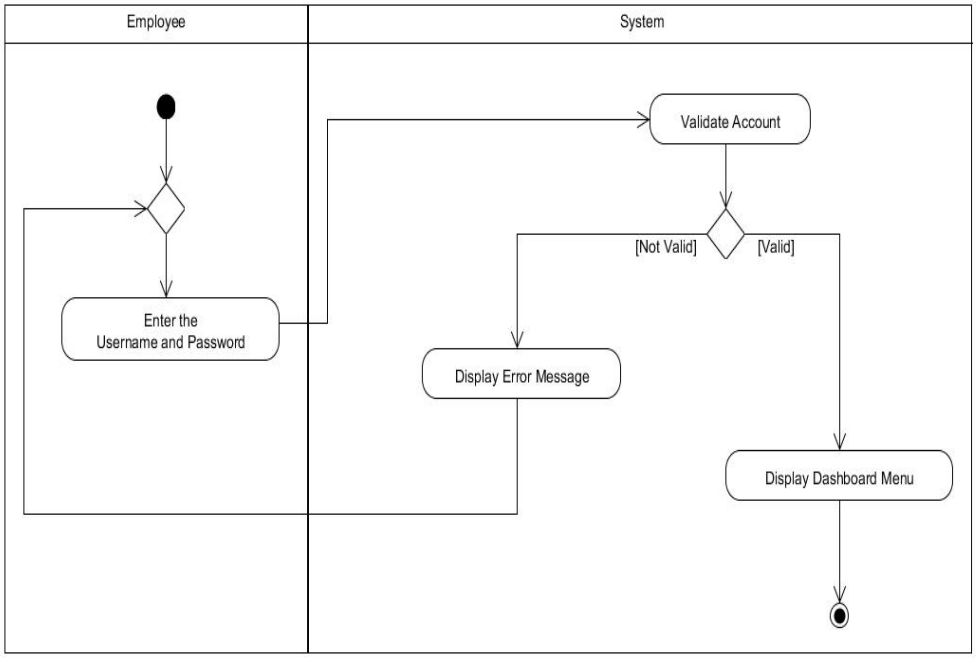

Figure 16. Activity Diagram for Login Subsystem

Figure 16 shows an Activity Diagram of Login Subsystem first the user will enter his/her username and password next the system will validate the account if the account is not valid it will display error message, if the account is valid the system will display Message that the account is successfully Login last it will display Dashboard Menu.

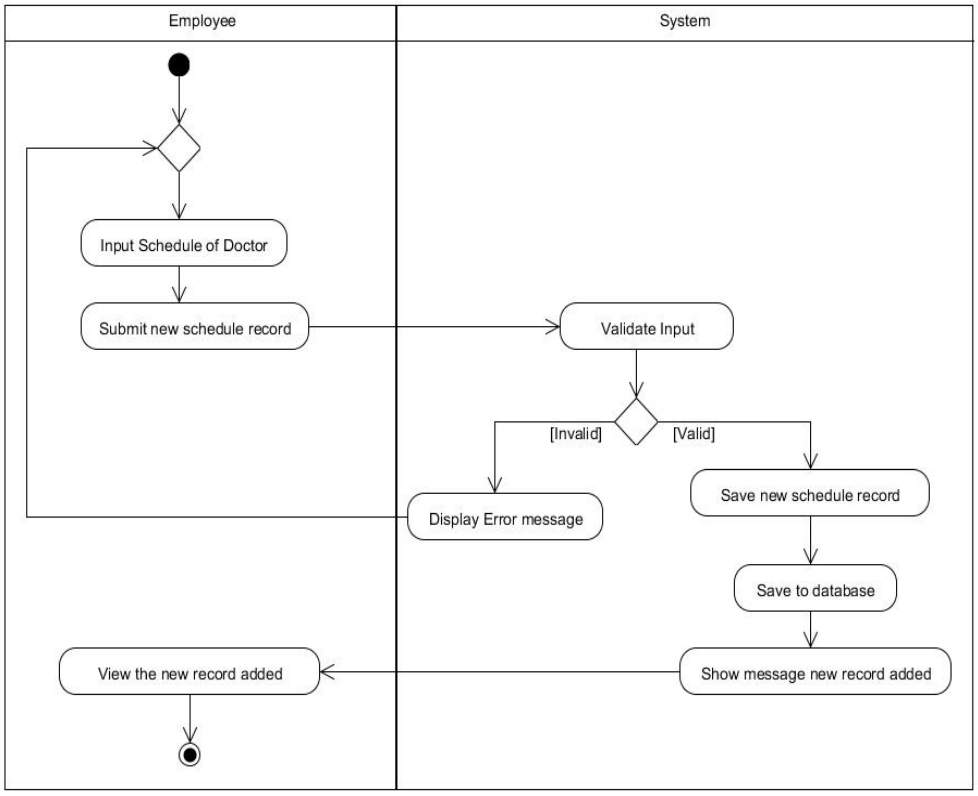

Figure 17. Activity Diagram for Doctor Scheduling

Figure 17 shows an activity diagram of Doctor Scheduling first the employee will input schedule of doctor, submit the new schedule record, When the user submit the schedule of doctor it will save to our system save to database and display message notification that the new record is successfully added.

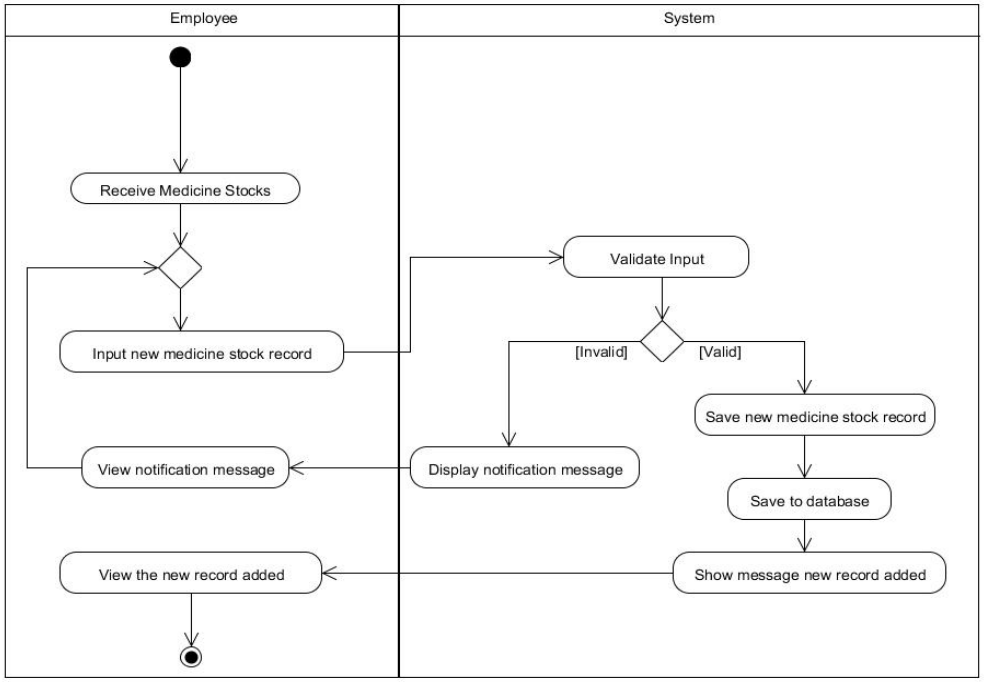

Figure 18. Activity Diagram for Receiving Medicine Stocks

Figure 18 shows an activity diagram of Receiving Medicine Stocks first the employee will receive medicine stocks, after that the employee will submit the new medicine stock record, When the user submit the medicine stock record it will save to our system save to database and display message notification that the new record is successfully added.

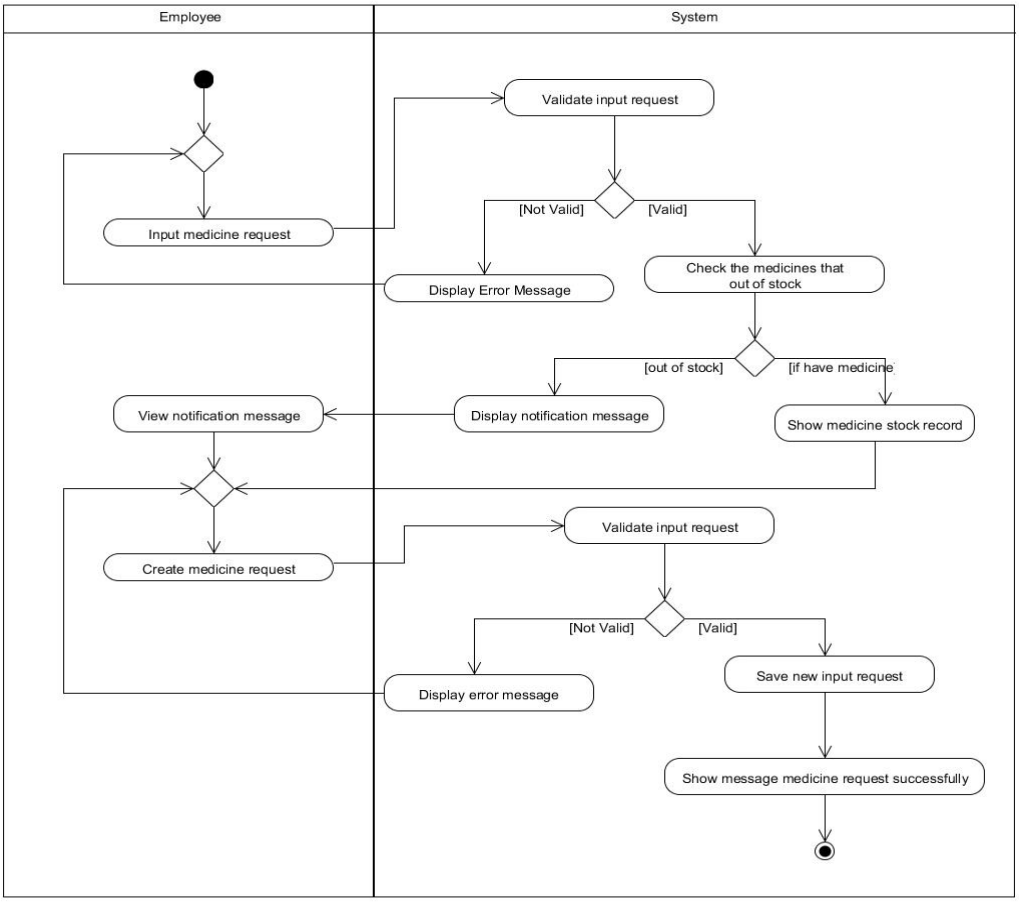

Figure 19. Activity Diagram for Creating Medicine Request

Figure 19 shows an activity diagram for Creating Medicine Request first the employee will create medicine request, after that the system will validate the input request if it is valid the system will check the medicines that out stock. If the medicines out of stock the system will display notification message. And if have medicine the system will show medicine stock record and turn to create medicine request. After creating medicine request the system

will validate again input request. If it is valid the system will save new input request and show message medicine request successfully. If is not valid the system will display error message and go back to create medicine request.

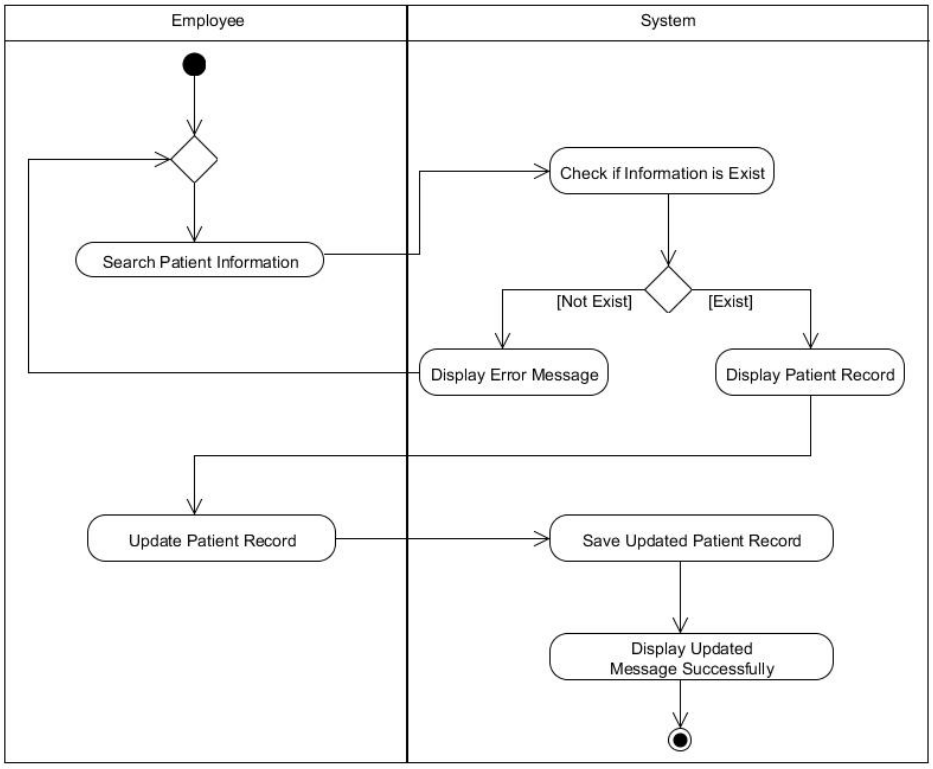

Figure 20. Activity Diagram for Updating Patient Record

Figure 20 shows an activity diagram for Updating Patient Record first the employee will search patient information then the system will check if information is exist the search information is exist the system will display the patient record then the employee

will able to update the search patient record and save updated patient record and last the system will display updated message successfully.

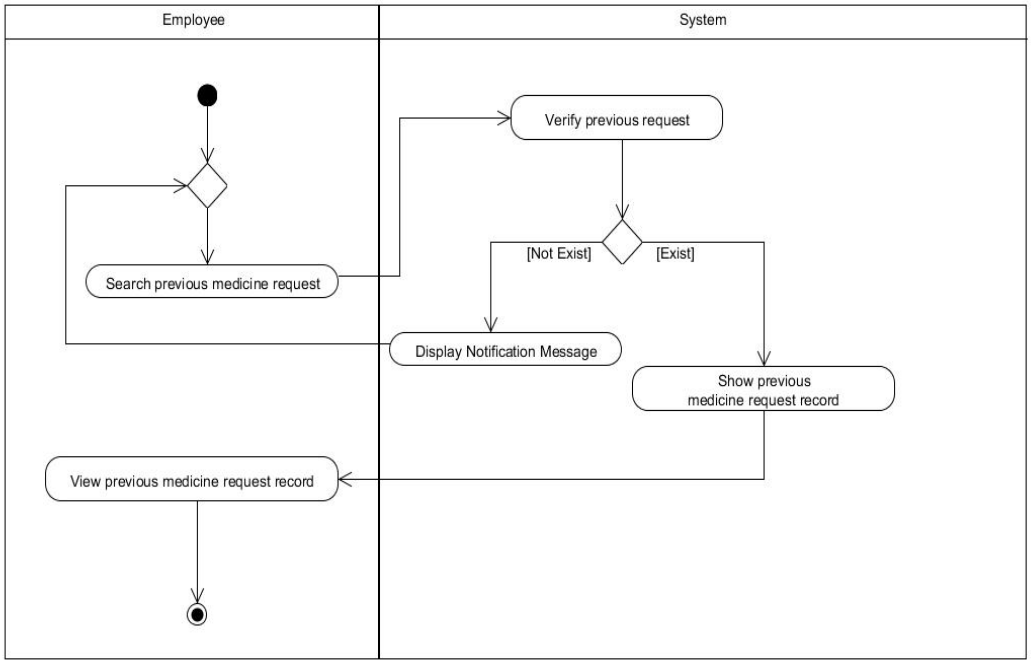

Figure 21. Activity Diagram for Searching Previous Medicine Request

Figure 21 shows an activity diagram for Searching Previous Medicine Request first the employee will search previous medicine request, after that the system will verify previous request. If is exist the system will show previous medicine request record and the employee will view the previous record for

checking the medicines record. And if is not exist the system will show notification message and go back to search previous medicine request.

System Development

The researchers will develop an Online Information System that will provide the information needed to help the patient as well as the Barangay Health Center Information System for Barangay Butong City of Cabuyao Laguna for the system design. The researchers use Sublime Text and XAMPP server for the proposed system to be more efficient and effective.

The researchers will use Unified Process for the software development of the proposed system. It helps the researchers to build software for better system and it is a popular software development process that is widely used. The researchers use the following methods.

Inception Phase. In this phase, the researchers determine the requirements, plan, goals as well as the scopes and limitations of the system. The Inception phase lets the researchers to further study the system and be able to understand the purpose of the system. It involves human, software and hardware requirements to determine the best plan to implement for developing the system.

Elaboration Phase. After determining the requirements and the plan, as well as the problems of the system, the system will be approved considering that it meets the requirement needed. The researchers will then proceed to the development.

Construction Phase. This phase involves the development of the system. The plan is translated into program code to add functionalities from the given design and an enhancement as it is effective for the end-user.

Transition Phase. This phase includes the validation and the researchers need to check the concept of the system. This will satisfy the requirements of the users more accurately and effectively. This phase also includes the installation of the system, once the system has been tested and meets the requirements.

CHAPTER IV

RESULTS AND DISCUSSIONS

This chapter contains the overall structure, process and procedure of the developed system. Among the topics included in the discussion are the problems encountered, system structure, system screenshots, system requirement and installation, testing procedure, system evaluation, assessment of the user and assessment of web developer experts.

Problems Encountered by the Barangay Butong Health Center in City of Cabuyao Laguna.

Barangay Butong Health Center, City of Cabuyao Laguna started of getting patient's information and recording it in a manual way. The Barangay Health Center workers keeps their records in a log book by writing it manually. The same way in scheduling the duty of the doctor and handling the inventory of medicine.

The Barangay health center cannot easily manage the filing of patient’s records, scheduling of the doctor and inventory of medicines. The health workers cannot easily find each patient's records during and previous checkup. Also, it is not the safest way by keeping or storing the records, as well

as by searching it on their file because it may be time consuming.

The staff of the barangay health center cannot even fix the scheduling the doctor's duty. The health workers sometimes cannot be sure in the availability of the doctor in daily checkup of different patients in different schedule. Sometimes it causes conflict because of the doctor’s other commitment in some hospital.

In handling the inventory of the medicines, it would take too much time and consume a lot of paper works by looking and recording the stocks, the brand, and dosage, expiry date, in and out of items, the remaining and the stocks to be needed. All of these should be recorded accurately in their files.

In generating patient records, the health center workers write manually all the patient’s information. After that, the processed files must be put on the filing cabinet. If the patient wants to update their information, the health center workers get the records of the patient from the filing cabinet, the problem they encountered are having difficulty in returning the files manually and it may be time consuming for them.

System Structure

The proposed system was developed using Sublime text editor in coding the system. The researchers used mainly AJAX, PHP and JavaScript for the system's functionalities. For the database, MySQL was used since it has the ability to handle lots of data. This was used for storing all the data needed by the user.

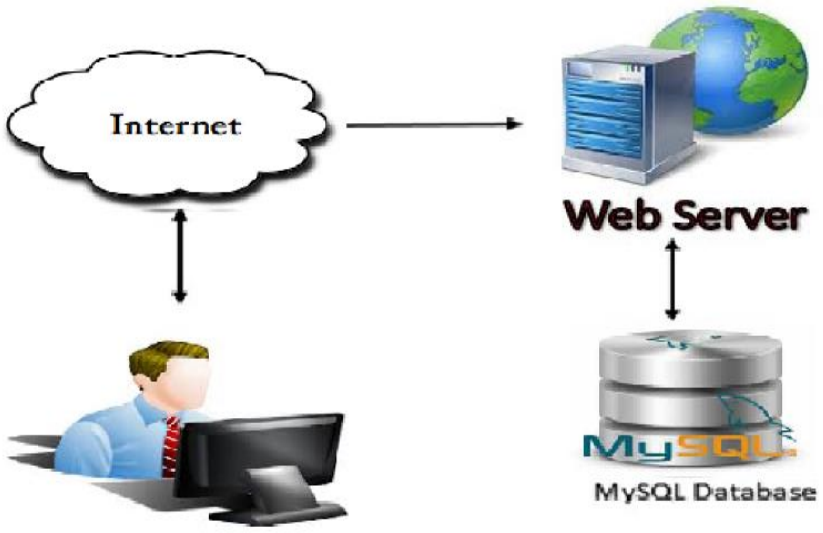

Figure 22. System Structure of Online Information System

Figure 22 shows the system structure of the proposed online application. The online application went to web hosting so that it will be available in the internet. The user can access the web server

through internet, the web server provides the web services of the web application. The web server accesses the database and throw the data needed back to the user.

Site Map

Functional Hierarchy Diagram enables you to model organizational functions in your business in a hierarchical structure (International Business Machines). This section shows the functional hierarchy of the An Online Barangay Health Center Information System for Barangay Butong City of Cabuyao Laguna.

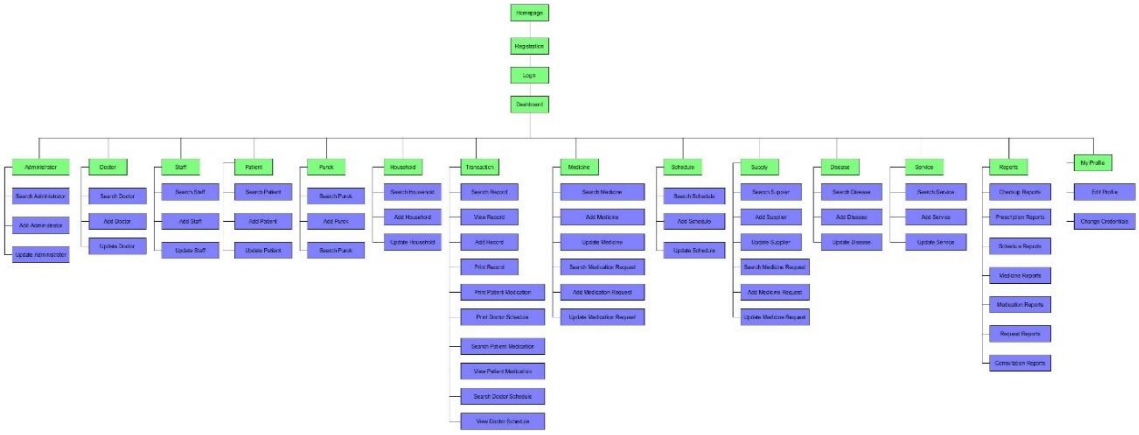

Figure 23. Site Map for Administrator Page

Figure 23 shows the features of the administrator management including the dash board, managing the administrator, doctor, staff, patient, purok household, transaction, medicine,

schedule, supply, disease, service, reports and my account. And the figure showed the activities that could be done by the administrator.

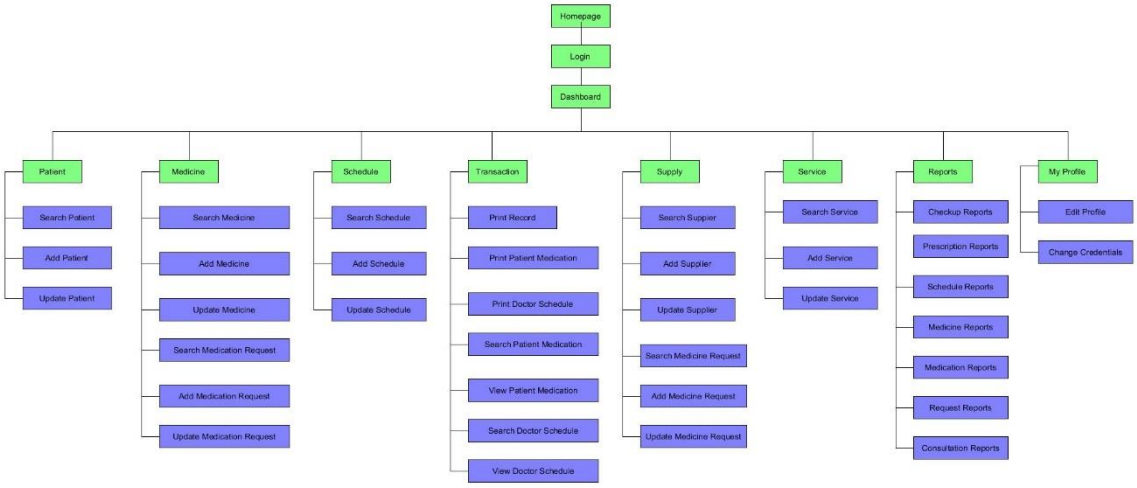

Figure 24. Site map for Staff Page

Figure 24 shows the features of the staff management including the dash board, managing the patient, medicine, schedule, transaction, supply, service to the patient, reports and my profile. And the figure showed the activities that could be done by the staff.

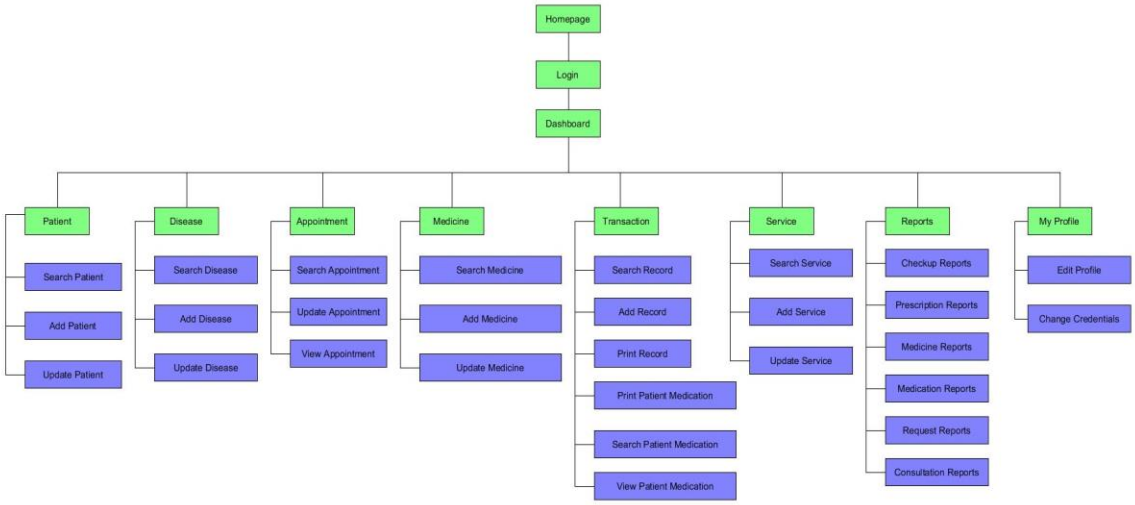

Figure 25. Sitemap for Doctor Page

Figure 25 shows the features of the doctor management including the dash board, managing the patient, disease, appointment, medicine, transaction, service to the patient, reports and my profile. And the figure showed the activities that could be done by the doctor.

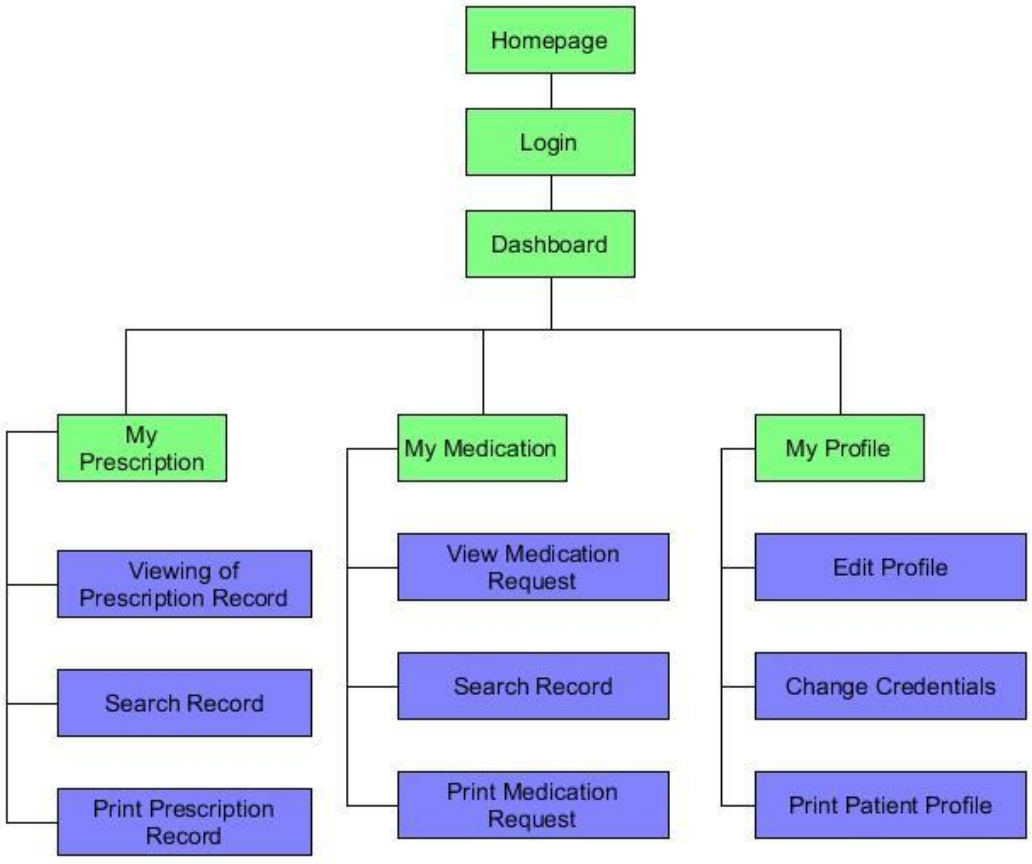

Figure 26. Sitemap for Patient Page

Figure 26 shows the features of the patient page including the dash board, my prescription, my medication and my. And the figure showed the activities that could be done by the patient.

Features of the Proposed System

The following figures showed the screenshots of the An Online Barangay Health Center Information System for Barangay Butong City of Cabuyao Laguna. It also displayed the output needed to be shown to the user. The structure of the system was reflected on the startup execution of the system and its Main menus. Upon the execution of the system, the login window and the main menu form would appear.

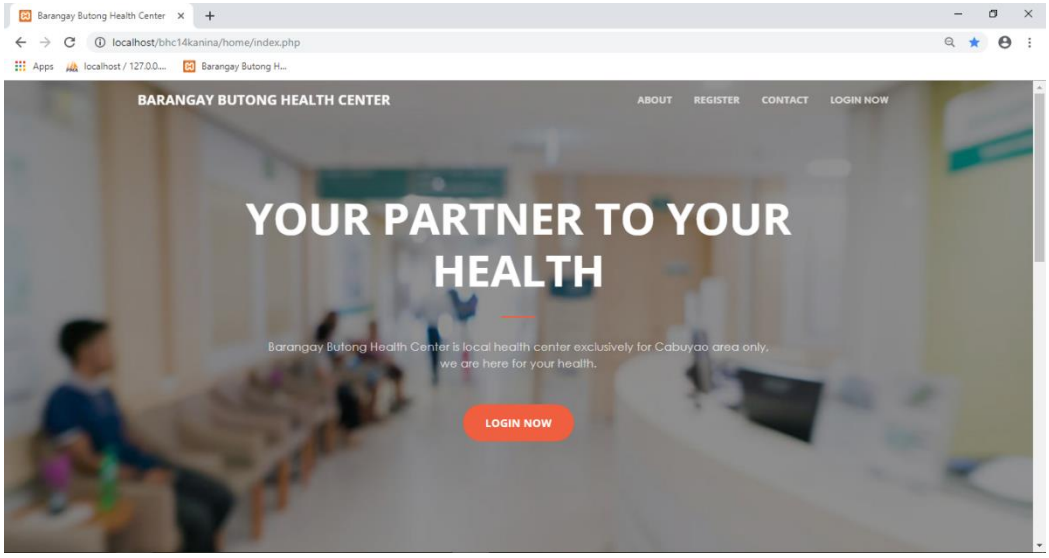

Figure 27. Homepage of the Proposed System

Figure 27 shows the homepage of the proposed system where in the user can see such as About, Register, Contact and Login.

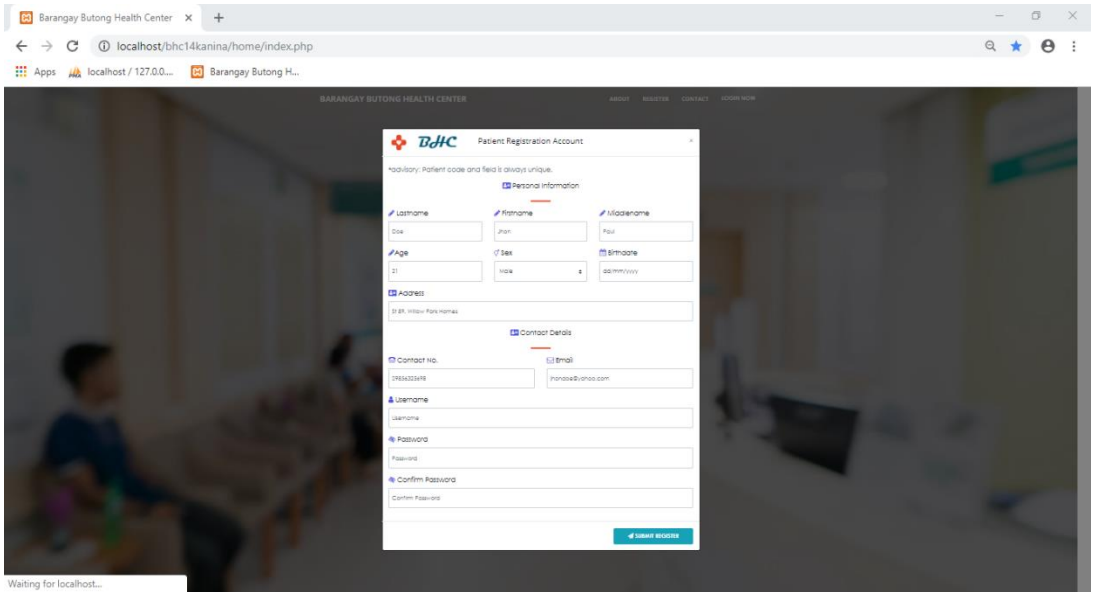

Figure 28. Patient Registration

Figure 28 shows the patient registration form where in the patient can register in order to access the website.

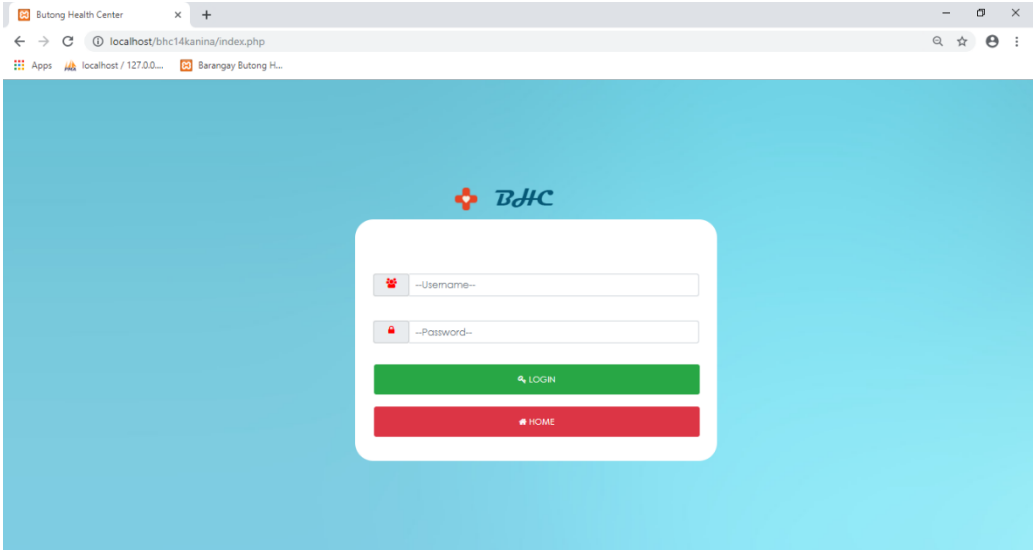

Figure 29. Login Page

Figure 29 shows the Login Page of the proposed system wherein authorize person can only access like administrator, staff of the barangay butong health center, doctor and patient who inquire.

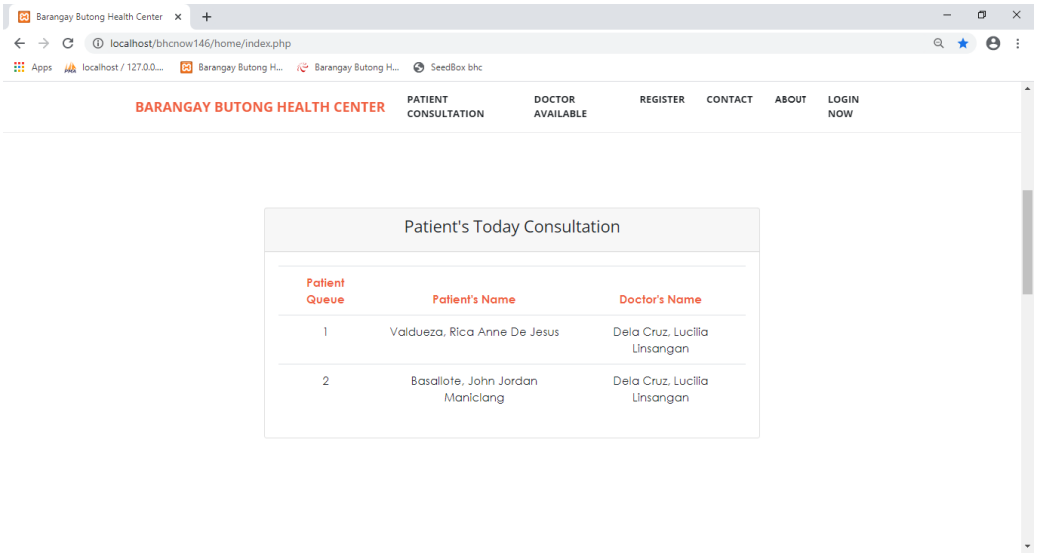

Figure 30. Patient Consultation Posting

Figure 30 shows the posting of Patient Consultation for that day and the doctor assign to each patient.

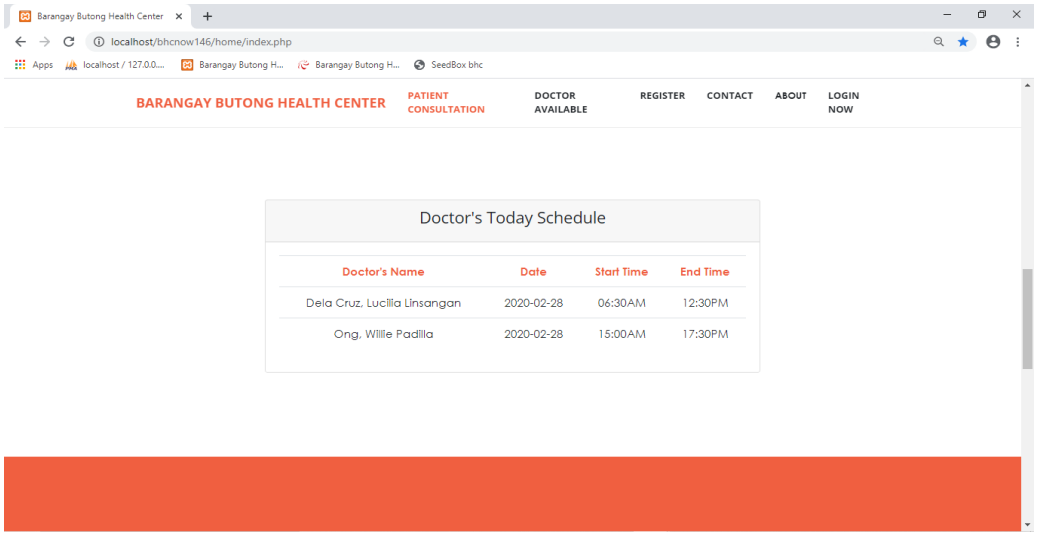

Figure 31. Doctor Schedule Posting

Figure 31 shows the Schedule of every doctor available on that day and their respective time.

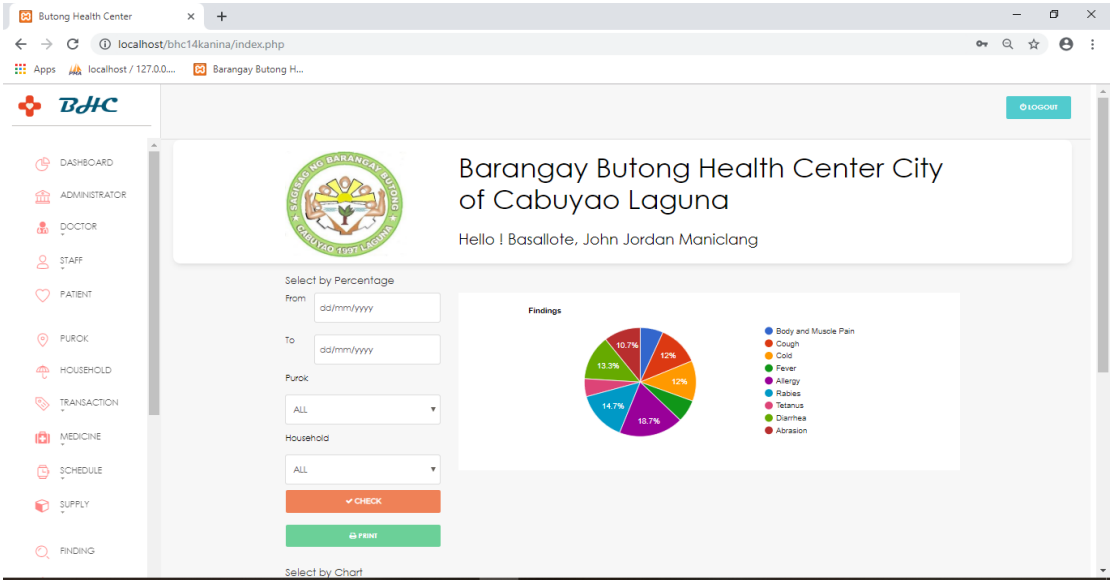

Figure 32. Dashboard of the Proposed System

Figure 32 shows the Dashboard of An Online Barangay Health Center Information System for Barangay Butong City of Cabuyao Laguna wherein the Administrator can access all the information.

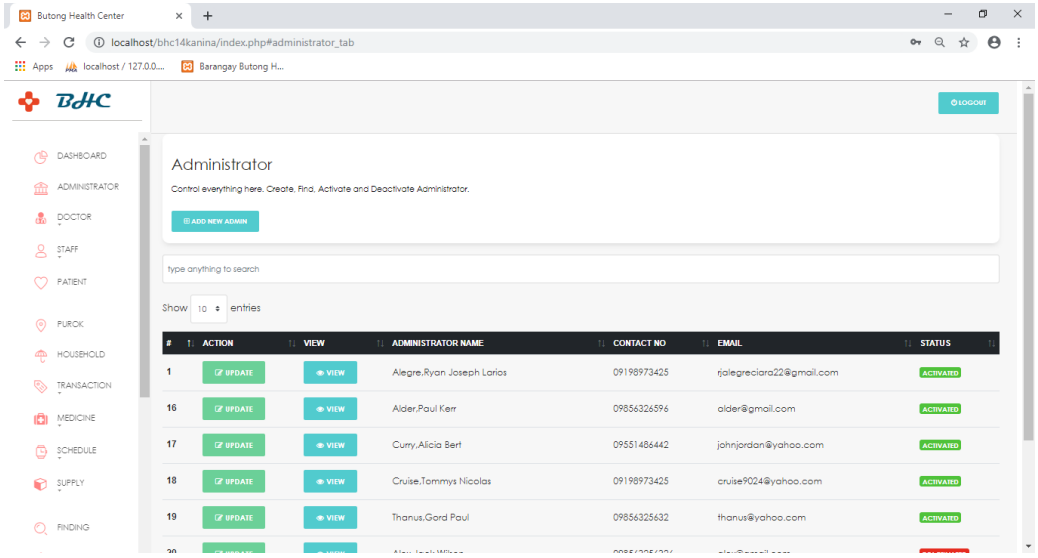

Figure 33. Managing Administrator

Figure 33 shows the managing of Administrator where in the Super Admin can search add update and view other admin

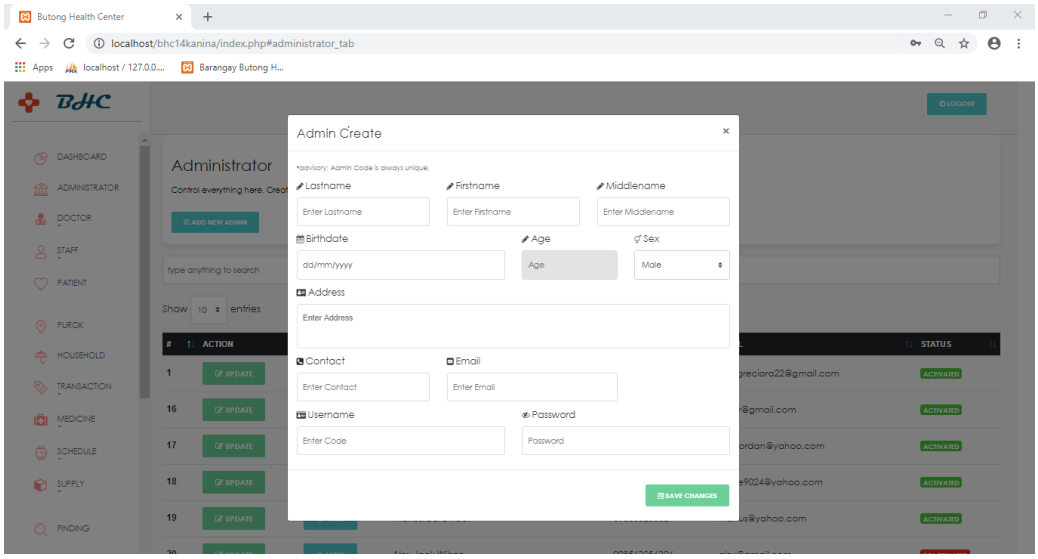

Figure 34. Administrator Create

Figure 34 shows the Administrator Create form wherein the administrator required to fill out all the information given in order to create another admin.

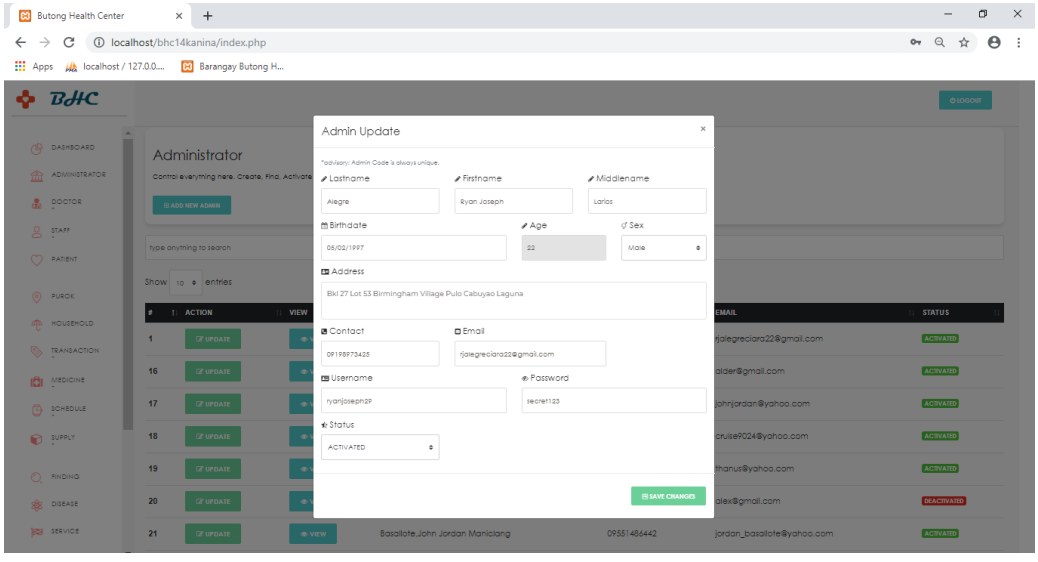

Figure 35. Administrator Update

Figure 35 shows the Administrator Update form wherein the administrator can update his/her own information.

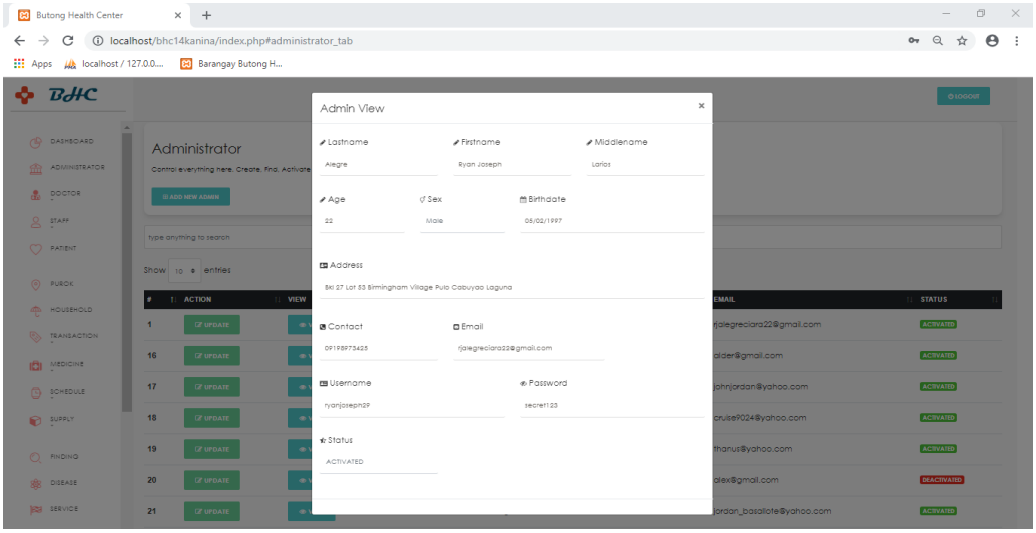

Figure 36. Administrator View

Figure 36 shows the Administrator View form wherein the administrator can view his/her account.

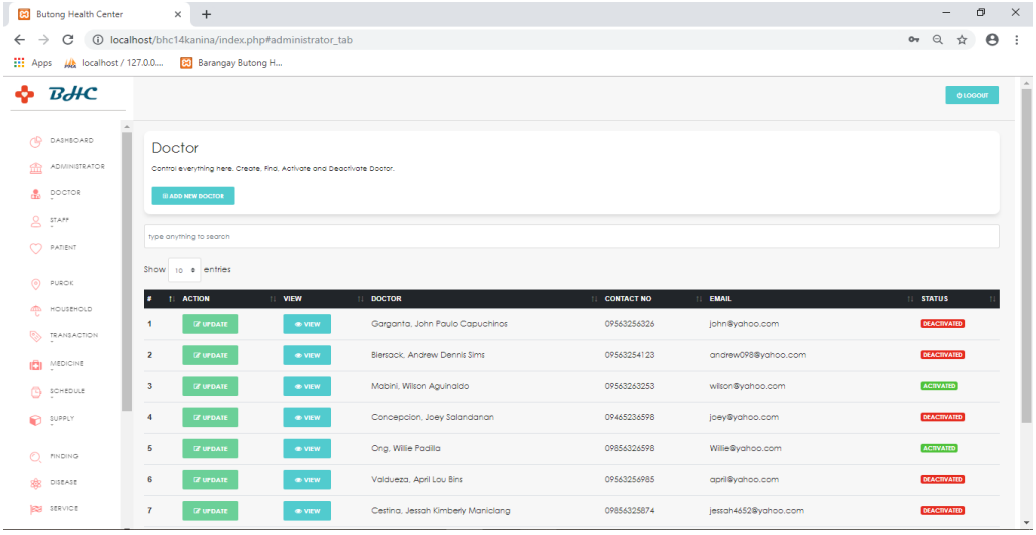

Figure 37. Managing Doctor

Figure 37 shows the Managing of Doctor wherein the administrator can add, update, view and search doctor.

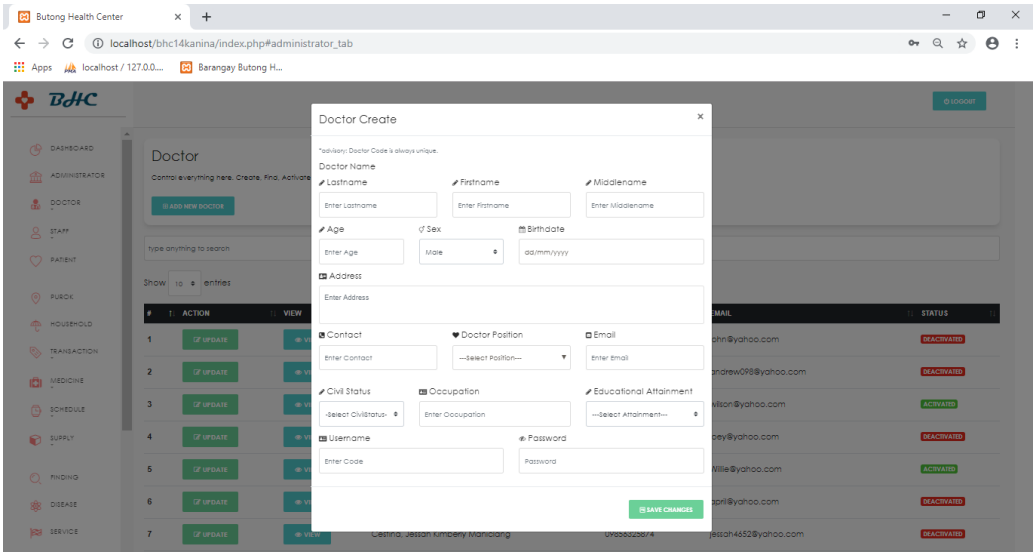

Figure 38. Doctor Create

Figure 38 shows the Doctor Create Form wherein the administrator required to fill up all the information given in order to create doctor.

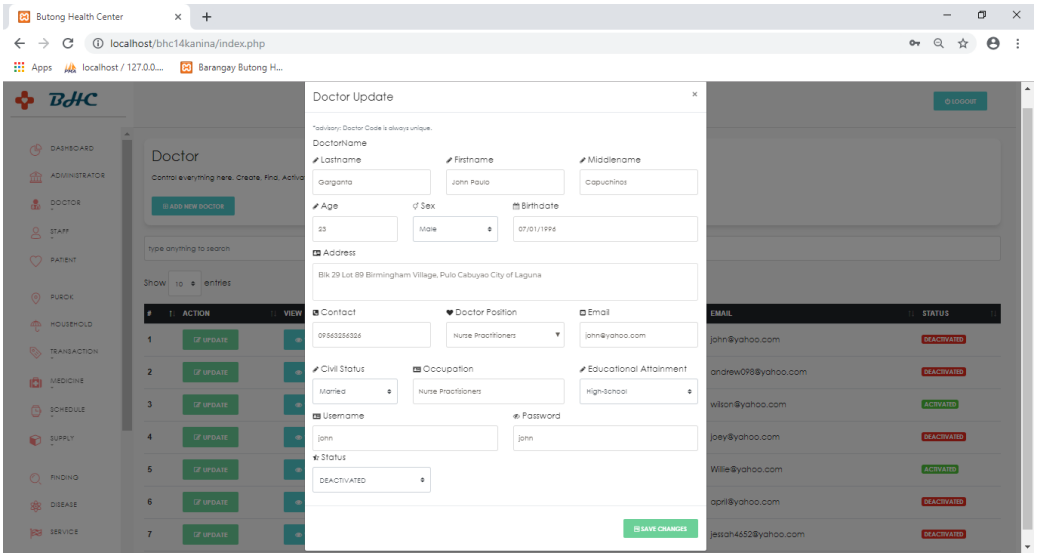

Figure 39. Doctor Update

Figure 39 shows the Doctor Update Form wherein the administrator can update the information of the doctor.

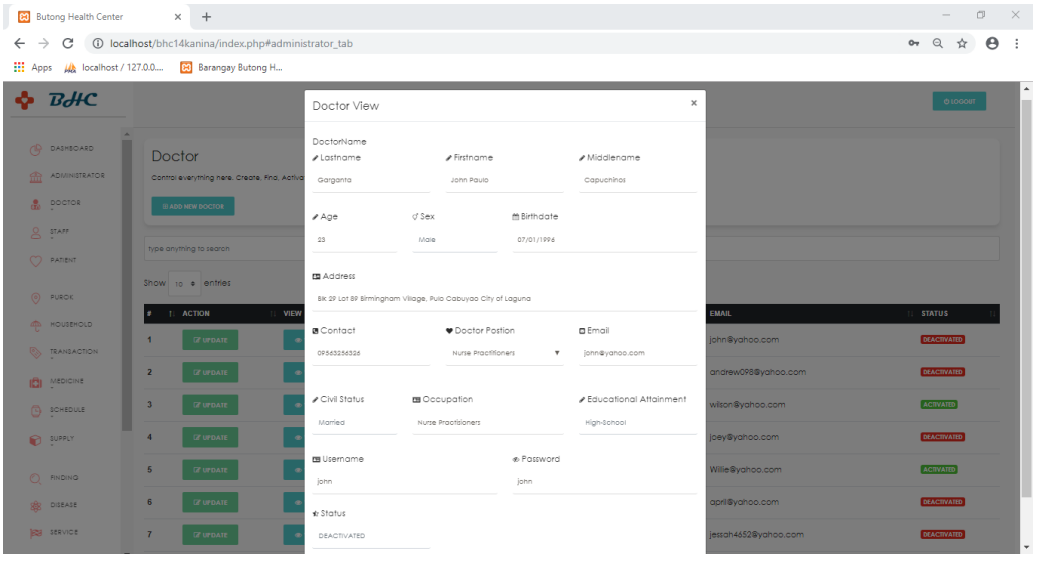

Figure 40. Doctor View

Figure 40 shows the Doctor View form wherein the administrator can view doctor information.

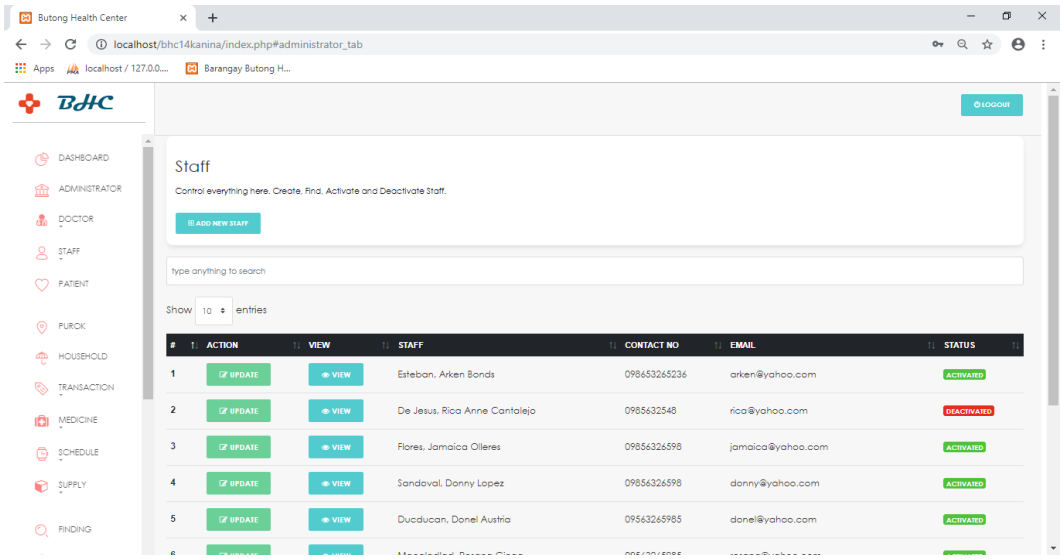

Figure 41. Managing Staff of the Barangay Butong Health Center

Figure 41 shows the Managing of Staff wherein the administrator can add, update, view and search the information of staff.

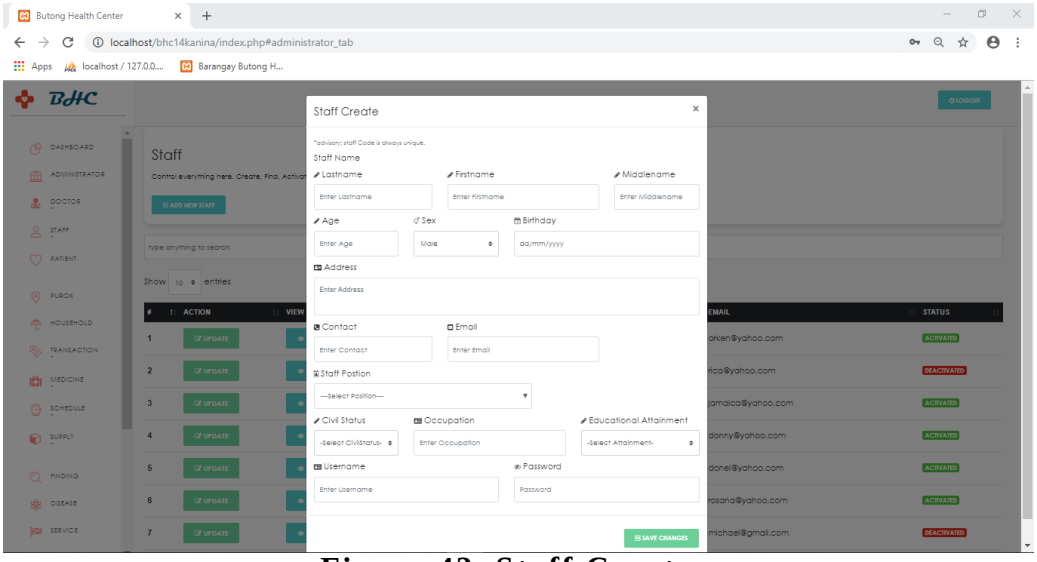

Figure 42. Staff Create

Figure 42 shows the Staff Create Form wherein the administrator required to fill up all the information given in order to create staff.

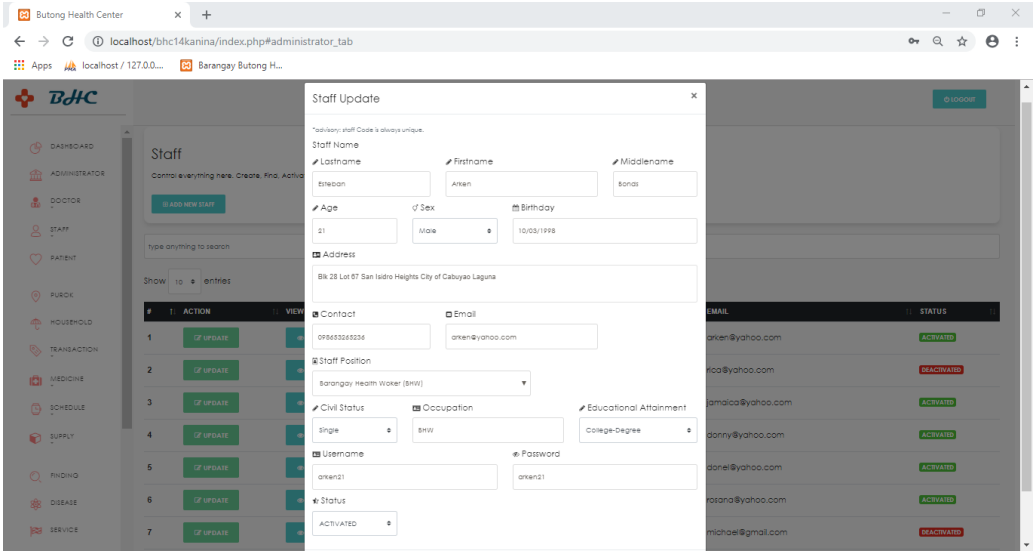

Figure 43. Staff Update

Figure 43 shows the Staff Update form wherein the administrator can update staff information.

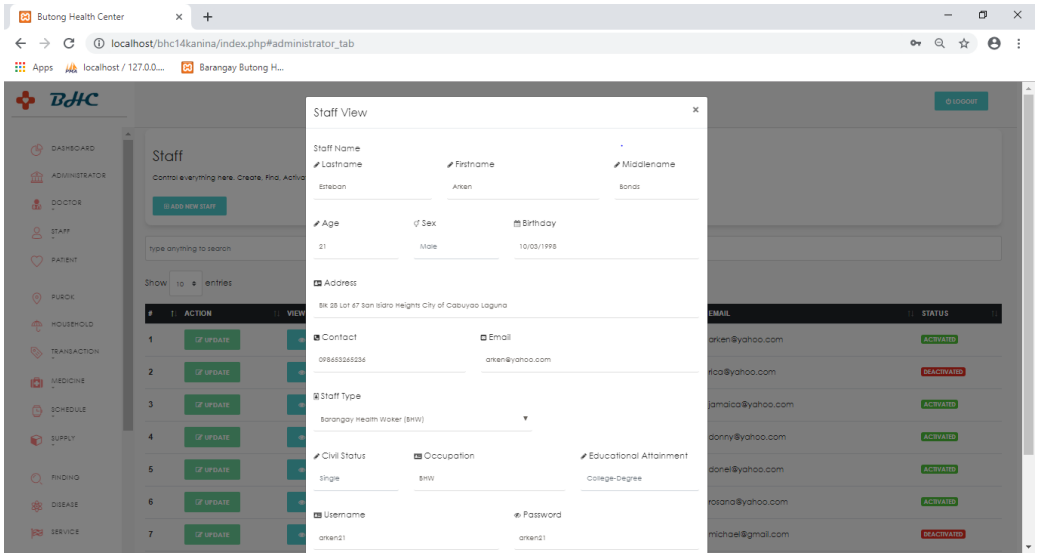

Figure 44. Staff View

Figure 44 shows the Staff View form wherein the administrator can view staff information.

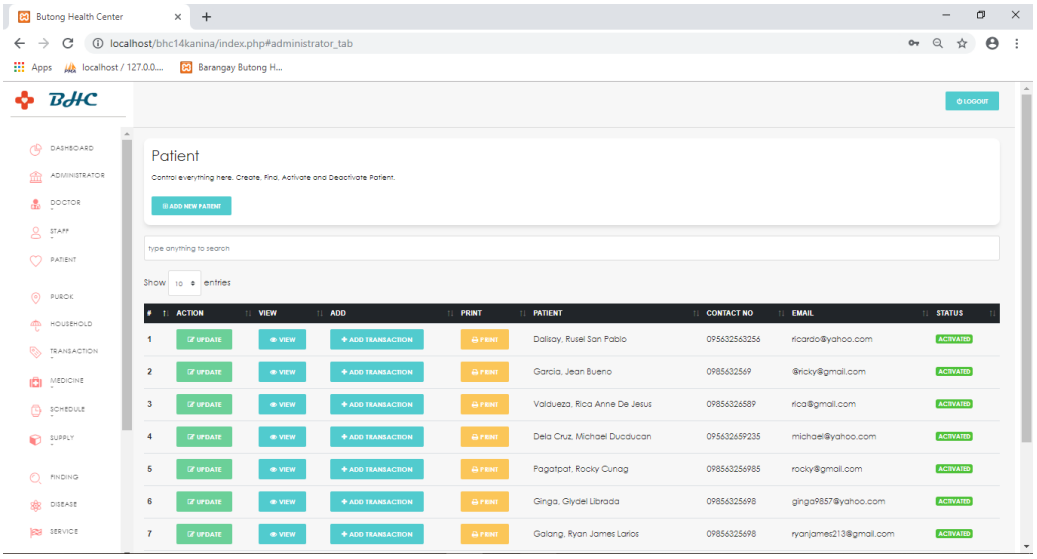

Figure 45. Managing Patient

Figure 45 shows the Managing of Patient wherein the administrator can add, update, view, add transaction, and search patient.

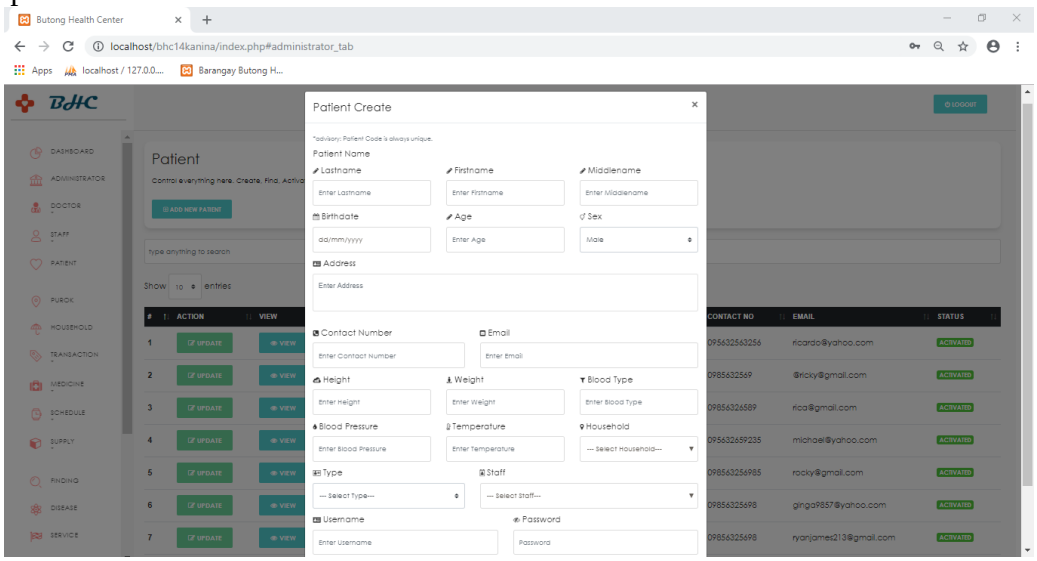

Figure 46. Patient Create

Figure 46 shows the Patient Create form wherein the administrator required to fill up all the information given in order to create Patient.

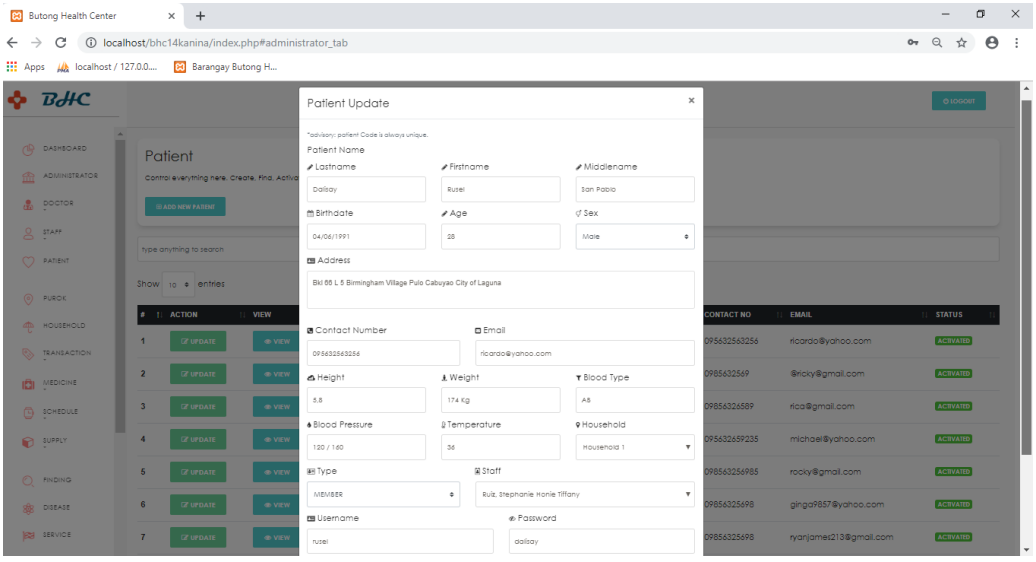

Figure 47. Patient Update

Figure 47 shows the Patient Update form wherein the administrator can update patient information.

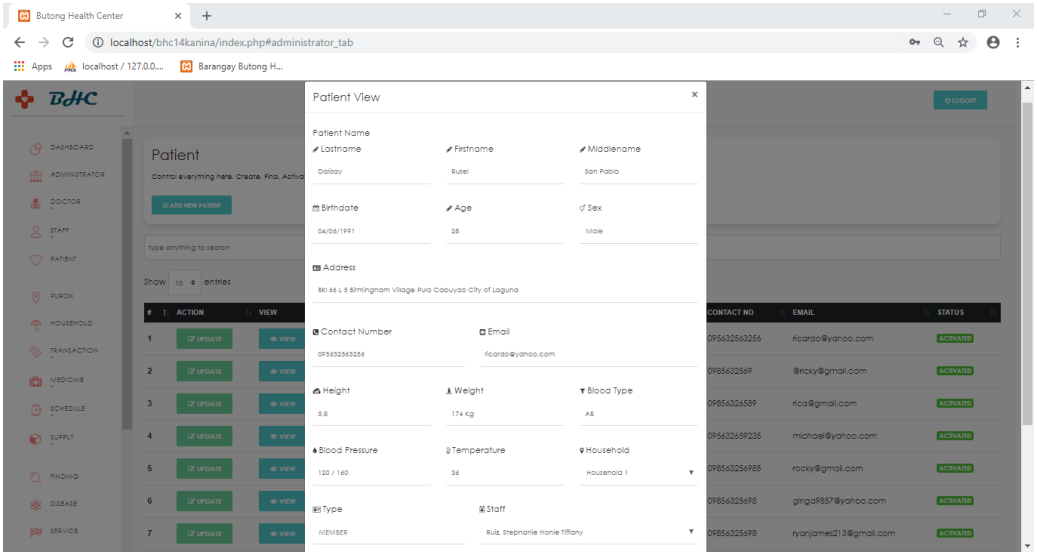

Figure 48. Patient View

Figure 48 shows the Patient View form wherein the administrator can view patient information.

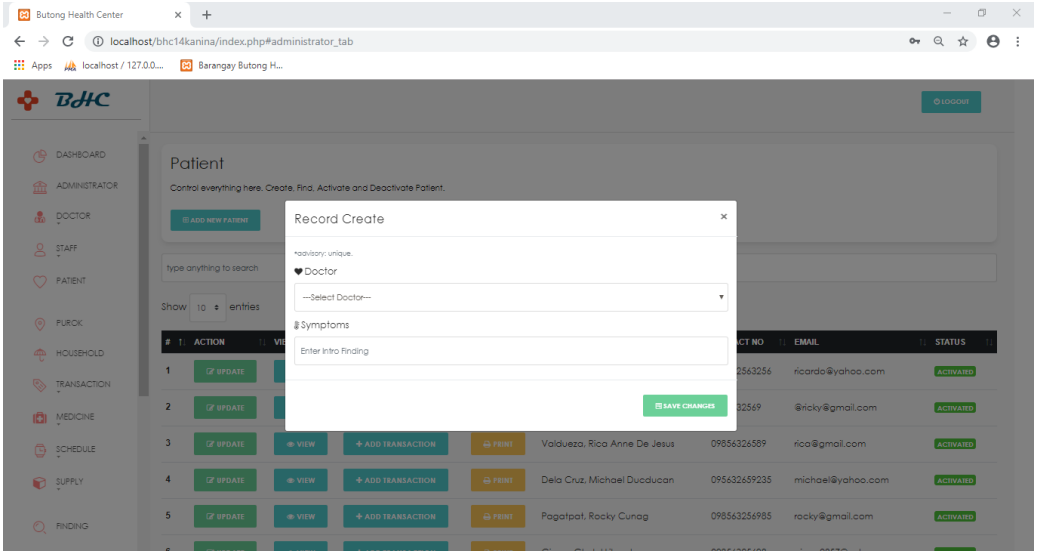

Figure 49. Patient Transaction Record Create

Figure 49 shows the Record Create form wherein the administrator can select doctor for that patient then fill out the symptoms of the patient and then click the save changes button.

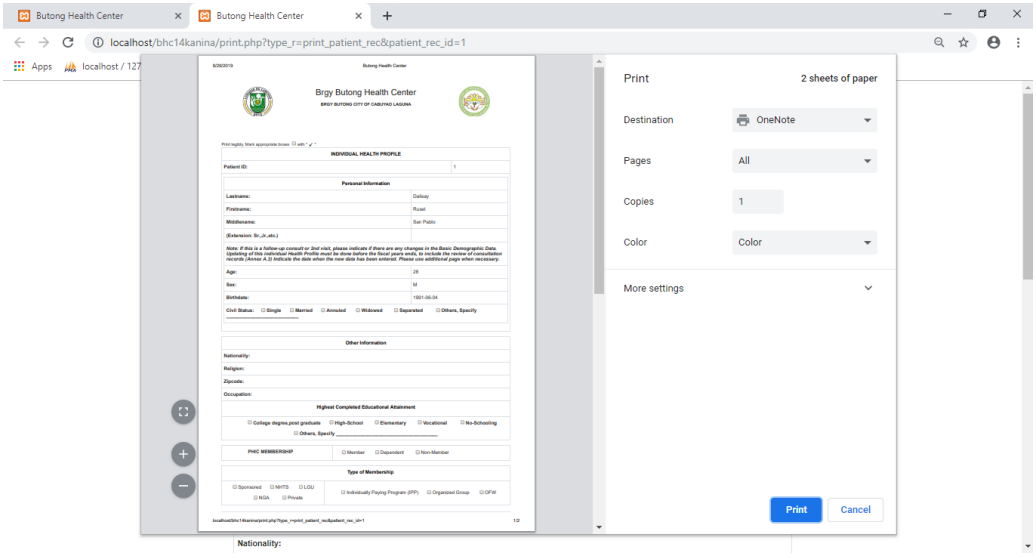

Figure 50. Patient Profile Print

Figure 50 shows the Patient Profile print tab wherein the administrator can print Patient Profile.

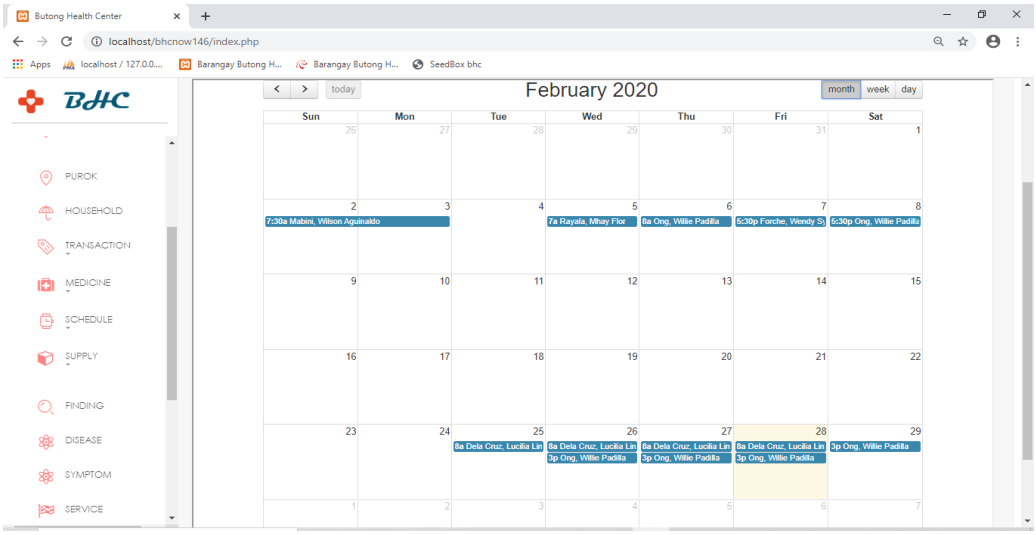

Figure 51. Scheduling by Month

Figure 51 shows the scheduling by month wherein the admin can see who doctor is available in that particular month.

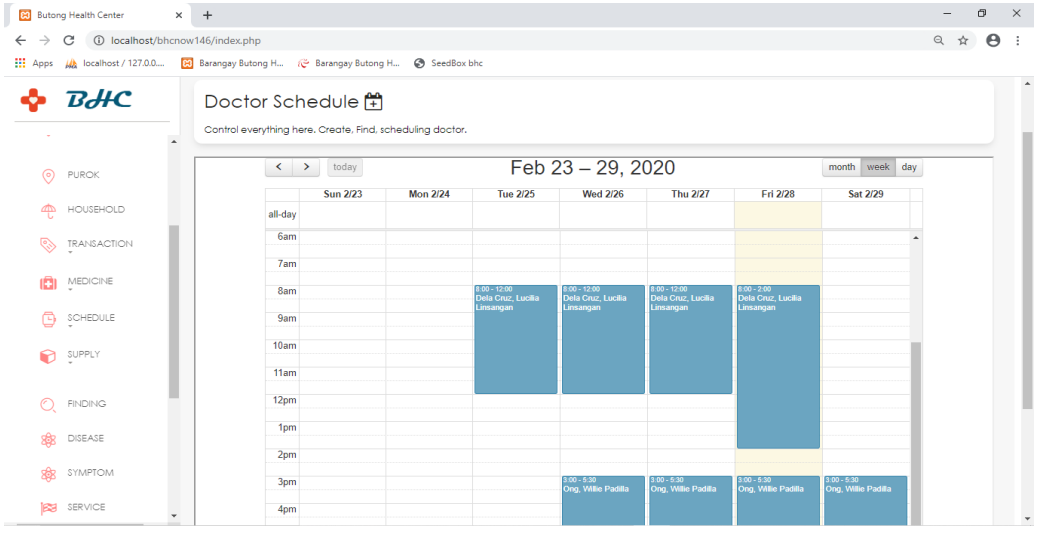

Figure 52. Scheduling by Week

Figure 52 shows the schedule by week wherein the admin can add doctors by week.

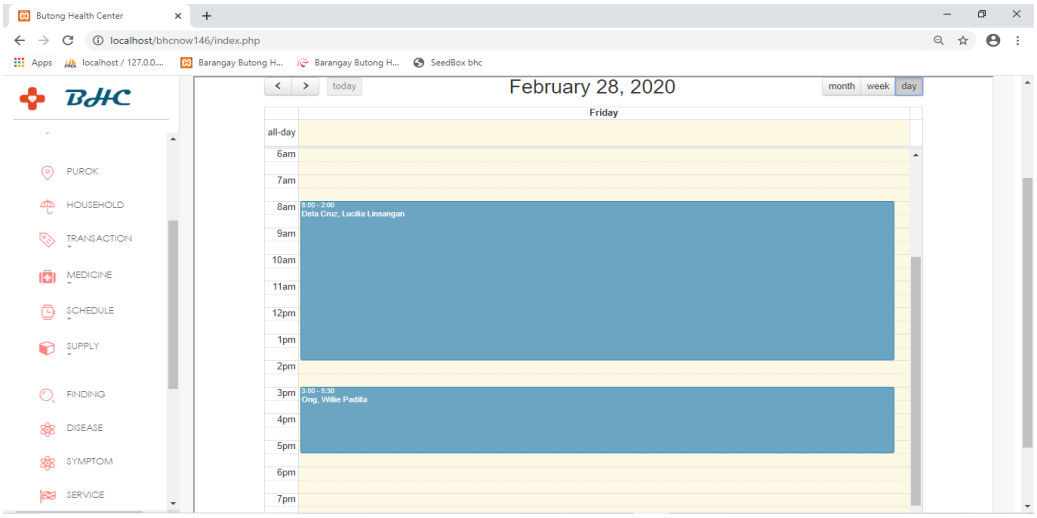

Figure 53. Scheduling by Day

Figure 53 shows the schedule by day wherein the administrator can view the schedule of the doctor in that particular day.

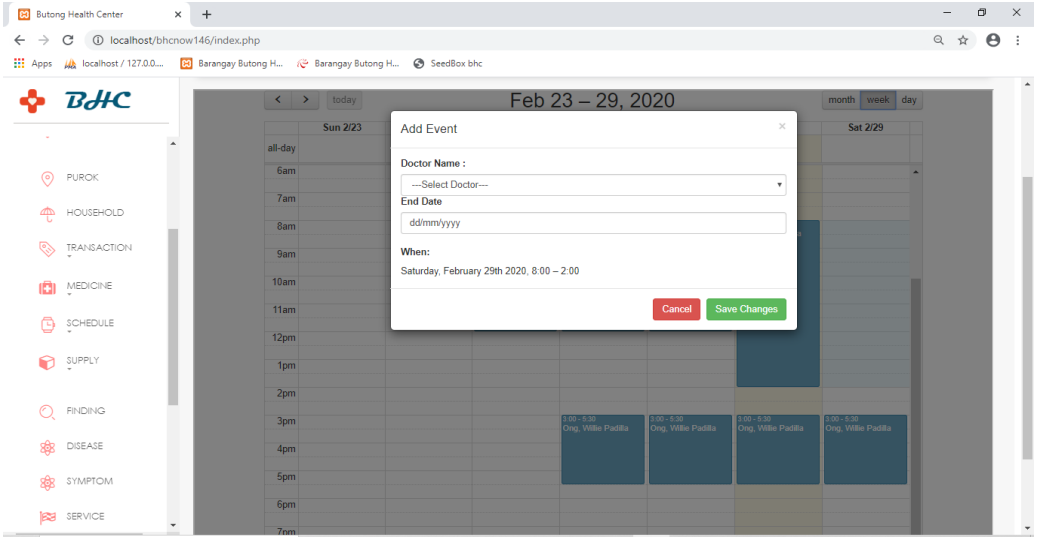

Figure 54. Adding of Scheduling by Week

Figure 54 shows the adding of schedule by week wherein the administrator can select in the dropdown list box for the available doctor.

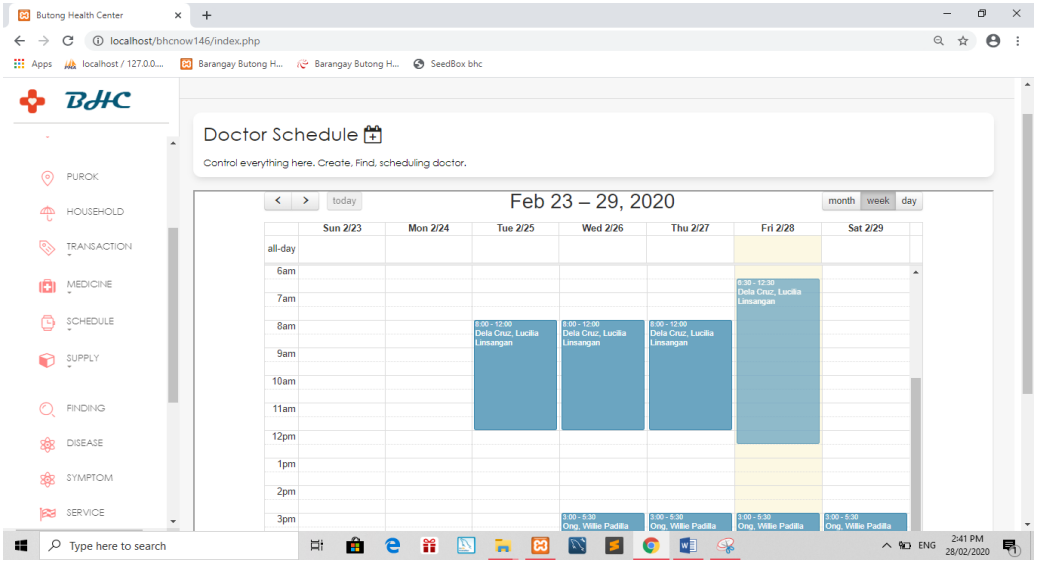

Figure 55. Scheduling Update

Figure 55 shows the updating of schedule wherein the administrator need to drag the corresponding scheduling.

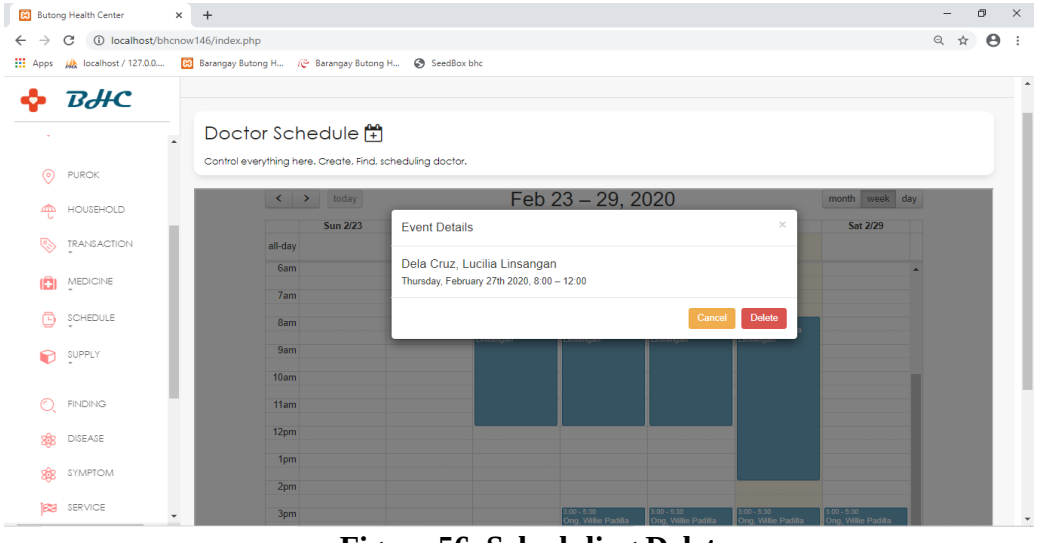

Figure 56. Scheduling Delete

Figure 56 shows the scheduling delete module wherein the administrator can delete the corresponding schedule of doctor.

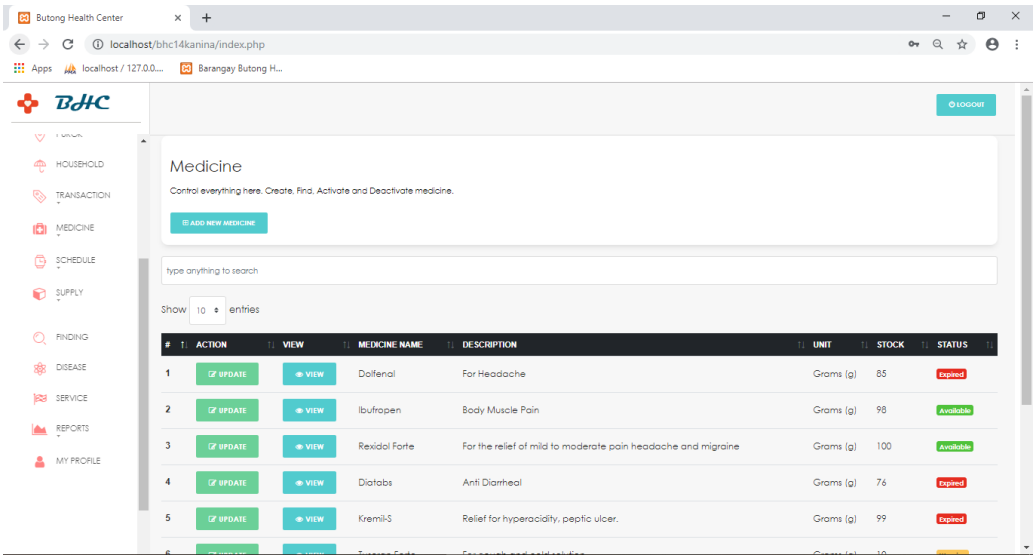

Figure 57. Managing Medicine Inventory

Figure 57 shows the Managing of Medicine Inventory wherein in the administrator can add update view and search medicine.

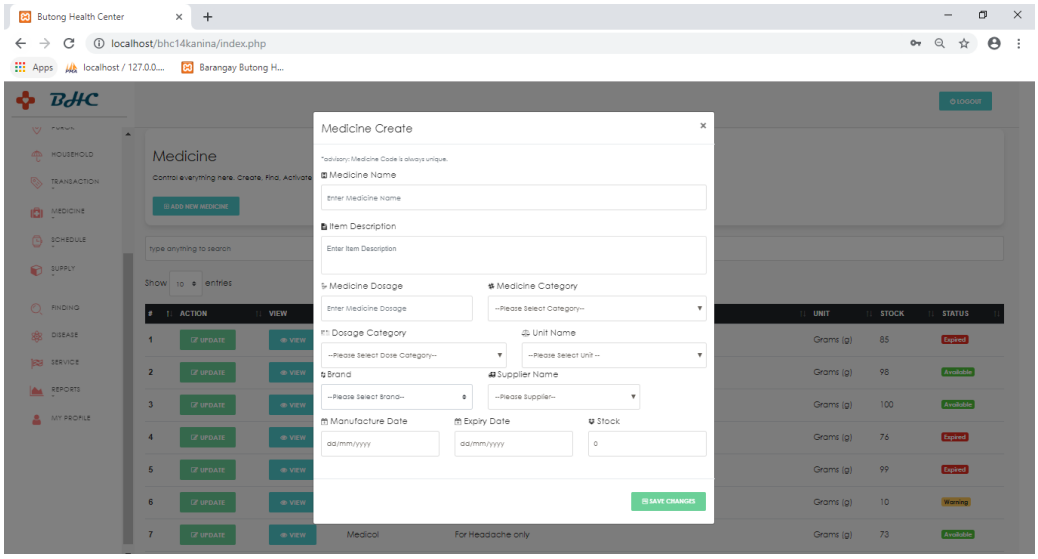

Figure 58. Adding of Medicine

Figure 58 shows the Medicine Create form wherein the administrator required to fill up all the information given in order to add new Medicine.

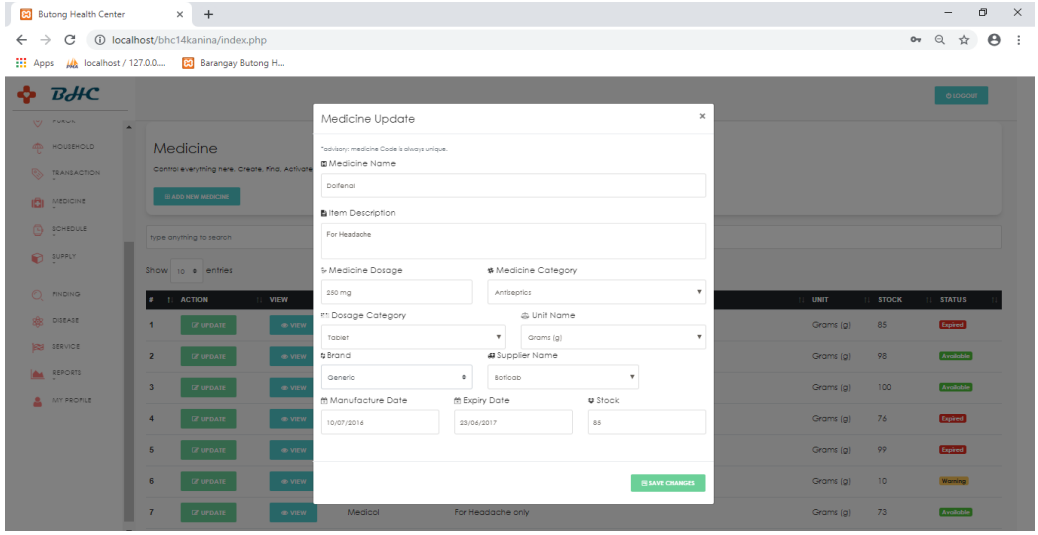

Figure 59. Update Medicine

Figure 59 shows the Medicine Update form wherein the administrator can update medicine information.

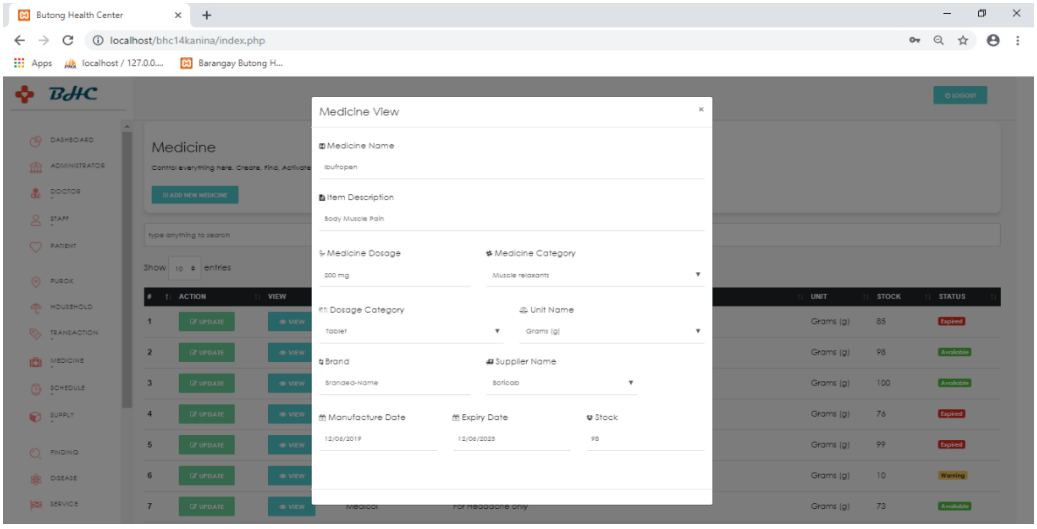

Figure 60. View Medicine

Figure 60 shows the Medicine View form wherein the administrator can view medicine information.

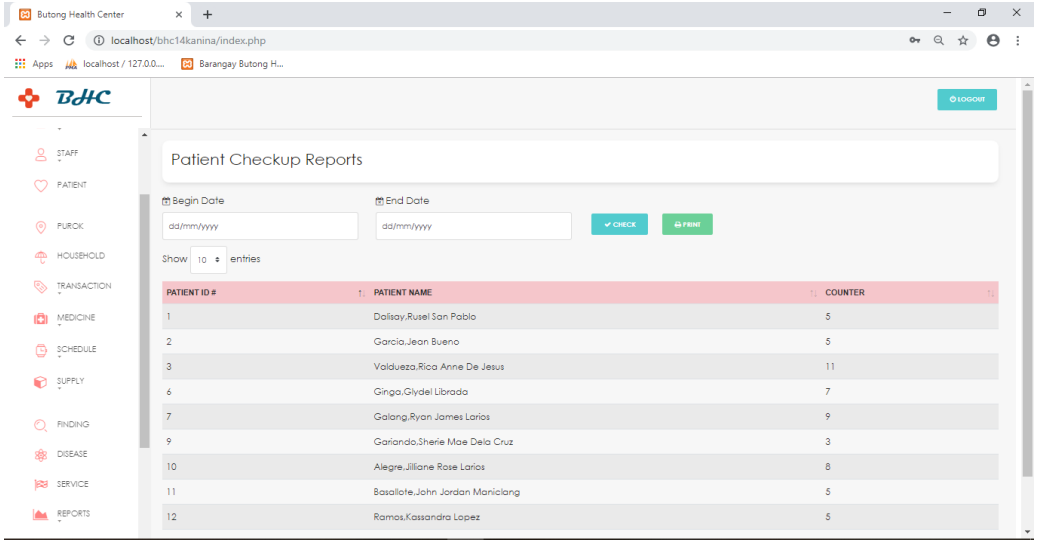

Figure 61. Patient Checkup Reports

Figure 61 shows the Patient Checkup Reports wherein the administrator can see how many patient is checking checkup and print it.

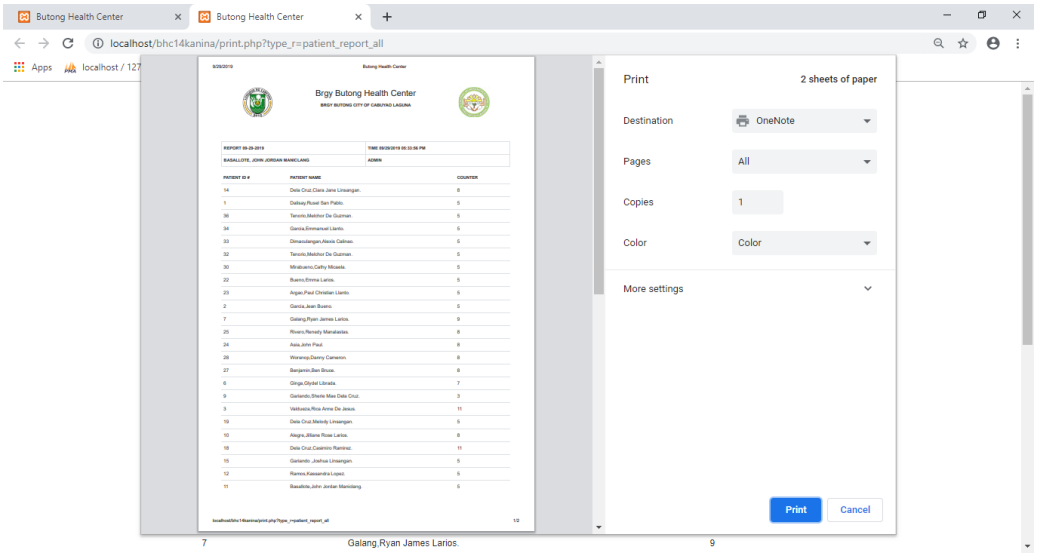

Figure 62. Patient Checkup Report Print

Figure 62 shows the Patient Checkup Report Print tab wherein the administrator can print all the Patient Checkup Reports data

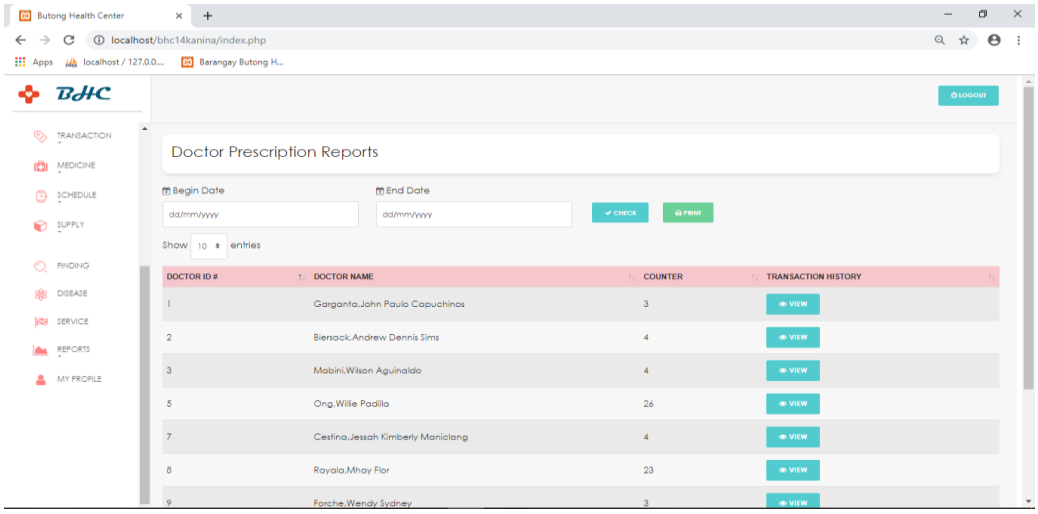

Figure 63. Doctor Prescription Reports

Figure 63 shows the Doctor Prescription Reports wherein the administrator can see how many patient will the doctor checkup and print it.

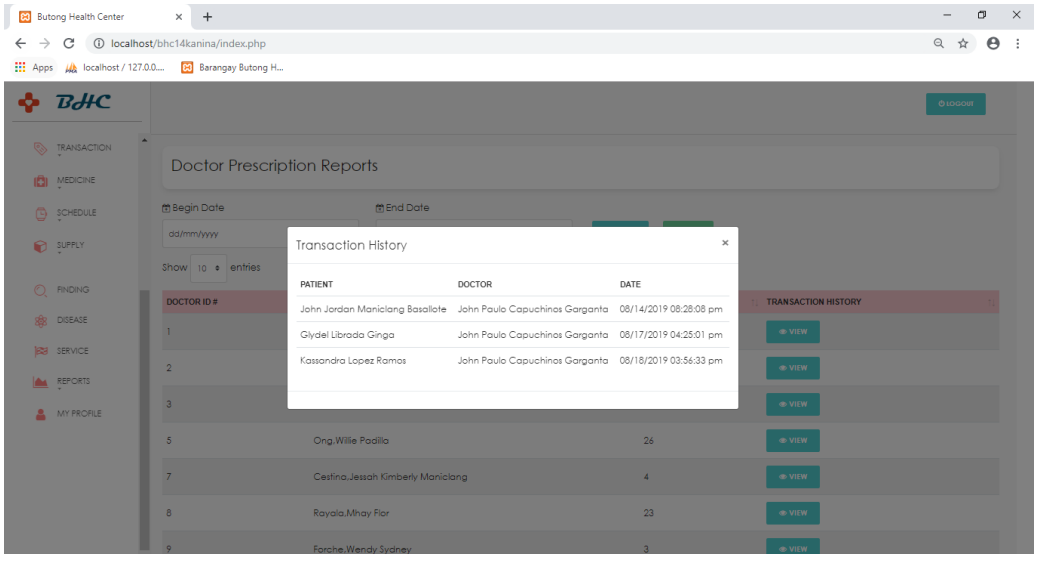

Figure 64. Doctor Prescription Reports View History

Figure 64 shows the Transaction History form wherein the administrator can see how many patient will the doctor checkup.

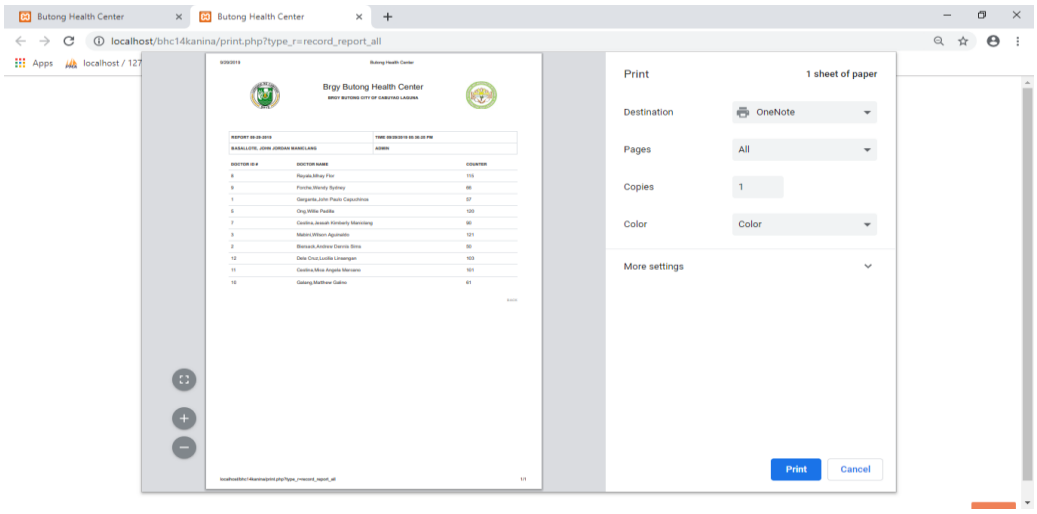

Figure 65. Doctor Prescription Reports Print

Figure 65 shows the Doctor Prescription Reports print tab wherein the administrator will able to print all the prescription.

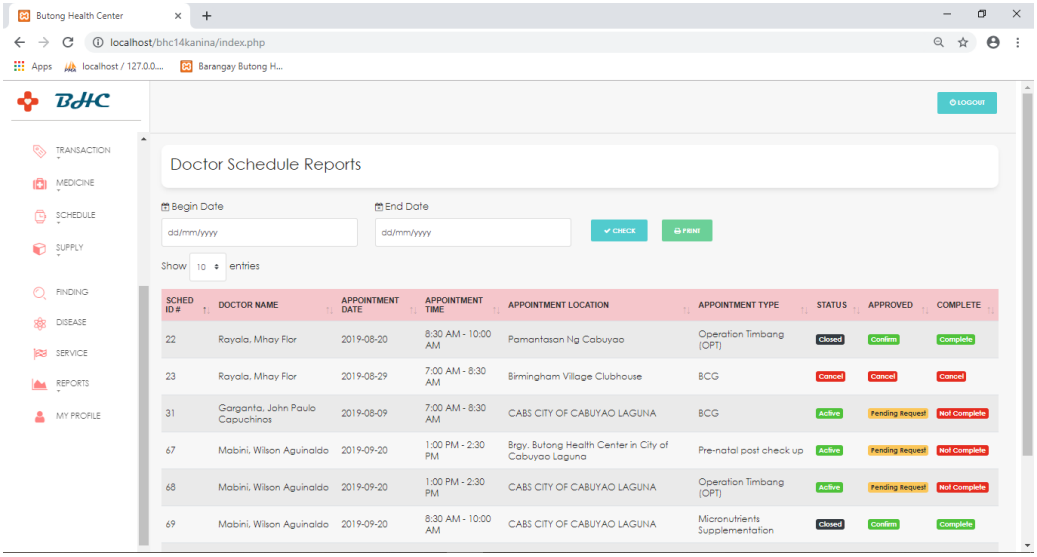

Figure 66. Doctor Schedule Reports

Figure 66 shows the Doctor Schedule Reports wherein the administrator can see the schedule of the doctors and able to print it.

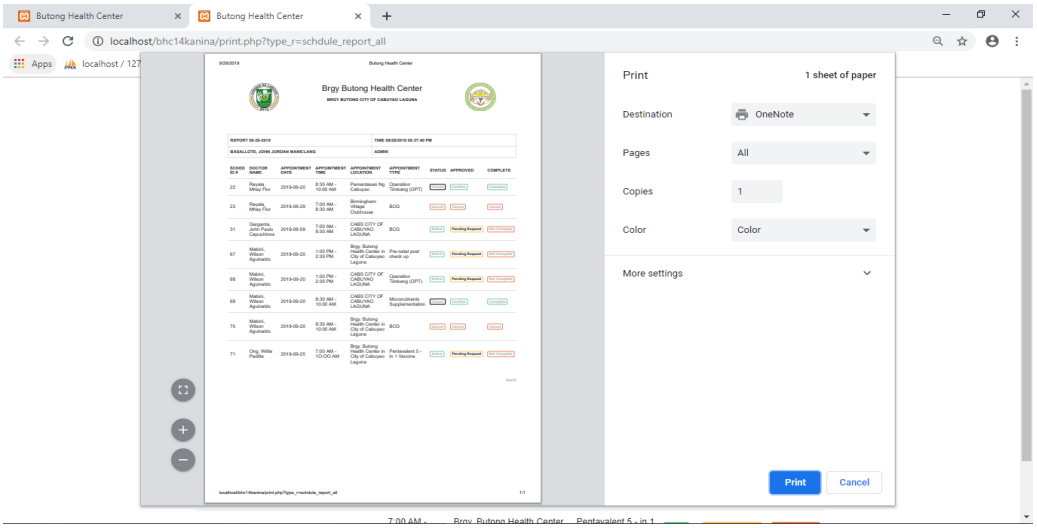

Figure 67. Doctor Schedule Report Print

Figure 67 shows the Doctor Schedule Report print tab wherein the administrator will able to print all the doctor schedule.

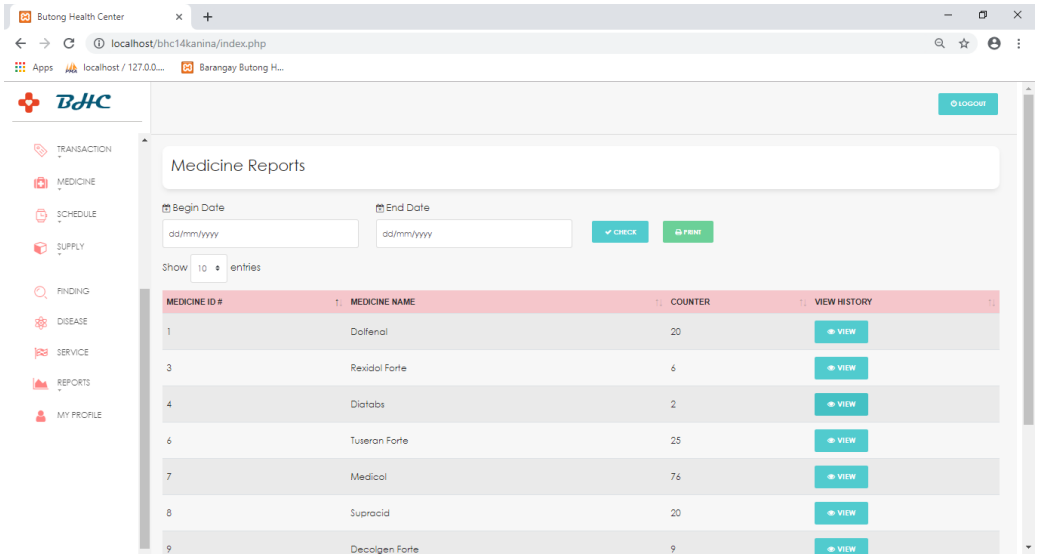

Figure 68. Medicine Reports

Figure 68 shows on the Medicine Reports wherein the administrator can see how many medicine is being distribute to the patient and able to print it.

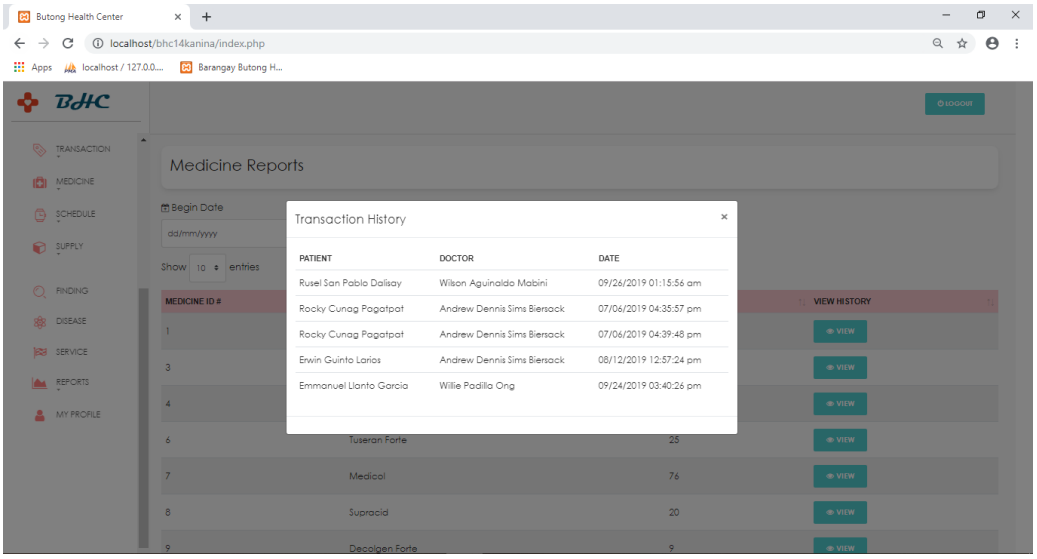

Figure 69. Medicine Report View History

Figure 69 shows Transaction History form wherein the administrator can see the name of the doctor and date where they distribute medicine to patient.

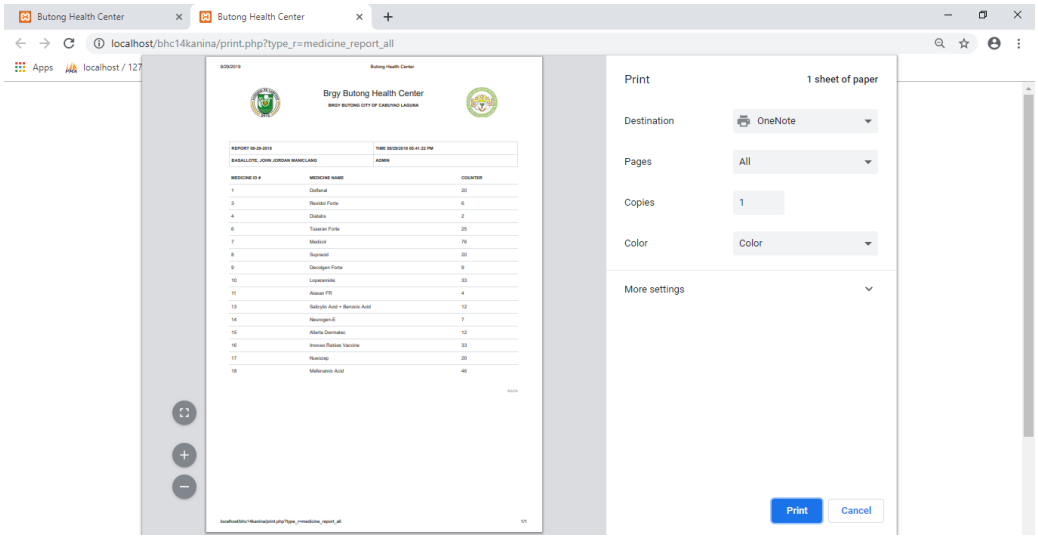

Figure 70. Medicine Report Print

Figure 70 shows the Medicine Report print tab wherein the administrator will able to print all the medicine

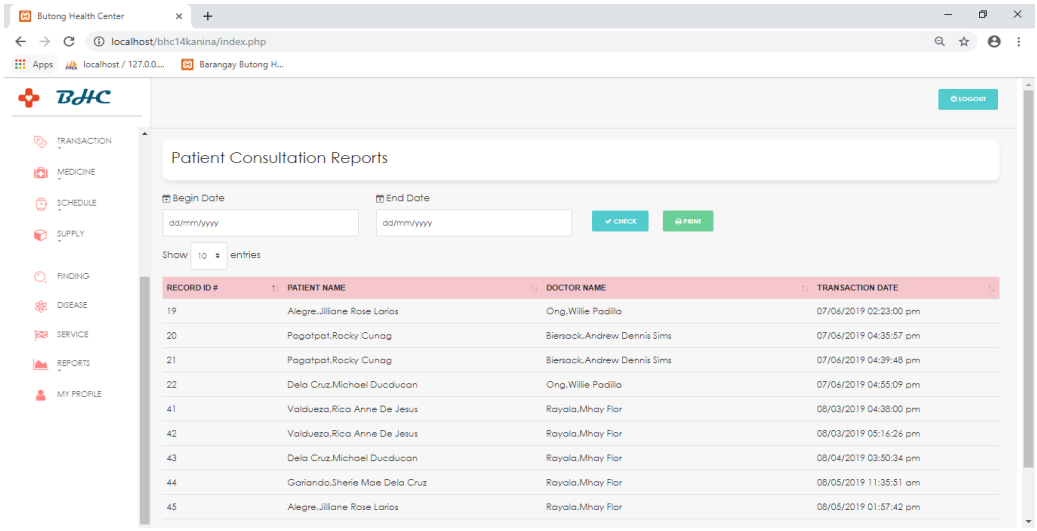

Figure 71. Patient Consultation Reports

Figure 71 shows the Patient Consultation Reports wherein the administrator can see the patient and who doctor check up every patient.

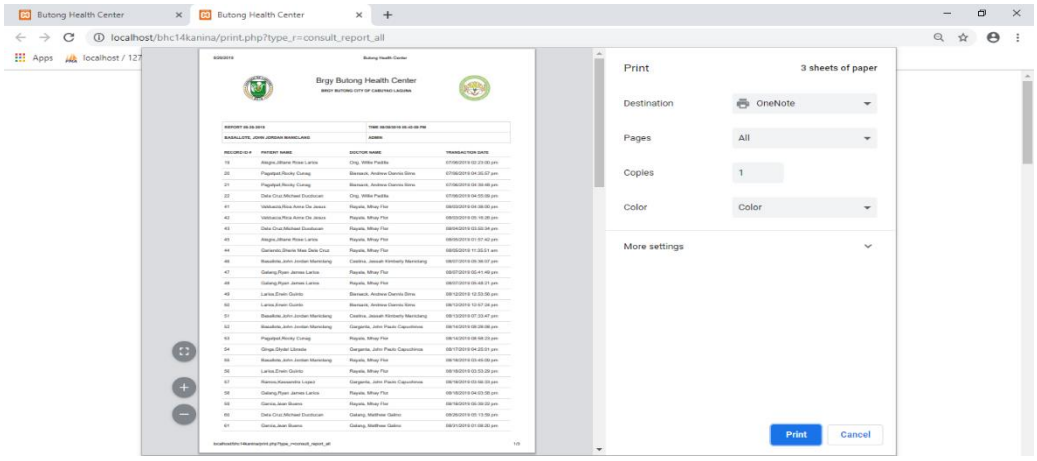

Figure 72. Patient Consultation Reports Print

Figure 72 shows the Patient Consultation Report print tab wherein the administrator will able to print all patient consultation with name of the patient doctor and date.

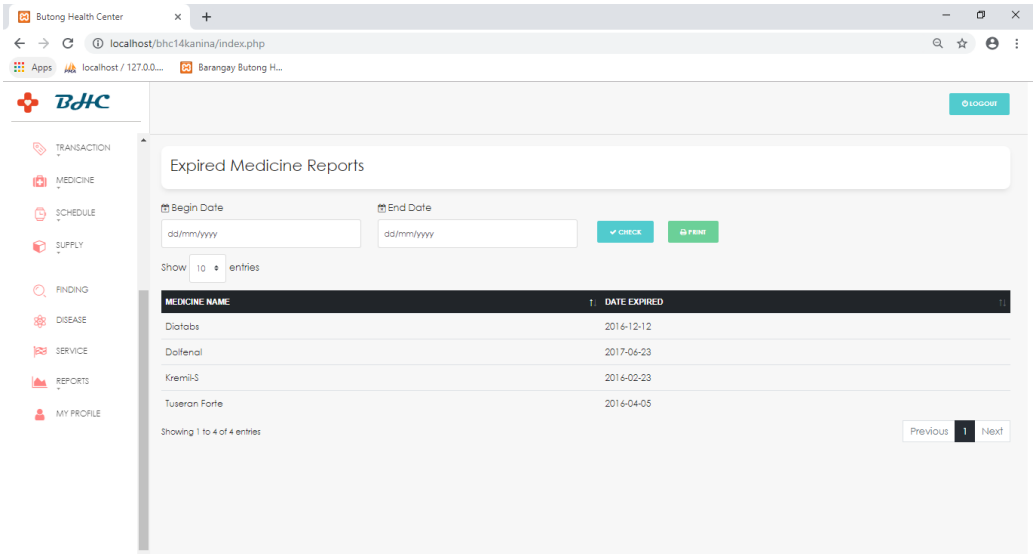

Figure 73. Expired Medicine Reports

Figure 73 shows the Expired Medicine Report wherein the administrator will able to see what medicine is already expired.

Figure 74. My Profile Information

Figure 74 shows the My Profile wherein the Administrator will able to change his/her profile.

System Requirements and Installation

The researchers provided a guide for the user about the requirements needed and the steps in installing the proposed system. This would guide the user to install and test the system correctly.

The following system hardware requirements are needed to use the system appropriately:

- 4 Gigabyte RAM or higher;
- Dual core processor or higher;
- Video card or On board graphics 512 Megabyte or higher;
- At least 2 Gigabyte hard disk space for data storage;
- Monitor, keyboard and mouse;

The computer must have the following installed software before using the system:

- Windows operating system 7/8/8.1/10;
- MySQL Workbench 6.3 CE;
- XAMPP Control Panel;
- Sublime Text 3;
- Web browser: Google Chrome 62.0, Mozilla Firefox 42.0

In order for the web application to access on personal computer, a web browser must be installed and a stable internet connection is needed. Since the web application is hosted on the

internet, there is no need to install the application on the users' personal computer or laptop.

The following steps are process in accessing the web application on the internet:

1. If there is no web browser installed to the current computer, the user must download a browser (i.e. Mozilla Firefox, Google chrome, etc.), then proceed to next step.
2. Open the web browser and access the web application by typing the URL of the web application: "seedbox.info.tm/blog/butong/home/index.php."
3. After accessing the URL, the user can now visit the web application and be able to browse, or create an account for better experience.

Testing Procedure

The following testing procedures were used to test the system's implementation. This guided the users to test the system correctly.

Alpha Testing. This is the first stage where developers observed the users and took down notes about the problem encountered by the users. The users found bugs that were not found during the progress of the system. It was conducted to enhance the system by discovering and fixing the said errors that were not exposed during the forgoing tests. The ending experiment was tested by the researchers by creating a transaction to test by entering an incorrect entry to check if the validation appears. The researchers tested the system to ensure that the system provides complete information.

Beta Testing. This was the last stage of testing where the developers brought the proposed system on client's site for initial test of the software. The testing was assessed as to whether the method conceded the necessities required, this was very significant for the developers to finalize their released version of the application.

System Evaluation

A system evaluation is utilized to determine if the performance of the system acquired all the necessary requirements to make it achieve its purpose. The End User evaluates under the criteria of Usability, Security, Functional Suitability, Performance Efficiency, while the Web Development Expert evaluates under the criteria of Security, Maintainability, Reliability and Usability. System evaluation verified and showed the result of the assessments of the users with accordance to the following:

For Users:

Usability. It refers to the system's capability in terms of being user-friendly.

Security. The system required username and password and confidential records could only be accessed by an authorized user of the system.

Functional Suitability. The system performs its function quickly especially in searching a particular record.

Performance Efficiency. The system will immediately performs what the user needs.

For Web Development Experts:

Security. It refers to the system's capability in providing security to the data of the users.

Maintainability. The system probability that an item is retained in or restored to a specified condition within a given period of time, when the maintenance is performed in accordance with prescribed procedures and resources.

Reliability. It refers in the system's capability in providing a fast and reliable access in recovery of records and in storing records properly.

Usability. It refers to the system's capability of the system to change to new specifications or operating environments.

Assessment of the Users on the proposed system in terms of:

The proposed system was evaluated by Thirty (30) users in terms of Usability, Security, Functional Suitability and Performance and Efficiency. Likert Scale was used for the evaluation and are analyzed and interpreted using median and percentage. The results are as follows:

Table 2. Median for the Assessment of the Users under the Category of Usability

Question	SA (5)	A (4)	U (3)	DA (2)	SD (1)	Total Percentage	Median
Q1	36.7%	50%	13.3%	0%	0%	100%	4.0
Q2	46.7%	33.3%	20%	6.7%	0%	100%	4.0
Q3	20%	66.7%	13.3%	0%	0%	100%	4.0

Where:

Q1 - The layout of the system interface is consistent.

Q2 - The system uses high contrast of colors which promotes readability.

Q3 - The graphic content is appropriate to the purpose of the system.

Table 2 displays the assessment of the user for the proposed system in terms of Usability. Q1, Q2 and Q3 received a median of

4.0. It implies that the users agree that the system interface is consistent, uses high contrast of colors which promotes readability and the graphic content appropriate to the purpose of the system.

Table 3. Median for the Assessment of the Users under the Category of Security

Question	SA (5)	A (4)	U (3)	DA (2)	SD (1)	Total Percentage	Median
Q1	0%	50%	33.3%	16.7%	0%	100%	4.0
Q2	10%	30%	33.3%	20%	6.7%	100%	4.0
Q3	20%	26.7%	33.3%	20%	0%	100%	4.0

Where:

Q1 - Users of the system are given appropriate access levels based on their category.

Q2 - All actions performed by the users involving access of data in the system are logged.

Q3 - All actions performed by the users involving modification (i.e., insertion, update and deletion) of data in the system are logged.

Table 3 displays the assessment of the user for the proposed system in terms of Security. Q1, Q2 and Q3 received a median of 4.0. It implies that the users agree that the system gives appropriate access levels based on their category, all actions performed by the users involving access of data in the system

are not logged and all actions performed by the users involving modification (i.e., insertion, update and deletion) of data in the system are not logged.

Table 4. Median for the Assessment of the Users under the Category of Functional Suitability

Question	SA (5)	A (4)	U (3)	DA (2)	SD (1)	Total Percentage	Median
Q1	6.7%	43.3%	50%	0%	0%	100%	4.0
Q2	6.7%	43.3%	50%	0%	0%	100%	4.0
Q3	0%	40%	50%	0%	10%	100%	4.0

Where:
Q1 - The system can perform all types of transactions needed by the user.
Q2 - The result of all transactions performed by the system is accurate.
Q3 - The system can perform all types of content management operations (e.g., posting of news, deletion of spam posts, and updating of page content) needed by the user.

Table 4 displays the assessment of the user for the proposed system in terms of Functional Suitability. Q1, Q2 and Q3 received a median of 4.0. It implies that the users agree that can perform all types of transactions needed by the user, the result of all

transactions performed by the system is accurate and can perform all types of content management operations (e.g., posting of news, deletion of spam posts, and updating of page content) needed by the user.

Table 5. Median for the Assessment of the Users under the Category of Performance Efficiency

Question	SA (5)	A (4)	U (3)	DA (2)	SD (1)	Total Percentage	Median
Q1	20%	30%	26.7%	20%	3.3%	100%	4.0
Q2	16.7%	33.3%	4%	10%	0%	100%	4.0
Q3	43.3%	16.7%	16.7%	16.7%	6.7%	100%	4.0

Where:
Q1 - The system displays any information requested by the user in a real-time manner.
Q2 - The system performs immediate modification (insertion, update, or deletion) of data in the database as requested by the user.
Q3 - The system performs transactions in a fast manner

Table 5 displays the assessment of the user for the proposed system in terms of Performance Efficiency. Q1, Q2 and Q3 received a median of 4.0. It implies that the system displays any information requested by the user in a real-time manner, performs immediate

modification (insertion, update, or deletion) of data in the database as requested by the user and performs transactions in a fast manner.

Assessment of the Web Development Experts

The proposed system was evaluated by two (2) Web Development Experts in terms of Security, Usability, Accuracy, and Functionality. Likert Scale was used for the evaluation and is analyzed and interpreted using median and percentage.

Table 6. Median for the Assessment of the Web Development Experts under the Category of Security

Question	SA (5)	A (4)	U (3)	DA (2)	SD (1)	Total Percentage	Median
Q1	0%	50%	50%	0%	0%	100%	4.0
Q2	0%	100%	0%	0%	0%	100%	4.0
Q3	0%	50%	50%	0%	0%	100%	4.0

Where:

Q1 - The system has features to protect sensitive data.

Q2 - Users of the system are given appropriate access levels based on their category.

Q3 - The system is not vulnerable to authentication-related attacks.

Table 6 displays the assessment of the Web Development Experts for the proposed system in terms of Security. Q1, Q2 and

Q3 received a median of 4.0. It implies that the experts agree that the system has a features to protect sensitive data, users of the system are given appropriate access levels based on their category and not vulnerable to authentication-related attacks.

Table 7. Median for the Assessment of the Web Development Experts under the Category of Maintainability

Question	SA (5)	A (4)	U (3)	DA (2)	SD (1)	Total Percentage	Median
Q1	0%	50%	50%	0%	0%	100%	4.0
Q2	0%	50%	50%	0%	0%	100%	4.0
Q3	0%	100%	0%	0%	0%	100%	4.0

Where:
Q1 - Identifiers (names of variables, methods, structures, etc.) are descriptive.
Q2 - A consistent naming convention for identifiers is observed.
Q3 - Codes are properly indented.

Table 7 displays the assessment of the Web Development Experts for the proposed system in terms of Maintainability. Q1, Q2 and Q3 received a median of 4.0. It implies that the experts agree that the system's identifiers (names of variables, methods, structures, etc.) are descriptive, consistent naming convention for identifiers is observed and codes are properly indented.

Table 8. Median for the Assessment of the Web Development Experts under the Category of Reliability

Question	SA (5)	A (4)	U (3)	DA (2)	SD (1)	Total Percentage	Median
Q1	0%	50%	50%	0%	0%	100%	4.0
Q2	0%	0%	100%	0%	0%	100%	3.0
Q3	0%	50%	50%	0%	0%	100%	4.0

Where:

Q1 - The system remains usable even in the case wherein the assets (e.g., stylesheets, scripts, and images) did not load properly.

Q2 - The system can recover lost data due to erroneous transactions made by the user.

Q3 - The system can be accessed using other modes of input in case of failure in the input device.

Table 8 displays the assessment of the Web Development Experts for the proposed system in terms of Reliability. Q1, Q2 and Q3 received a median of 4.0. It implies that the experts agree that the system remains usable even in the case wherein the assets (e.g., stylesheets, scripts, and images) did not load properly, can recover lost data due to erroneous transactions made by the user and can be accessed using other modes of input in case of failure in the input device.

Table 9. Median for the Assessment of the Web Development Experts under the Category of Usability

Question	SA (5)	A (4)	U (3)	DA (2)	SD (1)	Total Percentage	Median
Q1	0%	100%	0%	0%	0%	100%	4.0
Q2	0%	50%	50%	0%	0%	100%	4.0
Q3	0%	50%	50%	0%	0%	100%	4.0

Where:

Q1 - The layout of the system interface is consistent.

Q2 - The system uses high contrast of colors which promotes readability.

Q3 - The graphic content is appropriate to the purpose of the system.

Table 9 displays the assessment of the Web Development Experts for the proposed system in terms of Usability. Q1, Q2 and Q3 received a median of 4.0. It implies that the experts agree that the layout of the system interface is consistent, uses high contrast of colors which promotes readability and graphic content is appropriate to the purpose of the system.

CHAPTER V

SUMMARY OF FINDINGS, CONCLUSIONS AND RECOMMENDATIONS

This chapter presents the summary of conclusions, recommendations of the study based on the findings of the proposed system. The study tackled the general objectives of determining the current practices and problems encountered by Barangay Butong Health Center.

Summary of Findings

The main purpose of the study was to design and develop an Online Barangay Health Center Information System for Barangay Butong City of Cabuyao Laguna to make their task organize and faster.

Based on the statement of the problem, the following results were accomplished.

1. The current practices and problems encountered by the Barangay Health Center of Butong, Cabuyao Laguna in terms of:

    a. keeping and monitoring patient's record. In keeping and monitoring all the patient records

the Butong health center worker having hard time to find each record of the patient because it manually keeps in filing cabinet.

b. scheduling of doctor. The staff cannot fix the schedule of the doctor because the health workers sometimes cannot be sure in the availability of the doctor in daily checkup of different patients in different schedule.

c. handling medicine inventories. It would take too much time and consume a lot of paper works by looking and recording the stocks, the brand, dosage, expiry date, in and out of items manually.

d. generating reports. The health center worker having hard time in sorting the patient records and other paper works.

2. The proposed system be developed in such a way that it would provide:

    a. manage and monitor the patient's records. The proposed system can manage and monitor all the patients' record easily by sorting or searching it.

b. manage the schedules of doctor. The proposed system can manage the schedule of the doctor which means the patient will easily know which doctor is available.

c. manage medicine inventories. The proposed system can manage medicine inventory, the system will determine if the medicine is still available or it's already expired.

d. generate reports. The proposed system can generate and progress reports that will help the butong health center workers to manage all the result correctly.

3. The result of the assessment of the users on the system from categories of usability, security, functional suitability and performance efficiency which basically means from the Likert Scale was of AGREE category indicating that the system has been accepted and approved by users

4. The result of the assessment of the Web Development Experts on the system from categories security, maintainability, reliability and usability were analyzed. The assessment of the web development experts based from the evaluation survey was done which basically means from the

Likert Scale was of AGREE category indicating that the system has attained the requirements needed.

Conclusions

Based on the summary of findings the following conclusions were drawn by the researchers;

1. On the workflows and accustomed practices of the Barangay Butong Health Center, the researchers figured out that manual way of managing patients record, scheduling of doctor, managing medicine inventories and generating reports of Barangay Butong Health Center were time consuming for the health center workers.

2. The newly developed Online Barangay Health Center Information System for Barangay Butong City of Cabuyao Laguna met the requirements of the users, it also helped the client in giving visualization of the data and using it in more efficient use of information. The researchers concluded that the system helped the client to ease the load of work. It has a nice UI Design that can easily understand by normal user.

3. The result of the assessment of the users fell in the agreed category; therefore, the researchers concluded that the proposed system could perform its function effectively.

4. The result on the assessment of the web development experts was agreed; therefore, the researchers concluded that the proposed system was more appropriate to use then manual operation.

Recommendations

The following recommendations were formulated by the researchers based on the findings and conclusions of the study.

1. The researchers recommend the implementation of the new system and new workflows to replace the manual and traditional way of conducting the transaction.

2. The researchers recommend the development of the similar data systems with better database for handling a large amount of data in the future.

3. The user of the system should be well-informed and must have enough knowledge in operating the system and must know the flow of the whole system.

4. Maintenance of the system is required so that it could run properly and also update the records and information.

Implementation Plan

As the development of the proposed web application is done, the implementation is next. And to properly implement the web application the installation process must be done first before doing the other tasks.

Implementation is the part wherein the finished system is used based on the needs of the client to see if the web application have successfully met its requirements based on its function and performance.

If the web application has met all the requirements needed so that it could run on the internet, the web application is ready for online hosting. The first step in the implementation plan is that the proponents would look for a web hosting company that will enable the web application to be uploaded on the internet and be available for the user.

The next step, after looking for a web hosting company, the next thing to do is to buy a unique domain name for the site. And then, settle the accounts needed and the web application will be uploaded and be available online.

References

Amante, D. (2014) *Access to care and use of the internet to search for health information: Results from the US national health interview survey* (Retrieve on March 26, 2019 at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4430679/)

Civilek, I. (2015) *Blood platelet inventory management with protection levels* (Retrieve on March 26, 2019 at: https://www.sciencedirect.com/science/article/abs/pii/S0377221715000430/)

Crossman, A. (2018). *Understanding purposive sampling*. (Retrieved on March 26, 2019 at: https://www.thoughtco.com/purposive-sampling-3026727)

Diviani N. (2015*) Low health literacy and evaluation of online health information: a systematic review of the literature* (Retrieve on March 26, 2019 at: https://www.ncbi.nlm.nih.gov/pubmed/25953147)

Dudovskiy, J. (2018*). Descriptive research.* (Retrieved on March 26, 2019 at: https://research methodology.net/descriptive-research/)

Eysenbach, G. (2015*) Low health literacy and evaluation of online health information:* A Systematic review of the literature (Retrieve on March 26, 2019 at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4468598/)

Eysenbach, G. (2015) *eHealth literacy and web 2.0 health information seeking behaviors among baby boomers and older adults* (Retrieve on March 26, 2019 at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4381816/)

Gu Ziang & Li. (2014) *Inter-facility transport in inventory management and fulfillment systems*

(Retrieve on March 26, 2019 at: https://patents.google.com/patent/US8972045B1/en/)

Goel, V. and Slusky M. *(2015) Slusky constraint programming for LNG ship scheduling and inventory management* (Retrieve on March 26, 2019 at: https://www.sciencedirect.com/science/article/abs/pii/S0377221714007875/)

Jingyao, G., Guoqing, Z., Kevin W.Li (2015) *Efficient aircraft spare parts inventory management under demand uncertainty* (Retrieve on March 26, 2019 at: https://www.sciencedirect.com/science/article/abs/pii/S0969699714001288/)

Laurent (2014). *Secure web application development and execution environment* (Retrieve on March 26, 2019 at: https://patents.google.com/patent/US8800042B2/en)

Mowforth, L. (2018). *Essential tools for front end development* (Retrieve on March 11, 2019 at: https://careerfoundry.com/en/blog/web-development/7-essential-tools-for-front-end-development/)

Picincu, A. (2018). *Types records management system* (Retrieve on March 11, 2019 at: https://bizfluent.com/info-7928109-types-records-management-systems.html)

Simon, L. (2014). *Web application development is different and better* (Retrieve on March 11, 2019 at: https://www.oreilly.com/ideas/web-application-development-is-different-and-better)

Sittig, D. F., & Singh, H. (2015).*A new socio-technical model for studying health information technology in complex adaptive healthcare systems* (Retrieve on March 26, 2019 at: https://link.springer.com/chapter/10.1007/978-3-319-17272-9_4/)

Soyata, T. (2015) *Health monitoring and management using internet-of things (IoT) sensing with cloud-Based processing: Opportunities and Challenges from* (Retrieve on March 26, 2019 at: https://ieeexplore.ieee.org/abstract/document/7207365/)

Sunyaev, A. (2015), *Exploring the far side of mobile health: Information security and privacy of mobile health apps on ios and android* (Retrieve on March 26, 2019 at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4319144/)

Top web application development platforms. (2018). *Web application development platforms* (Retrieve on March 11, 2019 at: https://www.bluepiit.com/blog/web-application-development-platforms/)

Ward, S. (2018). *Creating a document management system* (Retrieve on March 11, 2019 at: https://www.thebalancesmb.com/creating-a-document-management-system-2948084)

Ziukov, S. (2015) *A literature review on models of inventory management under run certainty* (Retrieve on March 26, 2019 at: https://www3.mruni.eu/ojs/business-systems-and-economics/article/view/4130)

Zingg, Holmes, Dettenkofer (2015) *Hospital organization, management, and structure for prevention of health-care-associated infection: A systematic review and expert consensus* (Retrieve on March 26, 2019 at: (https://www.researchgate.net/publication/269173019_Hospital_organisation_management_and_structure_for_prevention_of_health-care-associated_infection_A_systematic_review_and_expert_consensus)

Appendix A – Questionnaire

An Online Barangay Health Center Information System for Barangay Butong City of Cabuyao Laguna

Name (Optional) _____ Date_____

The following questions will evaluate An Online Barangay Health Center Information System for Barangay Butong City Cabuyao Laguna. Please check the necessary box that may describe your answers.

Questionnaire for Users

SA- Strongly Agree A- Agree
SD- Strongly Disagree DA- Disagree
U – Undecided

Criteria	SA 5	A 4	U 3	DA 2	SD 1
USABILITY					
1. The layout of the system interface is consistent.					
2. The system uses high contrast of colors which promotes readability.					
3. The graphic content is appropriate to the purpose of the system.					
SECURITY					

4. Users of the system are given appropriate access levels based on their category.					
5. All actions performed by the users involving access of data in the system are logged.					
6. All actions performed by the users involving modification (i.e., insertion, update and deletion) of data in the system are logged.					
FUNCTIONAL SUITABILITY					
7. The system can perform all types of transactions needed by the user.					
8. The result of all transactions performed by the system is accurate.					
9. The system can perform all types of content management operations (e.g., posting of news, deletion of spam posts, and updating of page content) needed by the user.					
PERFORMANCE EFFICIENCY					
10. The system displays any information requested by the user in a real-time manner.					
11. The system performs immediate modification (insertion, update, or deletion) of data in the database as requested by the user.					
12. The system performs transactions in a fast manner.					

An Online Barangay Health Center Information System for Barangay Butong City of Cabuyao Laguna

Name (Optional) _____ Date_____

The following questions will evaluate An Online Barangay Health Center Information System for Barangay Butong City Cabuyao Laguna. Please check the necessary box that may describe your answers.

Questionnaire for Web Development Experts

SA- Strongly Agree A- Agree
SD- Strongly Disagree DA- Disagree
U – Undecided

Criteria	SA 5	A 4	U 3	DA 2	SD 1
SECURITY					
1. The system has features to protect sensitive data.					
2. Users of the system are given appropriate access levels based on their category.					
3. The system is not vulnerable to authentication-related attacks.					

MAINTAINABILITY					
4. Identifiers (names of variables, methods, structures, etc.) are descriptive.					
5. A consistent naming convention for identifiers is observed.					
6. Codes are properly indented.					
RELIABILITY					
7. The system remains usable even in the case wherein the assets (e.g., stylesheets, scripts, and images) did not load properly.					
8. The system can be accessed using other modes of input in case of failure in the input device.					
9. The system is designed to handle the expected number of users.					
USABILITY					
10. The layout of the system interface is consistent.					
11. The system uses high contrast of colors which promotes readability.					
12. The graphic content is appropriate to the purpose of the system.					

Appendix B – User's Manual

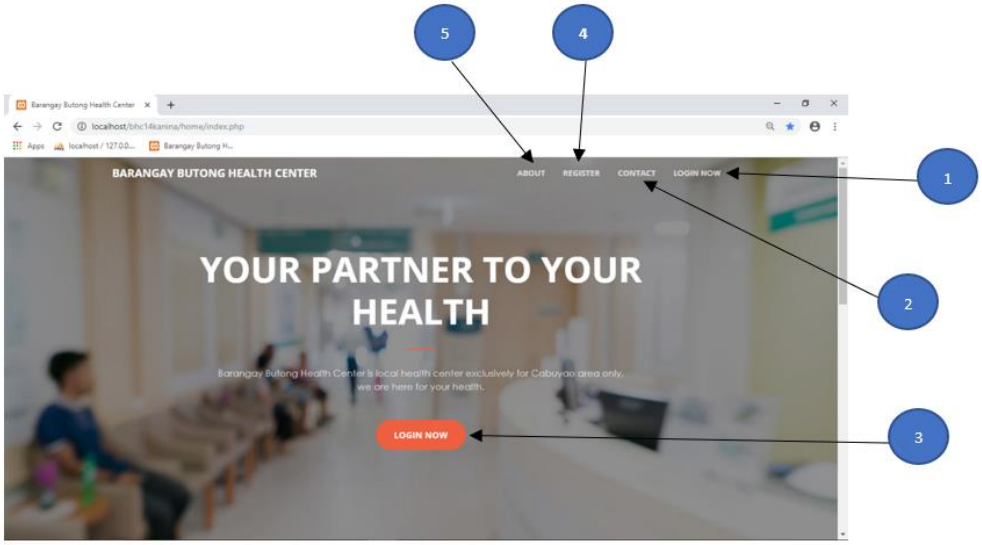

Figure 75. Homepage of the Proposed System

1. **Login Now –** Allows the user to login in the system.
2. **Contact –** Allows the user to see the contact information
3. **Login Button –** Allows the user to login in the system.
4. **Register –** Allows the user to Register in the system.
5. **About –** Allows the user to see all about the system.

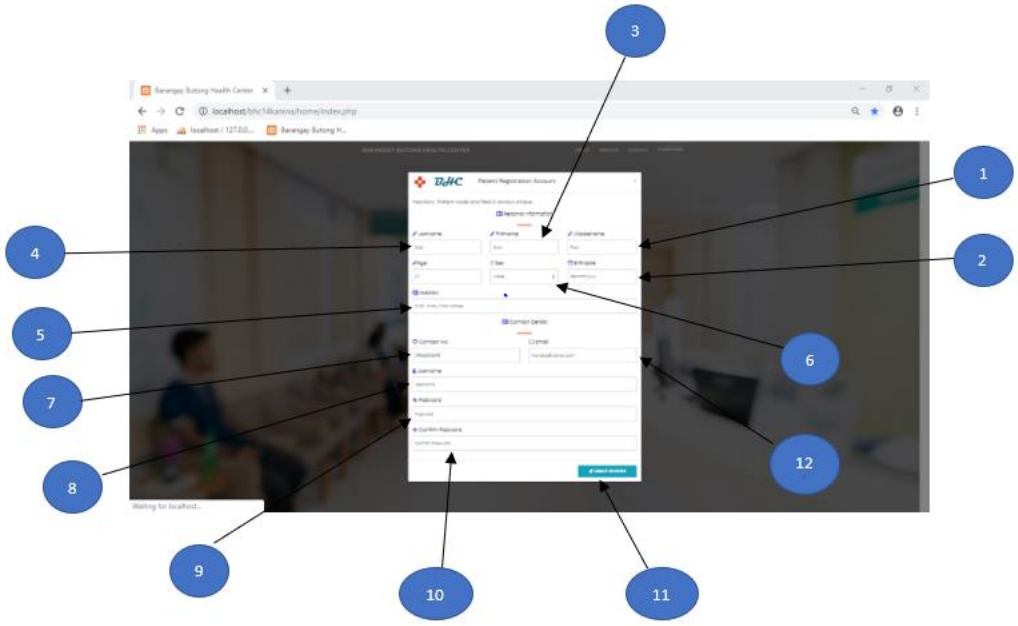

Figure 76. Patient Registration

1. Textbox Middle name– Allows the user to fill up middle name

2. Birthdate – Allows the user to select his/her own birthday

3. Textbox First name – Allows the user to fill up first name.

4. Textbox Last name – Allows the user to fill up last name.

5. Textbox Address – Allows the user to fill up address.

6. Dropdown List box – Allows the user to select sex or gender.

7. Textbox Contact No.– Allows the user to fill up contact number.

8. Textbox Username – Allows the user to fill up username.

9. Textbox Password – Allows the user to fill up password.

10. Textbox Confirm Password – Allows the user to fill up password.

11. Submit Button – Allows the user to submit the fill up form.

12. Textbox Email – Allows the user to fill up email.

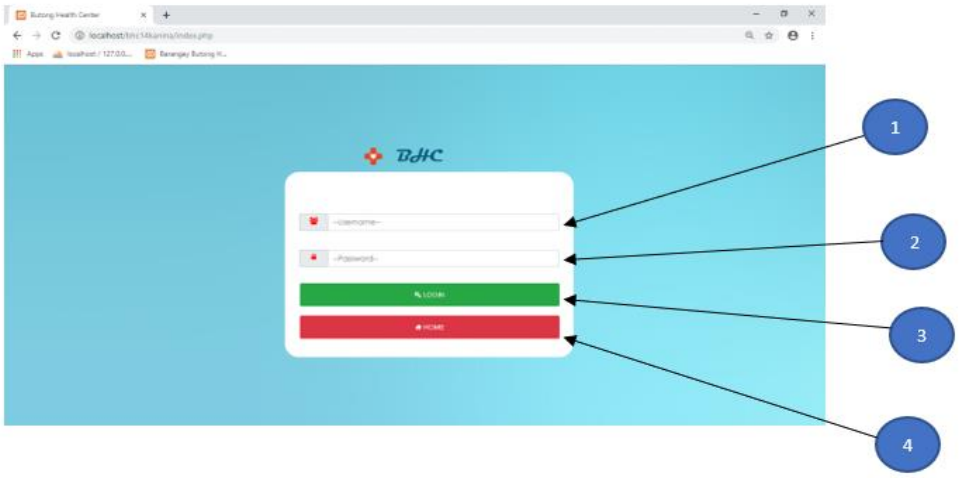

Figure 77. Login Page

1. Textbox Username– Allows the user to fill up username

2. Textbox Password – Allows the user to fill up password

3. Button Login – Allows the user to login in the system.

4. Button Home – Allows the user to back in homepage.

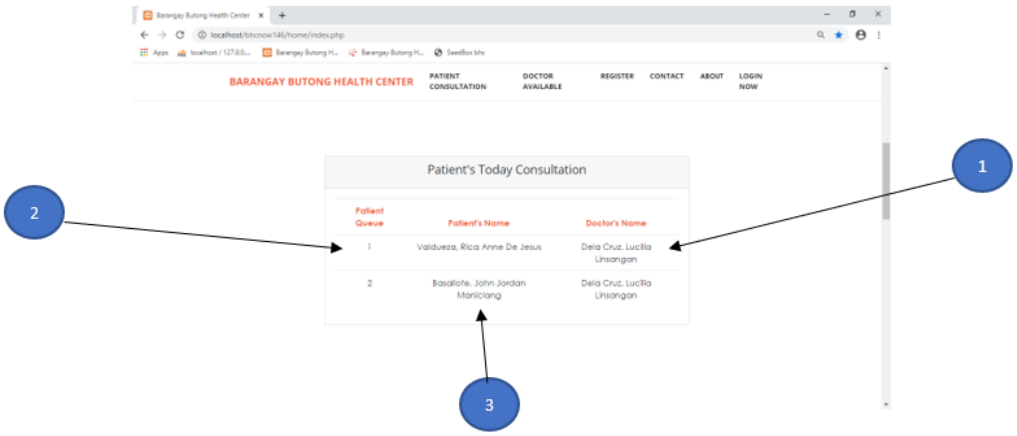

Figure 78. Patient Consultation Posting

1. **Doctor's Name –** Allows the user to see who is available doctor.
2. **Patient Queuing –** Allows the user to see the number of patient.
3. **Patient's Name –** Allows the user to see queuing of patient.

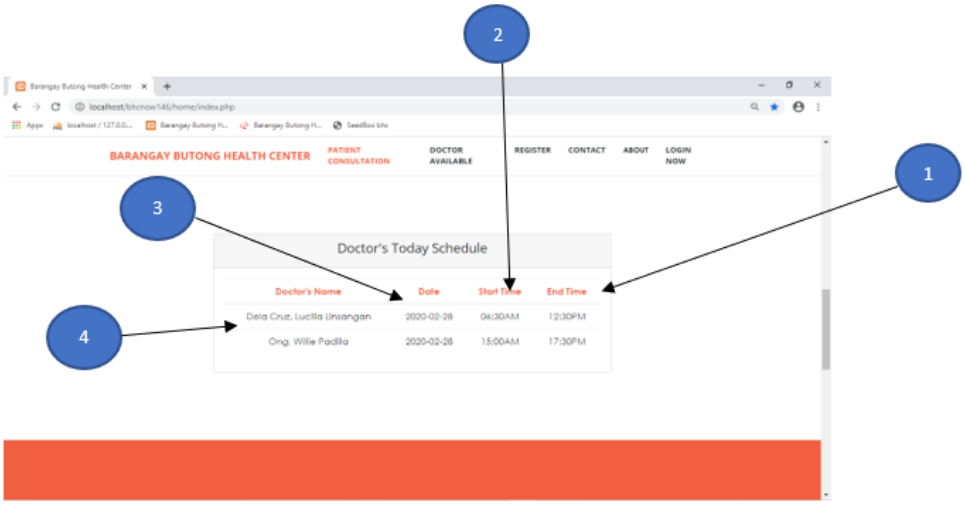

Figure 79. Doctor Schedule Posting

1. **Doctor's Name** – Allows the user to see who is available doctor.
2. **Date** – Allows the user to see the date of the available doctor.
3. **Start Time** – Allows the user to see the start time schedule.
4. **End Time** – Allows the user to see the end time schedule.

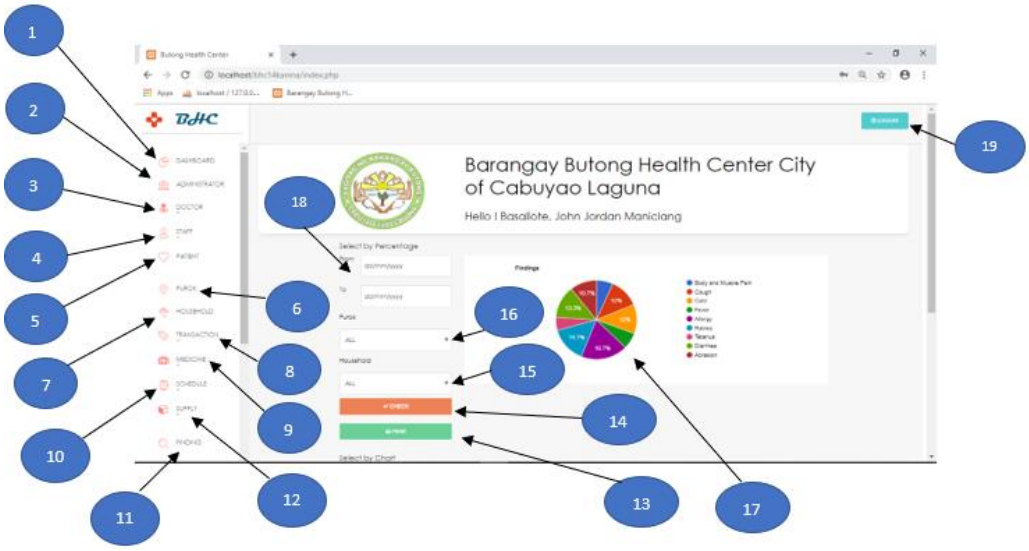

Figure 80. Dashboard of the Proposed System

1. **Dashboard** – Allows the user to navigate Dashboard.
2. **Administrator** – Allows the user to see Admin list.
3. **Doctor** – Allows the user to see Doctors List.
4. **Staff** – Allows the user to see Staff list.
5. **Patient** – Allows the user to see Patient list.
6. **Purok** – Allows the user to see Purok list.
7. **Household** – Allows the user to see Household List.
8. **Transaction** – Allows to see Transaction records.

9. Medicine – Allows to see the Medicine inventory list.

10. Schedule – Allows to see the schedule of doctor.

11. Findings – Allows to see the Findings list.

12. Supply – Allows to see Medicine supply available.

13. Button Print – Button which allows to print the reports.

14. Button Check – Button which allows to Check the reports before print.

15. Dropdown List box – Allows the user to choose from Household list.

16. Dropdown List box – Allows to user to choose from Purok list.

17. Reports – Report Generation of Findings.

18. Calendar Filter – Allows the user to filter from calendar.

19. Button Logout – Button which allows the user to back from Login page

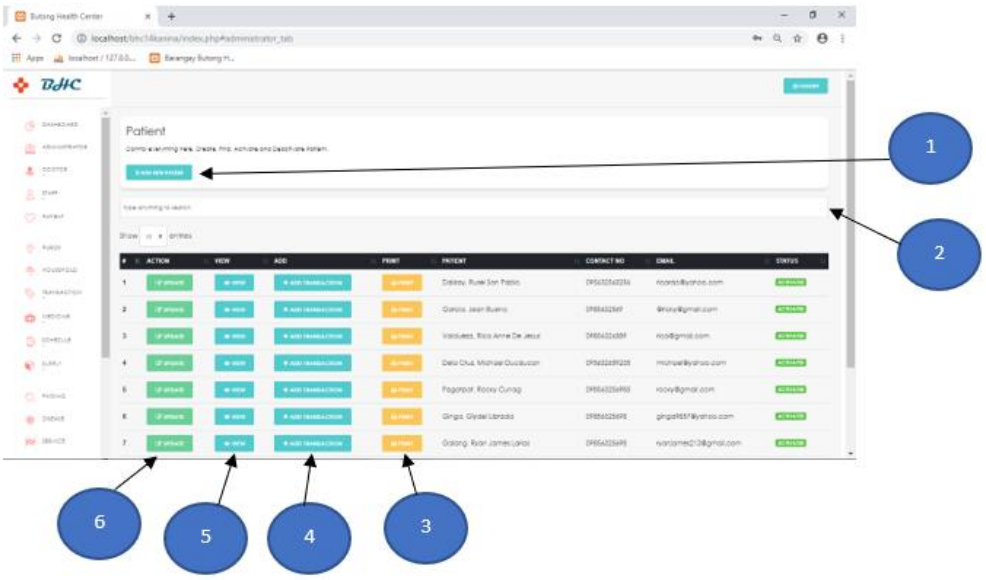

Figure 81. Managing Patient

1. Button Add – Allows the user to add patient.

2. Textbox Search – Allows the user to search patient.

3. Button Print – Allows the user to print patient profile.

4. Button Transaction – Allows the user to add transaction for a particular patient.

5. Button View – Allows the user to view the patient's information

6. Button Update – Allows the user to update patient's information.

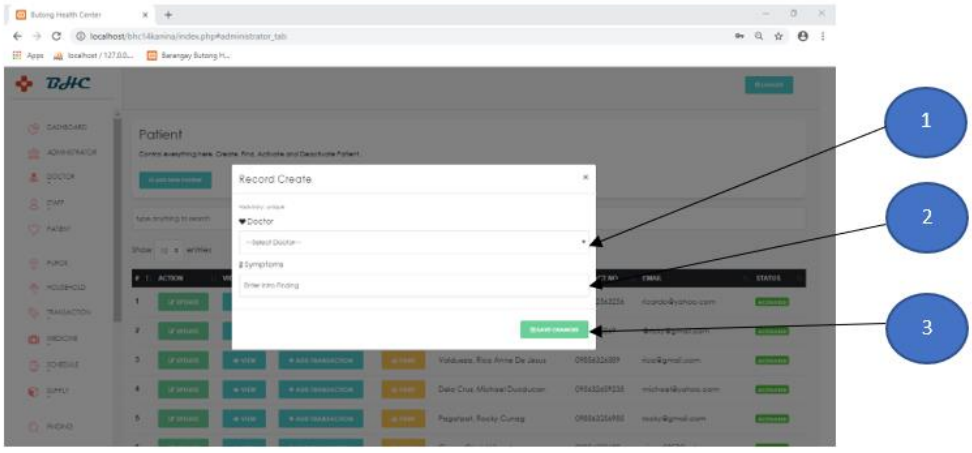

Figure 82. Patient Transaction Record Create

1. Dropdown List box – Allows the user to choose the available doctor.

2. Textbox Findings – Allows the user to fill up what is the patient finding.

3. Button Save – Allows the user to save patient transaction record.

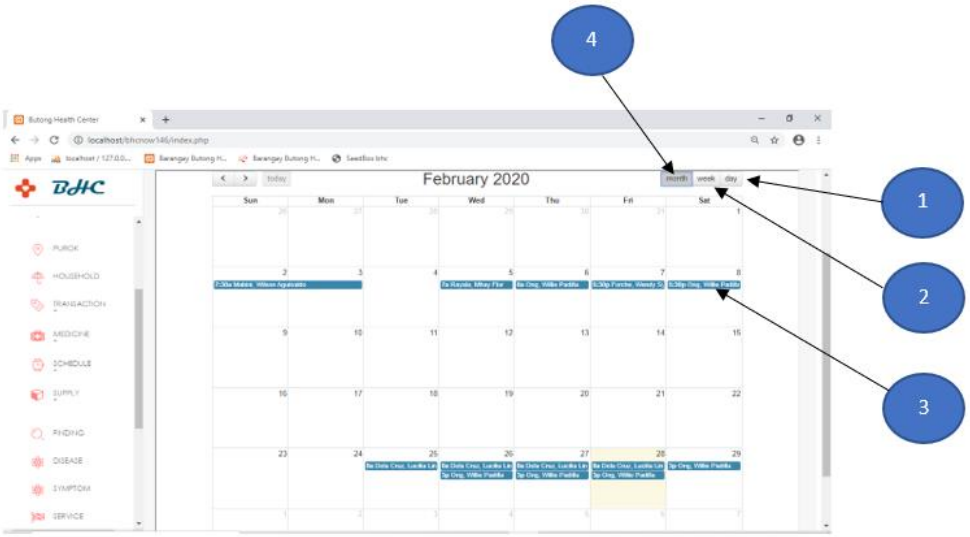

Figure 83. Schedule by month

1. Button Day – Allows the user to see the schedule by day.

2. Button Week – Allows the user to see the schedule by week.

3. Schedule by Month – Allows the user to see who doctor is available for a particular month.

4. Button Month -Allows the user to see who available doctor by month.

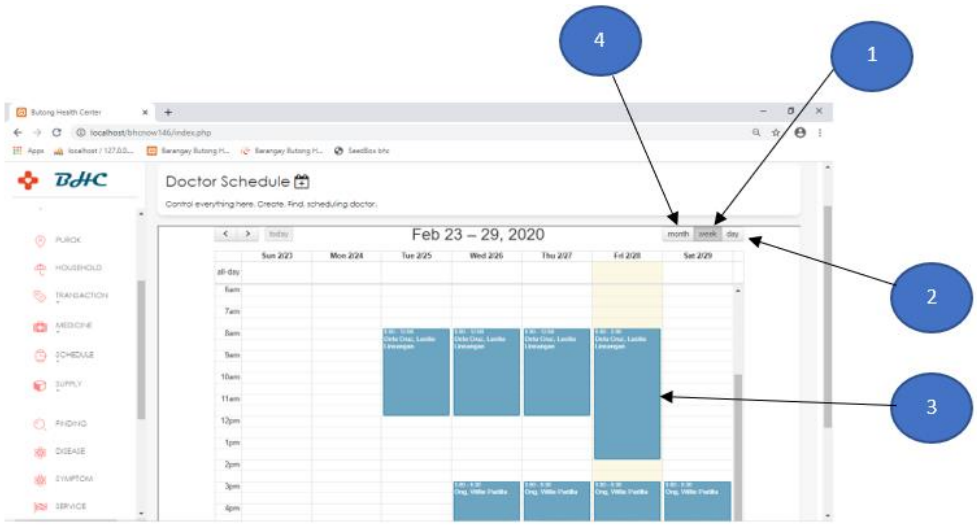

Figure 84. Schedule by week

1. Button Week – Allows the user to see the schedule by week

2. Button Day – Allows the user to see the schedule by day.

3. Schedule by Week – Allows the user to see who doctor is available for a particular week.

4. Button Month -Allows the user to see who available doctor by month.

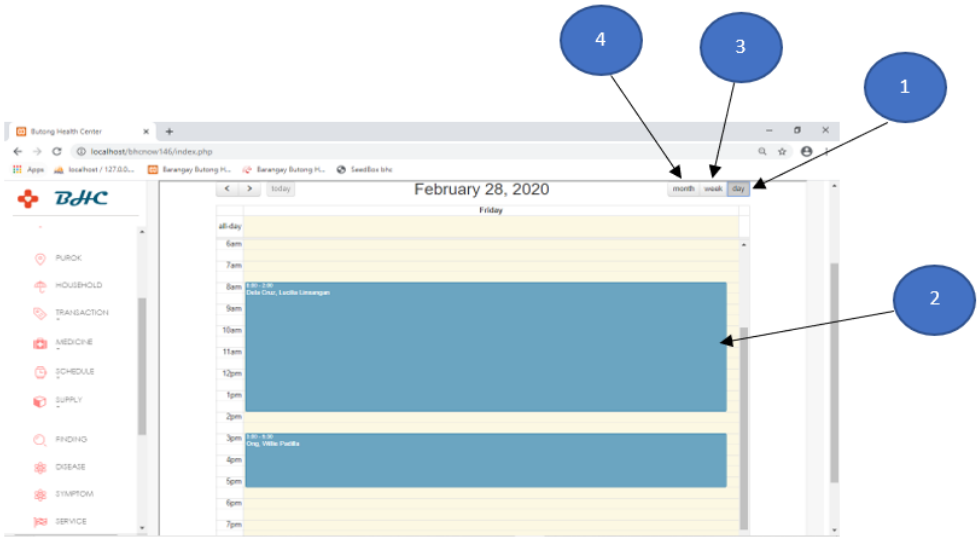

Figure 85. Schedule by day

1. Button Day – Allows the user to see the schedule by day.

2. Schedule by Day – Allows the user to see who doctor is available for a particular day.

3. Button Week – Allows the user to see the schedule by week.

4. Button Month -Allows the user to see who available doctor by month.

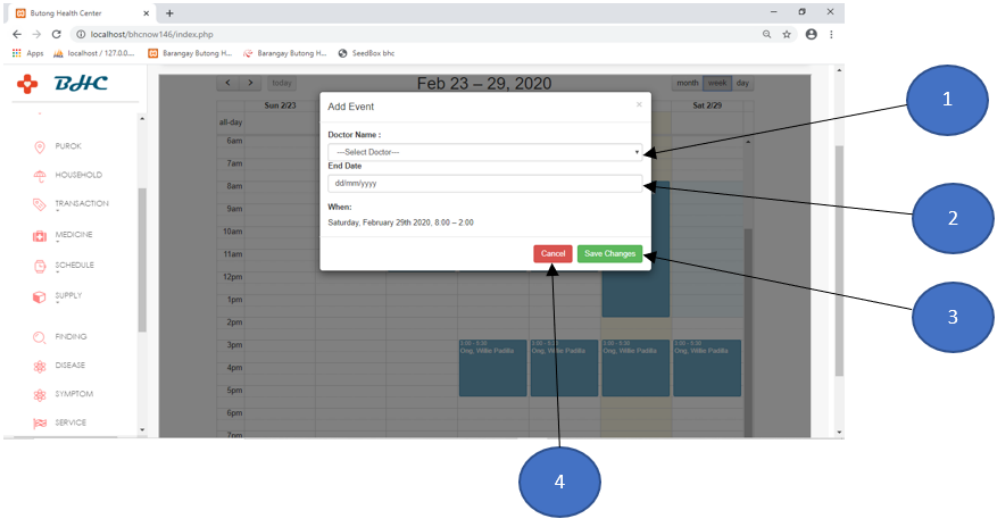

Figure 86. Adding Schedule by Week

1. Dropdown List box – Allows the user to see the available doctor.

2. Event Date – Allows the user to schedule the available doctor according to the date allotted.

3. Button Save – Allows the user to save the schedule for that week.

4. Button Cancel – Allows the user to cancel if the schedule is unavailable.

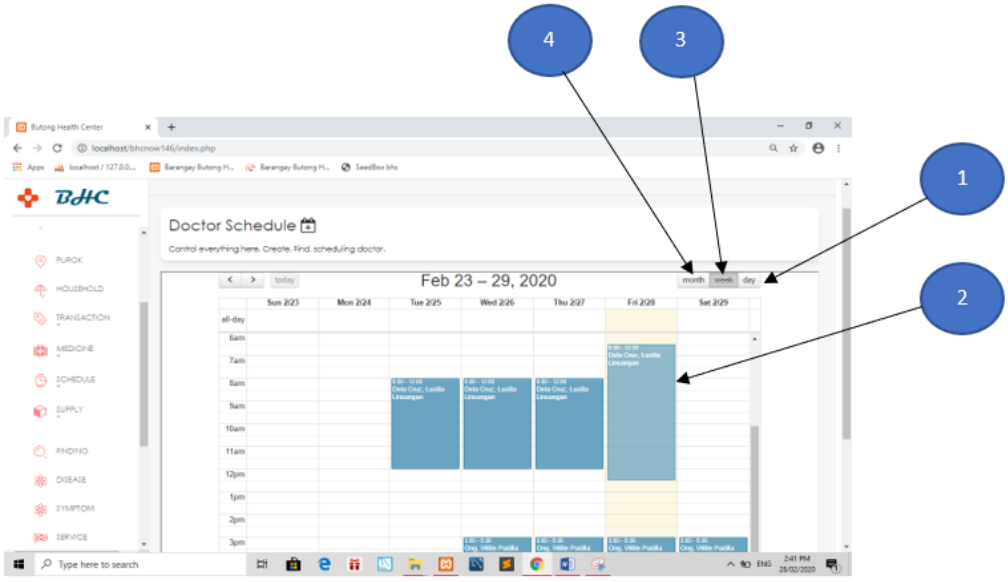

Figure 87. Schedule Update

1. Button Day – Allows the user to see the schedule by day.

2. Update Schedule by Week – Allows the user to update the schedule of doctor by dragging it.

3. Button Week – Allows the user to see the schedule by week.

4. Button Month -Allows the user to see who available doctor by month.

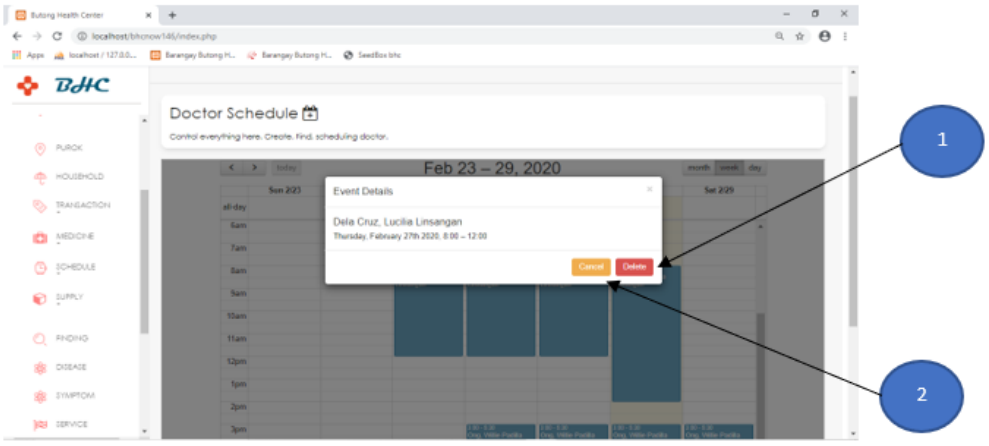

Figure 88. Scheduling Delete

1. Button Delete – Allows the user to delete the schedule if it is unavailable

2. Button Cancel – Allows the user to cancel if you do not delete.

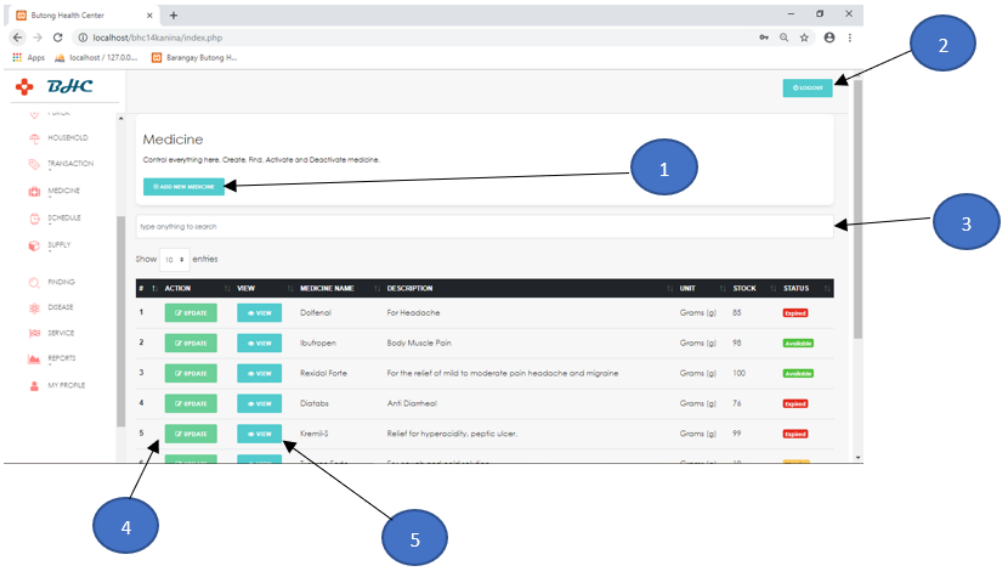

Figure 89. Managing Medicine Inventory

1. **Button Add –** Allows the user to add medicines in the inventory.
2. **Button Logout –** Allows the user to go back to the login page.
3. **Textbox Search –** Allows the user to search what medicine to be find.
4. **Button Update** – Allows the user to update the medicine.
5. **Button View –** Allows the user to view the medicine information.

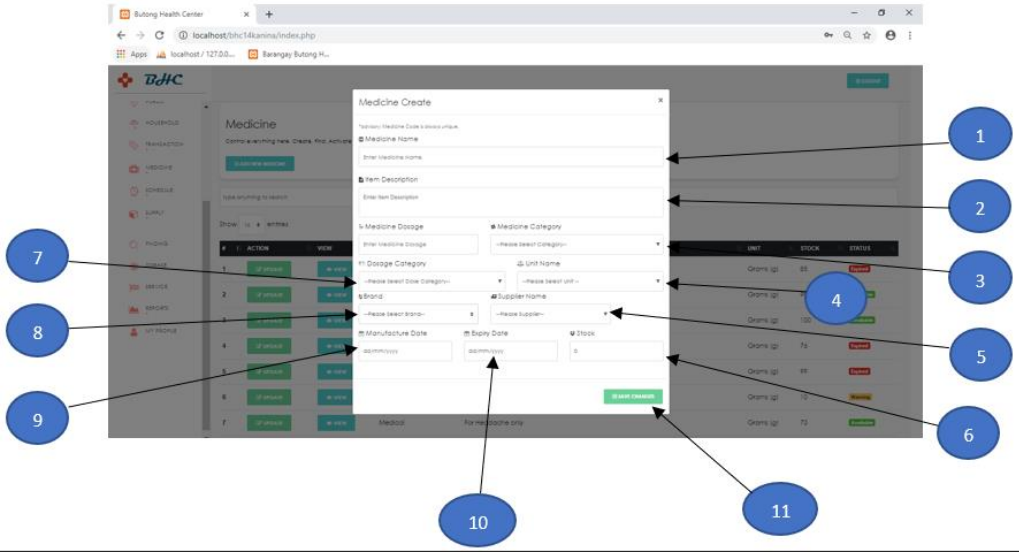

Figure 90. Adding of Medicine in Inventory

1. Textbox Medicine name – Allows the user to fill up medicine name.

2. Textbox Item Description – Allows the user to fill up item description of the medicine.

3. Dropdown List box – Allows the user to choose on what medicine category is appropriate.

4. Dropdown List box – Allows the user to choose on unit name.

5. Dropdown List box – Allows the user to choose on supplier name.

6. Textbox Stock – Allows the user to choose how many stocks will add.

7. Dropdown List box – Allows the user to choose on dosage category.

8. Dropdown List box – Allows the user to choose on brand.

9. Manufacture Date – Allows the user to set the manufacturing date of the medicine.

10. Expiry Date – Allows the user to set the expiry date of the medicine.

11. Button Save – Allows the user to add all medicine and save it in inventory.

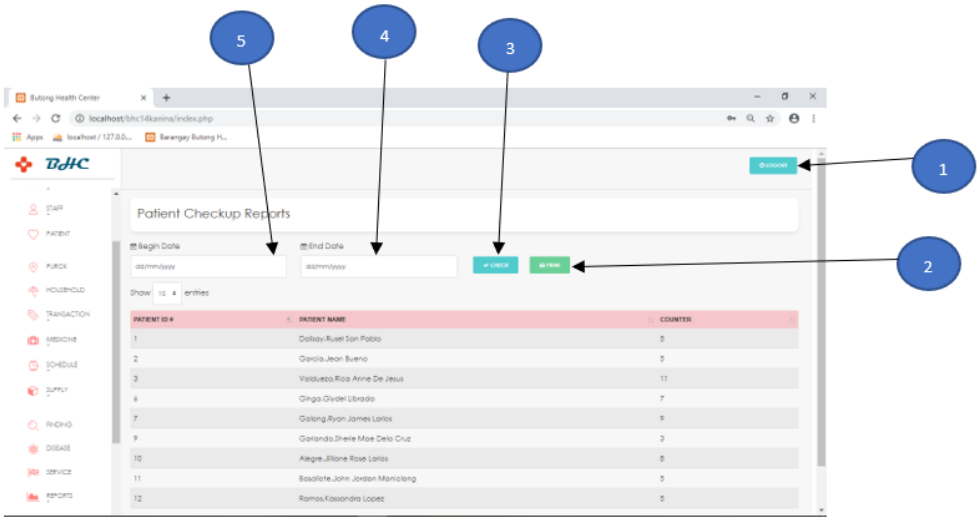

Figure 91. Patient Checkup Reports

1. Button Logout – Allows the user to go back to the login page.

2. Button Print – Allows the user to print the patient checkup reports.

3. Button Check – Allows the user to check the filtered patient checkup reports.

4. End date – Allows the user to choose which end date.

5. Begin Date – Allows the user to choose which begin date.

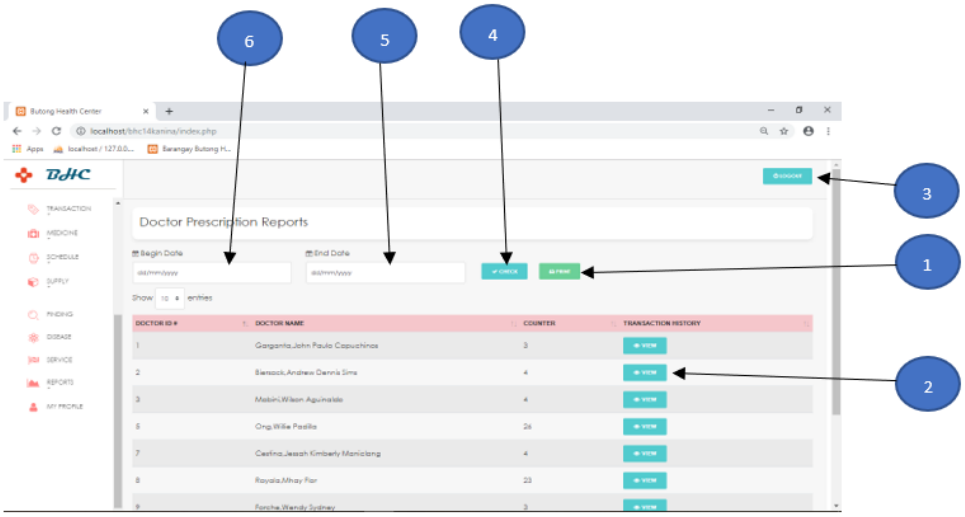

Figure 92. Doctor Prescription Reports

1. Button Print – Allows the user to print the patient doctor prescription reports.

2. Button View – Allows the user to view how many patients are assigned for each doctor.

3. Button Logout – Allows the user to go back to the login page.

4. Button Check – Allows the user to check the filtered doctor prescription reports.

5. End date – Allows the user to choose which end date.

6. Begin Date – Allows the user to choose which begin date.

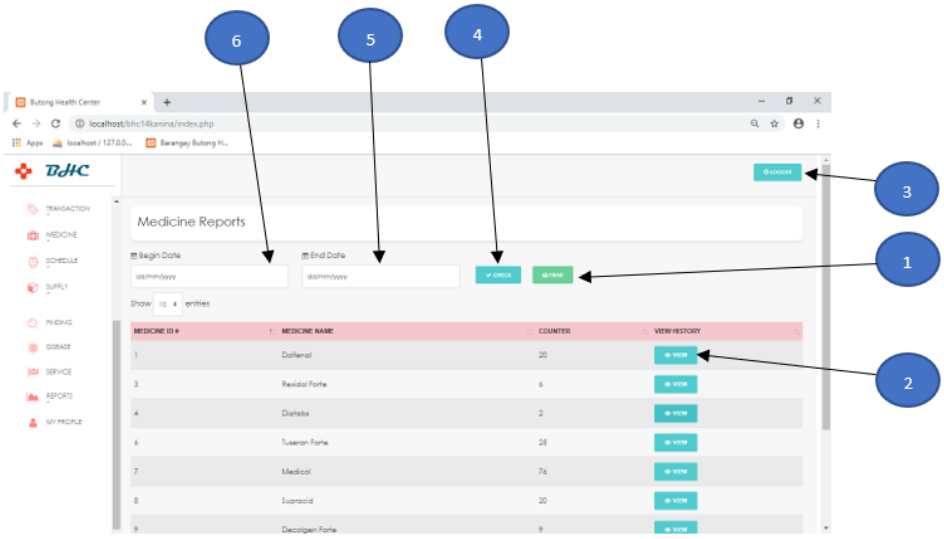

Figure 93. Medicine Reports

1. Button Print – Allows the user to print the medicine reports.

2. Button View – Allows the user to view how many medicines.

3. Button Logout – Allows the user to go back to the login page.

4. Button Check – Allows the user to check the filtered medicine reports.

5. End date – Allows the user to choose which end date.

6. Begin Date – Allows the user to choose which begin date.

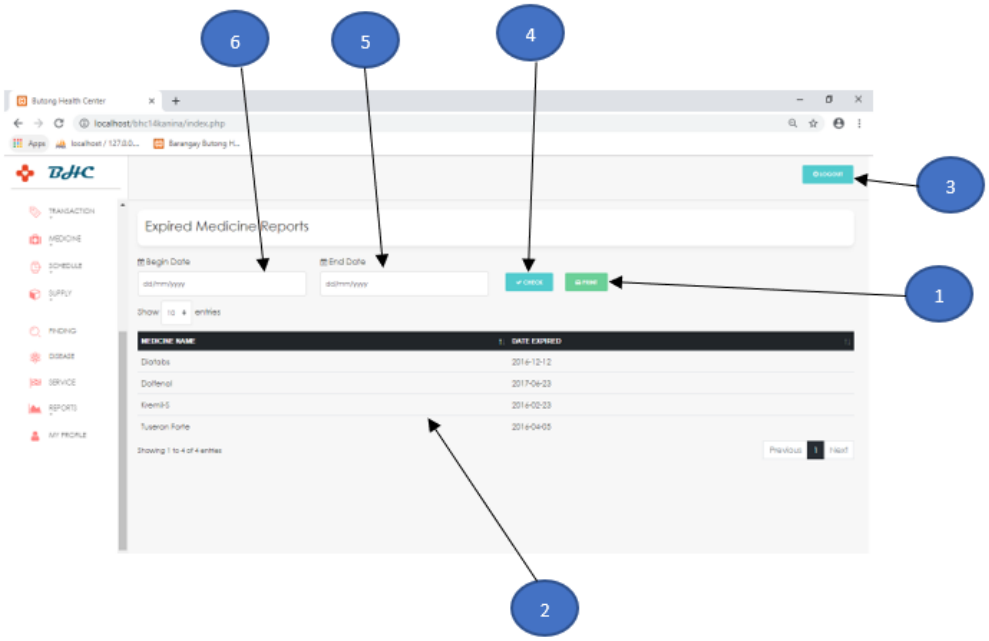

Figure 94. Expired Medicine Reports

1. Button Print – Allows the user to print the expired medicine reports.

2. Medicine Expired Table – Allows the user to see what medicine is already expired.

3. Button Logout - Allows the user to go back to the login page.

4. Button Check - Allows the user to check the filtered medicine reports.

5. End Date – Allows the user to choose which end date.

6. Begin Date – Allow the user to choose which begin date.

Appendix C – Source Code

functions.php

```
class Patient {
    private $db, $patient_id,
$lname, $fname, $mname, $age,
$age_range, $sex, $birthday,
$address, $contact_number,
$email, $height, $weight,
$bloodtype, $bloodpressure,
$temperature, $household_id,
$household_type, $staff_id,
$date_added, $code, $pass,
$status;

    public function
get_patient_credential($type)
{
        $patient_credential;
        switch($type) {
            case "patient_id";

$patient_credential = [$type
=> $this->patient_id];
            break;
            case "lname";

$patient_credential = [$type
=> $this->lname];
            break;
            case "fname";

$patient_credential = [$type
=> $this->fname];
            break;
            case "mname";

$patient_credential = [$type
=> $this->mname];
            break;
            case "age";

$patient_credential = [$type
=> $this->age];
            break;
            case "age_range";

$patient_credential = [$type
=> $this->age_range];
            break;
            case "sex";

$patient_credential = [$type
=> $this->sex];
            break;
            case "birthday";

$patient_credential = [$type
=> $this->birthday];
            break;
            case "address";

$patient_credential = [$type
=> $this->address];
            break;
            case
"contact_number";

$patient_credential = [$type
=> $this->contact_number];
            break;
            case "email";

$patient_credential = [$type
=> $this->email];
            break;
            case "height";

$patient_credential = [$type
=> $this->height];
            break;
            case "weight";

$patient_credential = [$type
=> $this->weight];
            break;
            case "bloodtype";

$patient_credential = [$type
=> $this->bloodtype];
            break;
            case
"bloodpressure";

$patient_credential = [$type
=> $this->bloodpressure];
            break;
            case
"temperature";
```

```
$patient_credential = [$type
=> $this->temperature];
            break;
            case
"household_id";

$patient_credential = [$type
=> $this->household_id];
            break;
            case
"household_type";

$patient_credential = [$type
=> $this->household_type];
            break;
            case "staff_id";

$patient_credential = [$type
=> $this->staff_id];
            break;
            case "date_added";

$patient_credential = [$type
=> $this->date_added];
            break;
            case "code";

$patient_credential = [$type
=> $this->code];
            break;
            case "pass";

$patient_credential = [$type
=> $this->pass];
            break;
            case "status";

$patient_credential = [$type
=> $this->status];
            break;
            case "all";

$patient_credential =
["patient_id" => $this-
>patient_id,
                "lname" =>
$this->lname,
                "fname" =>
$this->fname,
                "mname" =>
$this->mname,
                "age" =>
$this->age,
                "age_range" =>
$this->age_range,
                "sex" =>
$this->sex,
                "birthday" =>
$this->birthday,
                "address" =>
$this->address,

"contact_number" => $this-
>contact_number,
                "email" =>
$this->email,
                "height" =>
$this->height,
                "weight" =>
$this->weight,
                "bloodtype" =>
$this->bloodtype,

"bloodpressure" => $this-
>bloodpressure,
                "temperature"
=> $this->temperature,
                "household_id"
=> $this->household_id,

"household_type" => $this-
>household_type,
                "staff_id" =>
$this->staff_id,
                "date_added"
=> $this->date_added,
                "code" =>
$this->code,
                "pass" =>
$this->pass,
                "status" =>
$this->status];
            break;
            default:

$patient_credential = null;
        }
        return
$patient_credential;
```

```
    }

    public function
set_patient_credential($type,$
value) {
        switch($type) {
            case "patient_id";
                $this-
>patient_id = $value;
            break;
            case "lname";
                $this->lname =
$value;
            break;
            case "fname";
                $this->fname =
$value;
            break;
            case "mname";
                $this->mname =
$value;
            break;
            case "age";
                $this->age =
$value;
            break;
            case "age_range";
                $this-
>age_range = $value;
            break;
            case "sex";
                $this->sex =
$value;
            break;
            case "birthday";
                $this-
>birthday = $value;
            break;
            case "address";
                $this->address
= $value;
            break;
            case
"contact_number";
                $this-
>contact_number = $value;
            break;
            case "email";
                $this->email =
$value;
            break;
            case "height";
                $this->height
= $value;
            break;
            case "weight";
                $this->weight
= $value;
            break;
            case "bloodtype";
                $this-
>bloodtype = $value;
            break;
            case
"bloodpressure";
                $this-
>bloodpressure = $value;
            break;
            case
"temperature";
                $this-
>temperature = $value;
            break;
            case
"household_id";
                $this-
>household_id = $value;
            break;
            case
"household_type";
                $this-
>household_type = $value;
            break;
            case "staff_id";
                $this-
>staff_id = $value;
            break;
            case "date_added";
                $this-
>date_added = $value;
            break;
            case "code";
                $this->code =
$value;
            break;
            case "pass";
                $this->pass =
$value;
            break;
            case "status";
```

```
                $this->status
= $value;
            break;
            case "all";
                $this-
>patient_id =
$value["patient_id"];
                $this->lname =
$value["lname"];
                $this->fname =
$value["fname"];
                $this->mname =
$value["mname"];
                $this->age =
$value["age"];
                $this-
>age_range =
$value["age_range"];
                $this->sex =
$value["sex"];
                $this-
>birthday =
$value["birthday"];
                $this->address
= $value["address"];
                $this-
>contact_number =
$value["contact_number"];
                $this->email =
$value["email"];
                $this->height
= $value["height"];
                $this->weight
= $value["weight"];
                $this-
>bloodtype =
$value["bloodtype"];
                $this-
>bloodpressure =
$value["bloodpressure"];
                $this-
>temperature =
$value["temperature"];
                $this-
>household_id =
$value["household_id"];
                $this-
>household_type =
$value["household_type"];
                $this-
>staff_id =
$value["staff_id"];
                $this-
>date_added =
$value["date_added"];
                $this->code =
$value["code"];
                $this->pass =
$value["pass"];
                $this->status
= $value["status"];
            break;
        }
    }

    public function
startcreatepatient() {
        $status = true;
        if ($this->lname ==
null) $status = false;
        else if ($this->fname
== null) $status = false;
        else if ($this->mname
== null) $status = false;
        else if ($this->age ==
null) $status = false;
        else if ($this-
>age_range == null) $status =
false;
        else if ($this->sex ==
null) $status = false;
        else if ($this-
>birthday == null) $status =
false;
        else if ($this-
>address == null) $status =
false;
        else if ($this-
>contact_number == null)
$status = false;
        else if ($this->email
== null) $status = false;
        else if ($this->height
== null) $status = false;
        else if ($this->weight
== null) $status = false;
        else if ($this-
>bloodtype == null) $status =
false;
```

```
        else if ($this-
>bloodpressure == null)
$status = false;
        else if ($this-
>temperature == null) $status
= false;
        else if ($this-
>household_id == null) $status
= false;
        else if ($this-
>household_type == null)
$status = false;
        else if ($this-
>staff_id == null) $status =
false;
        else if ($this-
>date_added == null) $status =
false;
        else if ($this->code
== null) $status = false;
        else if ($this->pass
== null) $status = false;
        else if ($this->status
== null) $status = false;
        else $this-
>createpatient();
        return $status;
    }

    public function
startcreatepatient1() {
        $status = true;
        if ($this->lname ==
null) $status = false;
        else if ($this->fname
== null) $status = false;
        else if ($this->mname
== null) $status = false;
        else if ($this->age ==
null) $status = false;
        else if ($this->sex ==
null) $status = false;
        else if ($this-
>birthday == null) $status =
false;
        else if ($this-
>address == null) $status =
false;
        else if ($this-
>contact_number == null)
$status = false;
        else if ($this->email
== null) $status = false;
        else if ($this-
>staff_id == null) $status =
false;
        else if ($this->code
== null) $status = false;
        else if ($this->pass
== null) $status = false;
        else if ($this->status
== null) $status = false;
        else $this-
>createpatient();
        return $status;
    }

    public function
startreadpatient() {
        $this->patient_id =
[];
        $this->lname = [];
        $this->fname = [];
        $this->mname = [];
        $this->age = [];
        $this->age_range = [];
        $this->sex = [];
        $this->birthday = [];
        $this->address = [];
        $this->contact_number
= [];
        $this->email = [];
        $this->height = [];
        $this->weight = [];
        $this->bloodtype = [];
        $this->bloodpressure =
[];
        $this->temperature =
[];
        $this->household_id =
[];
        $this->household_type
= [];
        $this->staff_id = [];
        $this->date_added =
[];
        $this->code = [];
        $this->pass = [];
```

```
        $this->status = [];
        $this->readpatient();
    }

    public function
startreadpatientbyid($id) {
        $this->patient_id =
[];
        $this->lname = [];
        $this->fname = [];
        $this->mname = [];
        $this->age = [];
        $this->age_range = [];
        $this->sex = [];
        $this->birthday = [];
        $this->address = [];
        $this->contact_number
= [];
        $this->email = [];
        $this->height = [];
        $this->weight = [];
        $this->bloodtype = [];
        $this->bloodpressure =
[];
        $this->temperature =
[];
        $this->household_id =
[];
        $this->household_type
= [];
        $this->staff_id = [];
        $this->date_added =
[];
        $this->code = [];
        $this->pass = [];
        $this->status = [];
        $this-
>readpatientbyid($id);
    }

    public function
startreadpatientbydocid($id) {
        $this->patient_id =
[];
        $this->lname = [];
        $this->fname = [];
        $this->mname = [];
        $this->age = [];
        $this->age_range = [];
        $this->sex = [];
        $this->birthday = [];

        $this-
>readpatientbydocid($id);
    }
    public function
startreadpatientbymedid($id) {
        $this->patient_id =
[];
        $this->lname = [];
        $this->fname = [];
        $this->mname = [];
        $this->age = [];
        $this->age_range = [];
        $this->sex = [];
        $this->birthday = [];
        $this->pass = [];
        $this->status = [];

        $this-
>readpatientbymedid($id);
    }

    public function
startfindpatient($like) {
        $this->patient_id =
[];
        $this->lname = [];
        $this->fname = [];
        $this->mname = [];
        $this->age = [];
        $this->age_range = [];
        $this->sex = [];
        $this->birthday = [];
        $this->address = [];
        $this->contact_number
= [];
        $this->email = [];
        $this->height = [];
        $this->weight = [];
        $this->bloodtype = [];
        $this->bloodpressure =
[];
        $this->temperature =
[];
        $this->household_id =
[];
        $this->household_type
= [];
        $this->staff_id = [];
```

```
        $this->date_added =
[];
        $this->code = [];
        $this->pass = [];
        $this->status = [];
        $this-
>findpatient($like);
    }

    public function
startupdatepasswordpatient() {
        $status = true;
        $this-
>updatepatientbypassword();
        return $status;
    }

    public function
startupdatepatientbydetail() {
        $status = true;
        $this-
>updatepatientbydetail();
        return $status;
    }

    public function
startupdatepatientbyrecorddeta
il() {
        $status = true;
        $this-
>updatepatientbyrecorddetail()
;
        return $status;
    }

    public function
startupdatepatient() {
        $status = true;
        if ($this->patient_id
== null) $status = false;
        else if ($this->lname
== null) $status = false;
        else if ($this->fname
== null) $status = false;
        else if ($this->mname
== null) $status = false;
        else if ($this->age ==
null) $status = false;
        else if ($this-
>age_range == null) $status =
false;
        else if ($this->sex ==
null) $status = false;
        else if ($this-
>birthday == null) $status =
false;
        else if ($this-
>address == null) $status =
false;
        else if ($this-
>contact_number == null)
$status = false;
        else if ($this->email
== null) $status = false;
        else if ($this->height
== null) $status = false;
        else if ($this->weight
== null) $status = false;
        else if ($this-
>bloodtype == null) $status =
false;
        else if ($this-
>bloodpressure == null)
$status = false;
        else if ($this-
>temperature == null) $status
= false;
        else if ($this-
>household_id == null) $status
= false;
        else if ($this-
>household_type == null)
$status = false;
        else if ($this-
>staff_id == null) $status =
false;
        else if ($this-
>date_added == null) $status =
false;
        else if ($this->code
== null) $status = false;
        else if ($this->pass
== null) $status = false;
        else if ($this->status
== null) $status = false;
        else $this-
>updatepatient();
        return $status;
```

```
    }

    public function
__construct() {
        $this->db = new
PDO("mysql:host=localhost;dbna
me=bhc_db;charset=utf8mb4",
"root","");
        $this->db-
>setAttribute(PDO::ATTR_ERRMOD
E, PDO::ERRMODE_EXCEPTION);
        $this->db-
>setAttribute(PDO::ATTR_EMULAT
E_PREPARES, false);
    }

    protected function
createpatient() {
        try {
            $this->db-
>beginTransaction();
            $stmt = $this->db-
>prepare("INSERT INTO patient
( patient_id,  lname,  fname,
mname,  age, age_range, sex,
birthday,  address,
contact_number, email, height,
weight,  bloodtype,
bloodpressure,  temperature,
household_id,  household_type,
staff_id,  date_added,  code,
pass,  status) VALUES ( null,
:lname,  :fname,  :mname,
:age, :age_range, :sex,
:birthday,
:address,:contact_number,
:email,  :height,  :weight,
:bloodtype,  :bloodpressure,
:temperature,  :household_id,
:household_type,  :staff_id,
:date_added,  :code,  :pass,
:status)");
            $stmt-
>execute(array("lname" =>
$this->lname, "fname" =>
$this->fname, "mname" =>
$this->mname, "age" => $this-
>age, "age_range" => $this-
>age_range,
                "sex" =>
$this->sex, "birthday" =>
$this->birthday, "address" =>
$this->address,

"contact_number" => $this-
>contact_number, "email"
=>$this->email, "height" =>
$this->height, "weight" =>
$this->weight, "bloodtype" =>
$this->bloodtype,
"bloodpressure" => $this-
>bloodpressure, "temperature"
=> $this->temperature,
"household_id" => $this-
>household_id,
"household_type" => $this-
>household_type, "staff_id" =>
$this->staff_id, "date_added"
=> $this->date_added, "code"
=> $this->code, "pass" =>
$this->pass, "status" =>
$this->status));
            $affected_rows =
$stmt->rowCount();
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

    protected function
readpatient() {
        try {
            $this->db-
>beginTransaction();
            foreach($this->db-
>query("SELECT * FROM patient
order by patient_id desc") as
$row) {

array_push($this->patient_id,
$row["patient_id"]);
```

```
array_push($this->lname,
$row["lname"]);

array_push($this->fname,
$row["fname"]);

array_push($this->mname,
$row["mname"]);

array_push($this->age,
$row["age"]);

array_push($this->age_range,
$row["age_range"]);

array_push($this->sex,
$row["sex"]);

array_push($this->birthday,
$row["birthday"]);

array_push($this->address,
$row["address"]);

array_push($this-
>contact_number,
$row["contact_number"]);

array_push($this->email,
$row["email"]);

array_push($this->height,
$row["height"]);

array_push($this->weight,
$row["weight"]);

array_push($this->bloodtype,
$row["bloodtype"]);

array_push($this-
>bloodpressure,
$row["bloodpressure"]);

array_push($this->temperature,
$row["temperature"]);

array_push($this-
>household_id,
$row["household_id"]);

array_push($this-
>household_type,
$row["household_type"]);

array_push($this->staff_id,
$row["staff_id"]);

array_push($this->date_added,
$row["date_added"]);

array_push($this->code,
$row["code"]);

array_push($this->pass,
$row["pass"]);

array_push($this->status,
$row["status"]);
            }
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

    protected function
readpatientbyid($id) {
        try {
            $this->db-
>beginTransaction();
            foreach($this->db-
>query("SELECT * FROM patient
where patient_id=$id") as
$row) {

array_push($this->patient_id,
$row["patient_id"]);

array_push($this->lname,
$row["lname"]);
```

```
array_push($this->fname,
$row["fname"]);

array_push($this->mname,
$row["mname"]);

array_push($this->age,
$row["age"]);

array_push($this->age_range,
$row["age_range"]);

array_push($this->sex,
$row["sex"]);

array_push($this->birthday,
$row["birthday"]);

array_push($this->address,
$row["address"]);

array_push($this-
>contact_number,
$row["contact_number"]);

array_push($this->email,
$row["email"]);

array_push($this->height,
$row["height"]);

array_push($this->weight,
$row["weight"]);

array_push($this->bloodtype,
$row["bloodtype"]);

array_push($this-
>bloodpressure,
$row["bloodpressure"]);

array_push($this->temperature,
$row["temperature"]);

array_push($this-
>household_id,
$row["household_id"]);

array_push($this-
>household_type,
$row["household_type"]);

array_push($this->staff_id,
$row["staff_id"]);

array_push($this->date_added,
$row["date_added"]);

array_push($this->code,
$row["code"]);

array_push($this->pass,
$row["pass"]);

array_push($this->status,
$row["status"]);
            }
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

    protected function
readpatientbydocid($id) {
        try {
            $this->db-
>beginTransaction();
            foreach($this->db-
>query("SELECT
p.patient_id,p.lname,p.fname,p
.mname,d.lname dlname,d.fname
dfname,d.mname
dmname,r.date_added FROM
patient p join record r on
p.patient_id = r.patient_id
join finding ff on
ff.record_id = r.record_id
join doctor d on d.doctor_id =
r.doctor_id WHERE d.doctor_id
=$id") as $row) {
```

```
array_push($this->patient_id,
$row["patient_id"]);

array_push($this->lname,
$row["lname"]);

array_push($this->fname,
$row["fname"]);

array_push($this->mname,
$row["mname"]);

array_push($this->age,
$row["dlname"]);

array_push($this->age_range,
$row["dmname"]);

array_push($this->sex,
$row["dfname"]);

array_push($this->birthday,
$row["date_added"]);



            }
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

    protected function
readpatientbymedid($id) {
        try {
            $this->db-
>beginTransaction();
            foreach($this->db-
>query("SELECT
p.patient_id,p.lname,p.fname,p
.mname,d.lname dlname,d.fname
dfname,d.mname
dmname,r.date_added,ff.medicin
e_id,ff.quantity FROM patient
p join record r on
p.patient_id = r.patient_id
join finding ff on
ff.record_id = r.record_id
join doctor d on d.doctor_id =
r.doctor_id WHERE
ff.medicine_id  =$id") as
$row) {


array_push($this->patient_id,
$row["patient_id"]);

array_push($this->lname,
$row["lname"]);

array_push($this->fname,
$row["fname"]);

array_push($this->mname,
$row["mname"]);

array_push($this->age,
$row["dlname"]);

array_push($this->age_range,
$row["dmname"]);

array_push($this->sex,
$row["dfname"]);

array_push($this->birthday,
$row["date_added"]);

array_push($this->pass,
$row["medicine_id"]);

array_push($this->status,
$row["quantity"]);



            }
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
```

```
            $this->db->rollBack();
            echo $ex->getMessage();
        }
    }

    protected function findpatient($like) {
        try {
            $this->db->beginTransaction();
            foreach($this->db->query("SELECT * FROM patient WHERE  patient_id LIKE '%" . $like . "%'
                OR  lname LIKE '%" . $like . "%'
                OR  fname LIKE '%" . $like . "%'
                OR  mname LIKE '%" . $like . "%'
                OR  age LIKE '%" . $like . "%'
                OR  age_range LIKE '%" . $like . "%'
                OR  sex LIKE '%" . $like . "%'
                OR  birthday LIKE '%" . $like . "%'
                OR  address LIKE '%" . $like . "%'
                OR  contact_number LIKE '%" . $like . "%'
                OR  email LIKE '%" . $like . "%'
                OR  height LIKE '%" . $like . "%'
                OR  weight LIKE '%" . $like . "%'
                OR  bloodtype LIKE '%" . $like . "%'
                OR  bloodpressure LIKE '%" . $like . "%'
                OR  temperature LIKE '%" . $like . "%'
                OR  household_id LIKE '%" . $like . "%'
                OR  household_type LIKE '%" . $like . "%'
                OR  staff_id LIKE '%" . $like . "%'
                OR  date_added LIKE '%" . $like . "%'
                OR  code LIKE '%" . $like . "%'
                OR  pass LIKE '%" . $like . "%'
                OR  status LIKE '%" . $like . "%' ") as $row) {

                array_push($this->patient_id, $row["patient_id"]);

                array_push($this->lname, $row["lname"]);

                array_push($this->fname, $row["fname"]);

                array_push($this->mname, $row["mname"]);

                array_push($this->age, $row["age"]);

                array_push($this->age_range, $row["age_range"]);

                array_push($this->sex, $row["sex"]);

                array_push($this->birthday, $row["birthday"]);

                array_push($this->address, $row["address"]);

                array_push($this->contact_number, $row["contact_number"]);

                array_push($this->email, $row["email"]);
```

```
array_push($this->height,
$row["height"]);

array_push($this->weight,
$row["weight"]);

array_push($this->bloodtype,
$row["bloodtype"]);

array_push($this-
>bloodpressure,
$row["bloodpressure"]);

array_push($this->temperature,
$row["temperature"]);

array_push($this-
>household_id,
$row["household_id"]);

array_push($this-
>household_type,
$row["household_type"]);

array_push($this->staff_id,
$row["staff_id"]);

array_push($this->date_added,
$row["date_added"]);

array_push($this->code,
$row["code"]);

array_push($this->pass,
$row["pass"]);

array_push($this->status,
$row["status"]);
            }
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

    protected function
updatepatientbypassword() {
        try {
            $this->db-
>beginTransaction();
            $stmt = $this->db-
>prepare("UPDATE patient SET
code = :code, pass = :pass
                WHERE
patient_id = :patient_id");

        $stmt-
>execute(array("code" =>
$this->code, "pass" => $this-
>pass, "patient_id" => $this-
>patient_id));
            $affected_rows =
$stmt->rowCount();
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

    protected function
updatepatientbydetail() {
        try {
            $this->db-
>beginTransaction();
            $stmt = $this->db-
>prepare("UPDATE patient SET
                height =
:height, weight = :weight,
bloodpressure =
:bloodpressure, temperature =
:temperature
                WHERE
patient_id = :patient_id");

            $stmt-
>execute(array("patient_id" =>
$this->patient_id,
```

```
            "height" =>
$this->height,
            "weight" =>
$this->weight,
            "bloodpressure"
=> $this->bloodpressure,
            "temperature" =>
$this->temperature));
            $affected_rows =
$stmt->rowCount();
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

    protected function
updatepatientbyrecorddetail()
{
        try {
            $this->db-
>beginTransaction();
            $stmt = $this->db-
>prepare("UPDATE patient SET
                height =
:height, weight = :weight,
bloodpressure =
:bloodpressure,  temperature =
:temperature
                WHERE
patient_id = :patient_id");

            $stmt-
>execute(array("patient_id" =>
$this->patient_id, "height" =>
$this->height, "weight" =>
$this->weight, "bloodpressure"
=> $this->bloodpressure,
"temperature" => $this-
>temperature
                ));
            $affected_rows =
$stmt->rowCount();
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

    protected function
updatepatient() {
        try {
            $this->db-
>beginTransaction();
            $stmt = $this->db-
>prepare("UPDATE patient SET
lname = :lname,  fname =
:fname,  mname = :mname,  age
= :age,  age_range =
:age_range,
                sex = :sex,
birthday = :birthday,  address
= :address, contact_number =
:contact_number, email =
:email,
                height =
:height,  weight = :weight,
bloodtype = :bloodtype,
bloodpressure =
:bloodpressure,
                temperature =
:temperature,  household_id =
:household_id,  household_type
= :household_type,  staff_id =
:staff_id,  date_added =
:date_added,  code = :code,
pass = :pass,  status =
:status WHERE patient_id =
:patient_id");

            $stmt-
>execute(array("patient_id" =>
$this->patient_id, "lname" =>
$this->lname, "fname" =>
$this->fname, "mname" =>
$this->mname, "age" => $this-
>age,  "age_range" => $this-
```

```
>age_range, "sex" => $this-
>sex, "birthday" => $this-
>birthday, "address" => $this-
>address, "contact_number" =>
$this->contact_number, "email"
=>$this->email, "height" =>
$this->height, "weight" =>
$this->weight, "bloodtype" =>
$this->bloodtype,
"bloodpressure" => $this-
>bloodpressure, "temperature"
=> $this->temperature,
"household_id" => $this-
>household_id,
"household_type" => $this-
>household_type, "staff_id" =>
$this->staff_id, "date_added"
=> $this->date_added, "code"
=> $this->code, "pass" =>
$this->pass, "status" =>
$this->status));
            $affected_rows =
$stmt->rowCount();
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }
}
```

```
class Medicine {
      private $db,
  $medicine_id, $medicine_name,
  $item_description, $dosage,
  $unit_id,
  $medicine_category_id,
      $dosecategory_id, $brand,
  $supplier_id,
  $manufacture_date,
  $expiry_date,$stock,
  $quantity,
  $status,$currentDateTime;

      public function
  get_medicine_credential($type
  ) {
          $medicine_credential;
          switch($type) {
              case
  "medicine_id";

  $medicine_credential = [$type
  => $this->medicine_id];
              break;
              case
  "medicine_name";

  $medicine_credential = [$type
  => $this->medicine_name];
              break;
              case
  "item_description";

  $medicine_credential = [$type
  => $this->item_description];
              break;
              case "dosage";

  $medicine_credential = [$type
  => $this->dosage];
              break;
               case "unit_id";

  $medicine_credential = [$type
  => $this->unit_id];
              break;
              case
  "medicine_category_id";

  $medicine_credential = [$type
  => $this-
  >medicine_category_id];
              break;
              case
  "dosecategory_id";

  $medicine_credential = [$type
  => $this->dosecategory_id];
              break;
              case "brand";
```

```
$medicine_credential = [$type
=> $this->brand];
            break;
            case
"supplier_id";

$medicine_credential = [$type
=> $this->supplier_id];
            break;
            case
"manufacture_date";

$medicine_credential = [$type
=> $this->manufacture_date];
            break;
            case
"expiry_date";

$medicine_credential = [$type
=> $this->expiry_date];
            break;
            case "stock";

$medicine_credential = [$type
=> $this->stock];
            break;
            case "quantity";

$medicine_credential = [$type
=> $this->quantity];
            break;
            case "status";

$medicine_credential = [$type
=> $this->status];
            break;
            case "all";

$medicine_credential =
["medicine_id" => $this-
>medicine_id, "medicine_name"
=> $this->medicine_name,

"item_description" => $this-
>item_description, "dosage"
=> $this->dosage,
                "unit_id" =>
$this->unit_id,
"medicine_category_id" =>
$this->medicine_category_id,

"dosecategory_id" => $this-
>dosecategory_id, "brand" =>
$this->brand,
                "supplier_id"
=> $this->supplier_id,
"manufacture_date" => $this-
>manufacture_date,
                "expiry_date"
=> $this->expiry_date,
"stock" => $this->stock,
                "quantity" =>
$this->quantity, "status" =>
$this->status];
            break;
            default:

$medicine_credential = null;
        }
        return
$medicine_credential;
    }

    public function
set_medicine_credential($type
,$value) {
        switch($type) {
            case
"medicine_id";
                $this-
>medicine_id = $value;
            break;
            case
"medicine_name";
                $this-
>medicine_name = $value;
            break;
            case
"item_description";
                $this-
>item_description = $value;
            break;
            case "dosage";
                $this->dosage
= $value;
            break;
            case "unit_id";
```

```
                $this->unit_id = $value;
                break;
                case "medicine_category_id";
                $this->medicine_category_id = $value;
                break;
                case "dosecategory_id";
                $this->dosecategory_id = $value;
                break;
                case "brand";
                    $this->brand = $value;
                break;
                case "supplier_id";
                $this->supplier_id = $value;
                break;
                case "manufacture_date";
                $this->manufacture_date = $value;
                break;
                case "expiry_date";
                $this->expiry_date = $value;
                break;
                case "stock";
                    $this->stock = $value;
                break;
                case "quantity";
                    $this->quantity = $value;
                break;
                case "status";
                    $this->status = $value;
                break;
                case "all";
                    $this->medicine_id = $value["medicine_id"];
                $this->medicine_name = $value["medicine_name"];
                $this->item_description = $value["item_description"];
                $this->dosage = $value["dosage"];
                $this->unit_id = $value["unit_id"];
                $this->medicine_category_id = $value["medicine_category_id"];
                $this->dosecategory_id = $value["dosecategory_id"];
                $this->brand = $value["brand"];
                $this->supplier_id = $value["supplier_id"];
                $this->manufacture_date = $value["manufacture_date"];
                $this->expiry_date = $value["expiry_date"];
                $this->stock = $value["stock"];
                $this->quantity = $value["quantity"];
                $this->status = $value["status"];
                    break;
            }
        }

        public function startcreatemedicine() {
            $status = true;
            if ($this->medicine_name == null) $status = false;
            else if ($this->item_description == null) $status = false;
```

```
        else if ($this->dosage== null) $status = false;
        else if ($this->unit_id== null) $status = false;
        else if ($this->medicine_category_id== null) $status = false;
        else if ($this->dosecategory_id== null) $status = false;
        else if ($this->brand== null) $status = false;
        else if ($this->supplier_id== null) $status = false;
        else if ($this->manufacture_date == null) $status = false;
        else if ($this->expiry_date == null) $status = false;
        else if ($this->stock == null) $status = false;
        else if ($this->quantity == null) $status = false;
        else if ($this->status == null) $status = false;
        else $this->createmedicine();
        return $status;
    }

    public function startreadmedicine() {
        $this->medicine_id = [];
        $this->medicine_name = [];
        $this->item_description = [];
        $this->dosage = [];
        $this->unit_id = [];
        $this->medicine_category_id = [];
        $this->dosecategory_id = [];
        $this->brand = [];
        $this->supplier_id = [];
        $this->manufacture_date = [];
        $this->expiry_date = [];
        $this->stock = [];
        $this->quantity = [];
        $this->status = [];
        $this->readmedicine();
    }

    public function startfindmedicine($like) {
        $this->medicine_id = [];
        $this->medicine_name = [];
        $this->item_description = [];
        $this->dosage = [];
        $this->unit_id = [];
        $this->medicine_category_id = [];
        $this->dosecategory_id = [];
        $this->brand = [];
        $this->supplier_id = [];
        $this->manufacture_date = [];
        $this->expiry_date = [];
        $this->stock = [];
        $this->quantity = [];
        $this->status = [];
        $this->findmedicine($like);
    }

    public function startupdatemedicine() {
        $status = true;
        $this->updatemedicine();
        return $status;
```

```
    }

    public function
startupdatemedicinebypullout(
$id) {
        $status = true;
       $this-
>updatemedicine();
        return $status;
    }

     public function
startupdatemedicinebystock()
{
        $status = true;
        if ($this-
>medicine_id == null) $status
= false;
        else if ($this->stock
== null) $status = false;
        else if ($this-
>expiry_date == null) $status
= false;
        else $this-
>updatemedicinebystock();
        return $status;
    }

    public function
__construct() {
        $this->db = new
PDO("mysql:host=localhost;dbn
ame=bhc_db;charset=utf8mb4",
"root","");
        $this->db-
>setAttribute(PDO::ATTR_ERRMO
DE, PDO::ERRMODE_EXCEPTION);
        $this->db-
>setAttribute(PDO::ATTR_EMULA
TE_PREPARES, false);
    }

    protected function
createmedicine() {
        try {
            $this->db-
>beginTransaction();
            $stmt = $this-
>db->prepare("INSERT INTO
medicine ( medicine_id,
medicine_name,
item_description,  dosage,
unit_id,
medicine_category_id,
dosecategory_id, brand,
supplier_id,
manufacture_date,
expiry_date,  stock,
quantity, status) VALUES (
null,  :medicine_name,
:item_description, :dosage,
:unit_id,:medicine_category_i
d, :dosecategory_id, :brand,
:supplier_id,
:manufacture_date,
:expiry_date,  :stock,
:quantity, :status)");

            $stmt-
>execute(array(

"medicine_name" => $this-
>medicine_name,

"item_description" => $this-
>item_description,
                "dosage" =>
$this->dosage,
                "unit_id" =>
$this->unit_id,

"medicine_category_id" =>
$this->medicine_category_id,

"dosecategory_id" => $this-
>dosecategory_id,
                "brand" =>
$this->brand,
                "supplier_id"
=> $this->supplier_id,

"manufacture_date" => $this-
>manufacture_date,
                "expiry_date"
=> $this->expiry_date,
                "stock" =>
$this->stock,
                "quantity" =>
$this->quantity,
```

```
                "status" =>
$this->status));
            $affected_rows =
$stmt->rowCount();
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

    protected function
readmedicine() {
        try {
            $this->db-
>beginTransaction();
            foreach($this-
>db->query("SELECT * FROM
medicine") as $row) {

array_push($this-
>medicine_id,$row["medicine_i
d"]);

array_push($this-
>medicine_name,$row["medicine
_name"]);

array_push($this-
>item_description,$row["item_
description"]);

array_push($this-
>dosage,$row["dosage"]);

array_push($this-
>unit_id,$row["unit_id"]);

array_push($this-
>medicine_category_id,$row["m
edicine_category_id"]);

array_push($this-
>dosecategory_id,$row["doseca
tegory_id"]);

array_push($this-
>brand,$row["brand"]);

array_push($this-
>supplier_id,$row["supplier_i
d"]);

array_push($this-
>manufacture_date,$row["manuf
acture_date"]);

array_push($this-
>expiry_date,$row["expiry_dat
e"]);

array_push($this-
>stock,$row["stock"]);

array_push($this-
>quantity,$row["quantity"]);

array_push($this-
>status,$row["status"]);
            }
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

    protected function
findmedicine($like) {
        try {
            $this->db-
>beginTransaction();
            foreach($this-
>db->query("SELECT * FROM
medicine m
                join unit u
on u.unit_id = m.unit_id
                join
medicinecategory mc on
```

```
mc.medicine_category_id =
m.medicine_category_id
                join
dosecategory d on
d.dosecategory_id =
m.dosecategory_id
                join supplier
s on s.supplier_id =
m.supplier_id
                WHERE
medicine_id LIKE '%" . $like
. "%'
                OR
m.medicine_name LIKE '%" .
$like . "%'
                OR
m.item_description LIKE '%" .
$like . "%'
                OR  m.dosage
LIKE '%" . $like . "%'
                OR
u.unit_name LIKE '%" . $like
. "%'
                OR
mc.category_description LIKE
'%" . $like . "%'
                OR
d.dosecategory_name LIKE '%"
. $like . "%'
                OR  m.brand
LIKE '%" . $like . "%'
                OR
s.supplier_name LIKE '%" .
$like . "%'
                OR
m.manufacture_date LIKE '%" .
$like . "%'
                OR
m.expiry_date LIKE '%" .
$like . "%'
                OR  m.stock
LIKE '%" . $like . "%'
                OR
m.quantity LIKE '%" . $like .
"%'
                OR  m.status
LIKE '%" . $like . "%'
                 ") as $row)
{
                array_push($this-
>medicine_id,
$row["medicine_id"]);

                array_push($this-
>medicine_name,
$row["medicine_name"]);

                array_push($this-
>item_description,
$row["item_description"]);

                array_push($this->dosage,
$row["dosage"]);

                array_push($this->unit_id,
$row["unit_id"]);

                array_push($this-
>medicine_category_id,
$row["medicine_category_id"])
;

                array_push($this-
>dosecategory_id,
$row["dosecategory_id"]);

                array_push($this-
>brand,$row["brand"]);

                array_push($this-
>supplier_id,$row["supplier_i
d"]);

                array_push($this-
>manufacture_date,
$row["manufacture_date"]);

                array_push($this-
>expiry_date,
$row["expiry_date"]);

                array_push($this->stock,
$row["stock"]);

                array_push($this->quantity,
$row["quantity"]);
```

```
array_push($this->status,
$row["status"]);

            }
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

    protected function
updatemedicine() {
        try {
            $this->db-
>beginTransaction();
            $stmt = $this-
>db->prepare("UPDATE medicine
SET medicine_name =
:medicine_name,
item_description =
:item_description,
                dosage =
:dosage,  unit_id = :unit_id,

medicine_category_id =
:medicine_category_id,

dosecategory_id =
:dosecategory_id,
                brand =
:brand,
                supplier_id =
:supplier_id,

manufacture_date =
:manufacture_date,
                expiry_date =
:expiry_date,
                stock =
:stock,
                quantity =
:quantity,
                status =
:status
                WHERE
medicine_id = :medicine_id");

            $stmt-
>execute(array("medicine_id"
=> $this->medicine_id,
"medicine_name" => $this-
>medicine_name,
"item_description" => $this-
>item_description, "dosage"
=> $this->dosage,
                "unit_id" =>
$this->unit_id,
"medicine_category_id" =>
$this->medicine_category_id,

"dosecategory_id" => $this-
>dosecategory_id,  "brand" =>
$this->brand, "supplier_id"
=> $this->supplier_id,
"manufacture_date" => $this-
>manufacture_date,
                "expiry_date"
=> $this->expiry_date,
"stock" => $this->stock,
"quantity" => $this-
>quantity, "status" => $this-
>status));
            $affected_rows =
$stmt->rowCount();
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

    protected function
updatemedicinebypullout($id)
{
        try {
            $this->db-
>beginTransaction();
```

```
            $stmt = $this-
>db->prepare("UPDATE medicine
SET stock = :stock,

status:status,

manufacture_date:manufacture_
date,

expiry_date:expiry_date
                  WHERE
medicine_id = :medicine_id");

            $stmt-
>execute(array("stock" =>
$this->stock,"status" =>
$this-
>status,"manufacture_date" =>
$this-
>manufacture_date,"expiry_dat
e" => $this-
>expiry_date,"medicine_id" =>
$id));
            $affected_rows =
$stmt->rowCount();
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

class Scheduling {
    private $db,
$scheduling_id, $doctor_id,
$dated, $starttime, $endtime,
$date_added, $remarks,
$status;

    public function
get_scheduling_credential($typ
e) {

$scheduling_credential;
        switch($type) {
            case
"scheduling_id";

$scheduling_credential =
[$type => $this-
>scheduling_id];
            break;
            case "doctor_id";

$scheduling_credential =
[$type => $this->doctor_id];
            break;
            case "dated";
```

```
    protected function
updatemedicinebystock() {
        try {
            $this->db-
>beginTransaction();
            $stmt = $this-
>db->prepare("UPDATE medicine
SET
                stock =
:stock,  expiry_date =
:expiry_date
                WHERE
medicine_id = :medicine_id");

            $stmt-
>execute(array("medicine_id"
=> $this->medicine_id,"stock"
=> $this->stock,
"expiry_date" => $this-
>expiry_date));
            $affected_rows =
$stmt->rowCount();
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }
 }
```

```
$scheduling_credential =
[$type => $this->dated];
            break;
            case "starttime";

$scheduling_credential =
[$type => $this->starttime];
            break;
            case "endtime";

$scheduling_credential =
[$type => $this->endtime];
            break;
            case "date_added";

$scheduling_credential =
[$type => $this->date_added];
            break;
            case "remarks";

$scheduling_credential =
[$type => $this->remarks];
            break;
            case "status";

$scheduling_credential =
[$type => $this->status];
            break;
            case "all";

$scheduling_credential =
["scheduling_id" => $this-
>scheduling_id,
                "doctor_id" =>
$this->doctor_id,
                "dated" =>
$this->dated,
                "starttime" =>
$this->starttime,
                "endtime" =>
$this->endtime,
                "date_added"
=> $this->date_added,
                "remarks" =>
$this->remarks,
                "status" =>
$this->status];
            break;
            default:
$scheduling_credential = null;
        }
        return
$scheduling_credential;
    }

    public function
set_scheduling_credential($typ
e,$value) {
        switch($type) {
            case
"scheduling_id";
                $this-
>scheduling_id = $value;
            break;
            case "doctor_id";
                $this-
>doctor_id = $value;
            break;
            case "dated";
                $this->dated =
$value;
            break;
            case "starttime";
                $this-
>starttime = $value;
            break;
            case "endtime";
                $this->endtime
= $value;
            break;
            case "date_added";
                $this-
>date_added = $value;
            break;
            case "remarks";
                $this->remarks
= $value;
            break;
            case "status";
                $this->status
= $value;
            break;
            case "all";
                $this-
>scheduling_id =
$value["scheduling_id"];
```

```
                $this->doctor_id = $value["doctor_id"];
                $this->dated = $value["dated"];
                $this->starttime = $value["starttime"];
                $this->endtime = $value["endtime"];
                $this->date_added = $value["date_added"];
                $this->remarks = $value["remarks"];
                $this->status = $value["status"];
                break;
            }
        }

        public function startcreatescheduling() {
            $status = true;
            $this->createscheduling();
            return $status;
        }

        public function startreadscheduling() {
            $this->scheduling_id = [];
            $this->doctor_id = [];
            $this->dated = [];
            $this->starttime = [];
            $this->endtime = [];
            $this->date_added = [];
            $this->remarks = [];
            $this->status = [];
            $this->readscheduling();
        }

        public function startfindscheduling($like) {
            $this->scheduling_id = [];
            $this->doctor_id = [];
            $this->dated = [];
            $this->starttime = [];
            $this->endtime = [];
            $this->date_added = [];
            $this->remarks = [];
            $this->status = [];
            $this->findscheduling($like);
        }

        public function startreadschedulingbydate($from,$to) {
            $this->scheduling_id = [];
            $this->doctor_id = [];
            $this->dated = [];
            $this->starttime = [];
            $this->endtime = [];
            $this->date_added = [];
            $this->remarks = [];
            $this->status = [];
            $this->readschedulingbydate($from,$to);
            }

        public function startupdatescheduling() {
            $status = true;
            $this->updatescheduling();
            return $status;
        }

        public function __construct() {
            $this->db = new PDO("mysql:host=localhost;dbname=bhc_db;charset=utf8mb4", "root","");
            $this->db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
            $this->db->setAttribute(PDO::ATTR_EMULATE_PREPARES, false);
        }
```

```
    protected function
createscheduling() {
        try {
            $this->db-
>beginTransaction();
            $stmt = $this->db-
>prepare("INSERT INTO
scheduling ( scheduling_id,
doctor_id, dated,
                starttime,
endtime, date_added, remarks,
status) VALUES ( null,
:doctor_id, :dated,
:starttime, :endtime,
:date_added, :remarks,
:status)");
            $stmt-
>execute(array("doctor_id" =>
$this->doctor_id,
                "dated" =>
$this->dated,
                "starttime" =>
$this->starttime,
                "endtime" =>
$this->endtime,
                "date_added"
=> $this->date_added,
                "remarks" =>
$this->remarks,
                "status" =>
$this->status));
            $affected_rows =
$stmt->rowCount();
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

    protected function
readscheduling() {
        try {
            $this->db-
>beginTransaction();
            foreach($this->db-
>query("SELECT * FROM
scheduling") as $row) {

array_push($this-
>scheduling_id,
$row["scheduling_id"]);

array_push($this->doctor_id,
$row["doctor_id"]);

array_push($this->dated,
$row["dated"]);

array_push($this->starttime,
$row["starttime"]);

array_push($this->endtime,
$row["endtime"]);

array_push($this->date_added,
$row["date_added"]);

array_push($this->remarks,
$row["remarks"]);

array_push($this->status,
$row["status"]);
            }
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

    protected function
readschedulingbydate($from,$to
) {
        try {
            $this->db-
>beginTransaction();
```

```
            foreach($this->db-
>query("SELECT

s.scheduling_id, s.doctor_id,
s.dated, s.starttime,
s.endtime,
                s.date_added,
s.remarks, s.status
                FROM
scheduling s
                join doctor d
on d.doctor_id = s.doctor_id
                where
s.date_added BETWEEN '$from'
AND '$to'") as $row) {

array_push($this-
>scheduling_id,
$row["scheduling_id"]);

array_push($this->doctor_id,
$row["doctor_id"]);

array_push($this->dated,
$row["dated"]);

array_push($this->starttime,
$row["starttime"]);

array_push($this->endtime,
$row["endtime"]);

array_push($this->date_added,
$row["date_added"]);

array_push($this->remarks,
$row["remarks"]);

array_push($this->status,
$row["status"]);
            }
            $this->db-
>commit();
        }
        catch (PDOException
$ex){
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
            }
        }

    protected function
findscheduling($like) {
        try {
            $this->db-
>beginTransaction();
            foreach($this->db-
>query("SELECT * FROM
scheduling s
                join doctor d
on d.doctor_id = s.doctor_id
                WHERE
scheduling_id LIKE '%" . $like
. "%'
                OR  d.lname
LIKE '%" . $like . "%'
                OR  d.fname
LIKE '%" . $like . "%'
                OR  d.mname
LIKE '%" . $like . "%'
                OR  s.dated
LIKE '%" . $like . "%'
                OR
s.starttime LIKE '%" . $like .
"%'
                OR  s.endtime
LIKE '%" . $like . "%'
                OR
s.date_added LIKE '%" . $like
. "%'
                OR  s.remarks
LIKE '%" . $like . "%'
                OR  s.status
LIKE '%" . $like . "%' ") as
$row) {

array_push($this-
>scheduling_id,
$row["scheduling_id"]);

array_push($this->doctor_id,
$row["doctor_id"]);

array_push($this->dated,
$row["dated"]);

array_push($this->starttime,
$row["starttime"]);
```

```
array_push($this->endtime,
$row["endtime"]);

array_push($this->date_added,
$row["date_added"]);

array_push($this->remarks,
$row["remarks"]);

array_push($this->status,
$row["status"]);
            }
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

    protected function
updatescheduling() {
        try {
            $this->db-
>beginTransaction();
            $stmt = $this->db-
>prepare("UPDATE scheduling
SET
                doctor_id =
:doctor_id,
                dated =
:dated,
                starttime =
:starttime,


class FifoMedicine {
    private $fifomedicine_id,
$medicine_id, $unit_id,
$supplier_id,
$manufacture_date,
$expiry_date,
    $stock, $quantity,
$date_added, $status;
```

```
                endtime =
:endtime,
                date_added =
:date_added,
                remarks =
:remarks,
                status =
:status WHERE scheduling_id =
:scheduling_id");
            $stmt-
>execute(array("scheduling_id"
=> $this->scheduling_id,
                "doctor_id" =>
$this->doctor_id,
                "dated" =>
$this->dated,
                "starttime" =>
$this->starttime,
                "endtime" =>
$this->endtime,
                "date_added"
=> $this->date_added,
                "remarks" =>
$this->remarks,
                "status" =>
$this->status));
            $affected_rows =
$stmt->rowCount();
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }
}

    public function
get_fifomedicine_credential($t
ype) {

$fifomedicine_credential;
        switch($type) {
            case
"fifomedicine_id";
```

```
$fifomedicine_credential =
[$type => $this-
>fifomedicine_id];
            break;
            case
"medicine_id";

$fifomedicine_credential =
[$type => $this->medicine_id];
            break;
            case "unit_id";

$fifomedicine_credential =
[$type => $this->unit_id];
            break;
            case
"supplier_id";

$fifomedicine_credential =
[$type => $this->supplier_id];
            break;
            case
"manufacture_date";

$fifomedicine_credential =
[$type => $this-
>manufacture_date];
            break;
            case
"expiry_date";

$fifomedicine_credential =
[$type => $this->expiry_date];
            break;
            case "stock";

$fifomedicine_credential =
[$type => $this->stock];
            break;
            case "quantity";

$fifomedicine_credential =
[$type => $this->quantity];
            break;
            case "date_added";

$fifomedicine_credential =
[$type => $this->date_added];
            break;
            case "status";

$fifomedicine_credential =
[$type => $this->status];
            break;
            case "all";

$fifomedicine_credential =
["fifomedicine_id" => $this-
>fifomedicine_id,
"medicine_id" => $this-
>medicine_id, "unit_id" =>
$this->unit_id, "supplier_id"
=> $this->supplier_id,

"manufacture_date" => $this-
>manufacture_date,
"expiry_date" => $this-
>expiry_date,
                    "stock" =>
$this->stock, "quantity" =>
$this->quantity, "date_added"
=> $this->date_added, "status"
=> $this->status];
            break;
            default:

$fifomedicine_credential =
null;
        }
        return
$fifomedicine_credential;
    }

    public function
set_fifomedicine_credential($t
ype,$value) {
        switch($type) {
            case
"fifomedicine_id";
                $this-
>fifomedicine_id = $value;
            break;
            case
"medicine_id";
                $this-
>medicine_id = $value;
            break;
            case "unit_id";
```

```
                $this->unit_id
= $value;
            break;
            case
"supplier_id";
                $this-
>supplier_id = $value;
            break;
            case
"manufacture_date";
                $this-
>manufacture_date = $value;
            break;
            case
"expiry_date";
                $this-
>expiry_date = $value;
            break;
            case "stock";
                $this->stock =
$value;
            break;
            case "quantity";
                $this-
>quantity = $value;
            break;
            case "date_added";
                $this-
>date_added = $value;
            break;
            case "status";
                $this->status
= $value;
            break;
            case "all";
                $this-
>fifomedicine_id =
$value["fifomedicine_id"];
                $this-
>medicine_id =
$value["medicine_id"];
                $this->unit_id
= $value["unit_id"];
                $this-
>supplier_id =
$value["supplier_id"];
                $this-
>manufacture_date =
$value["manufacture_date"];
                $this-
>expiry_date =
$value["expiry_date"];
                $this->stock =
$value["stock"];
                $this-
>quantity =
$value["quantity"];
                $this-
>date_added =
$value["date_added"];
                $this->status
= $value["status"];
                break;
        }
    }

    public function
startcreatefifomedicine() {
        $status = true;
        $this-
>createfifomedicine();
        return $status;
    }

    public function
startreadfifomedicine() {
        $this->fifomedicine_id
= [];
        $this->medicine_id =
[];
        $this->unit_id = [];
        $this->supplier_id =
[];
        $this-
>manufacture_date = [];
        $this->expiry_date =
[];
        $this->stock = [];
        $this->quantity = [];
        $this->date_added =
[];
        $this->status = [];
        $this-
>readfifomedicine();
    }

      public function
startreadfifomedicinebydate($f
rom,$to) {
```

```
        $this->fifomedicine_id
= [];
        $this->medicine_id =
[];
        $this->unit_id = [];
        $this->supplier_id =
[];
        $this-
>manufacture_date = [];
        $this->expiry_date =
[];
        $this->stock = [];
        $this->quantity = [];
        $this->date_added =
[];
        $this->status = [];
        $this-
>readfifomedicinebydate($from,
$to);
        }

        protected function
readfifomedicinebydate($from,$
to) {
        try {
            $this->db-
>beginTransaction();
            foreach($this->db-
>query("SELECT

ff.fifomedicine_id,

ff.medicine_id,
                ff.unit_id,

ff.supplier_id,

ff.manufacture_date,

ff.expiry_date,
                ff.stock,
                ff.quantity,
                ff.date_added,
                ff.status
                FROM
fifomedicine ff
                join supplier
s on s.supplier_id =
ff.supplier_id
                join medicine
m on m.medicine_id =
ff.medicine_id
                join unit u on
u.unit_id = ff.unit_id
                where
ff.date_added BETWEEN '$from'
AND '$to'") as $row) {

array_push($this-
>fifomedicine_id,
$row["fifomedicine_id"]);

array_push($this->medicine_id,
$row["medicine_id"]);

array_push($this->unit_id,
$row["unit_id"]);

array_push($this->supplier_id,
$row["supplier_id"]);

array_push($this-
>manufacture_date,
$row["manufacture_date"]);

array_push($this->expiry_date,
$row["expiry_date"]);

array_push($this->stock,
$row["stock"]);

array_push($this->quantity,
$row["quantity"]);

array_push($this->date_added,
$row["date_added"]);

array_push($this->status,
$row["status"]);
            }
            $this->db-
>commit();
        }
        catch (PDOException
$ex){
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
```

```
        }
    }

    public function
startfindfifomedicine($like) {
        $this->fifomedicine_id
= [];
        $this->medicine_id =
[];
        $this->unit_id = [];
        $this->supplier_id =
[];
        $this-
>manufacture_date = [];
        $this->expiry_date =
[];
        $this->stock = [];
        $this->quantity = [];
        $this->date_added =
[];
        $this->status = [];
        $this-
>findfifomedicine($like);
    }

    public function
startupdatefifomedicine() {
        $status = true;
        $this-
>updatefifomedicine();
        return $status;
    }

    public function
startupdatefifomedicinebystock
() {
        $status = true;
        $this-
>updatefifomedicinebystock();
        return $status;
    }

    public function
__construct() {
        $this->db = new
PDO("mysql:host=localhost;dbna
me=bhc_db;charset=utf8mb4",
"root","");
        $this->db-
>setAttribute(PDO::ATTR_ERRMOD
E, PDO::ERRMODE_EXCEPTION);
        $this->db-
>setAttribute(PDO::ATTR_EMULAT
E_PREPARES, false);
    }

    protected function
createfifomedicine() {
        try {
            $this->db-
>beginTransaction();
            $stmt = $this->db-
>prepare("INSERT INTO
fifomedicine (
fifomedicine_id,  medicine_id,
unit_id, supplier_id,
manufacture_date, expiry_date,
stock, quantity, date_added,
status) VALUES ( null,
:medicine_id, :unit_id,
:supplier_id,
:manufacture_date,
:expiry_date, :stock,
:quantity, :date_added,
:status)");
            $stmt-
>execute(array(
                "medicine_id"
=> $this->medicine_id,
                "unit_id" =>
$this->unit_id,
                "supplier_id"
=> $this->supplier_id,

"manufacture_date" => $this-
>manufacture_date,
                "expiry_date"
=> $this->expiry_date,
                "stock" =>
$this->stock,
                "quantity" =>
$this->quantity,
                "date_added"
=> $this->date_added,
                "status" =>
$this->status));
            $affected_rows =
$stmt->rowCount();
```

```
            $this->db->commit();
        }
        catch (PDOException $ex) {
            $this->db->rollBack();
            echo $ex->getMessage();
        }
    }

    protected function updatefifomedicinebystock() {
        try {
            $this->db->beginTransaction();
            $stmt = $this->db->prepare("UPDATE fifomedicine SET
                stock = :stock
                WHERE fifomedicine_id = :fifomedicine_id");

            $stmt->execute(array("fifomedicine_id" => $this->fifomedicine_id,"stock" => $this->stock));
            $affected_rows = $stmt->rowCount();
            $this->db->commit();
        }
        catch (PDOException $ex) {
            $this->db->rollBack();
            echo $ex->getMessage();
        }
    }

    protected function readfifomedicine() {
        try {
            $this->db->beginTransaction();
            foreach($this->db->query("SELECT * FROM fifomedicine order by fifomedicine_id desc") as $row) {

array_push($this->fifomedicine_id, $row["fifomedicine_id"]);

array_push($this->medicine_id, $row["medicine_id"]);

array_push($this->unit_id, $row["unit_id"]);

array_push($this->supplier_id, $row["supplier_id"]);

array_push($this->manufacture_date, $row["manufacture_date"]);

array_push($this->expiry_date, $row["expiry_date"]);

array_push($this->stock, $row["stock"]);

array_push($this->quantity, $row["quantity"]);

array_push($this->date_added, $row["date_added"]);

array_push($this->status, $row["status"]);
            }
            $this->db->commit();
        }
        catch (PDOException $ex) {
            $this->db->rollBack();
            echo $ex->getMessage();
        }
    }
```

```
    protected function
findfifomedicine($like) {
        try {
            $this->db-
>beginTransaction();
            foreach($this->db-
>query("SELECT * FROM
fifomedicine f
                join medicine
m on m.medicine_id =
f.medicine_id
                join supplier
s on s.supplier_id =
f.supplier_id
                join unit u on
u.unit_id = f.unit_id
                WHERE
fifomedicine_id LIKE '%" .
$like . "%'
                OR
m.medicine_name LIKE '%" .
$like . "%'
                OR
u.unit_name LIKE '%" . $like .
"%'
                OR
s.supplier_name LIKE '%" .
$like . "%'
                OR
f.manufacture_date LIKE '%" .
$like . "%'
                OR
f.expiry_date LIKE '%" . $like
. "%'
                OR  f.stock
LIKE '%" . $like . "%'
                OR  f.quantity
LIKE '%" . $like . "%'
                OR
f.date_added LIKE '%" . $like
. "%'
                OR  f.status
LIKE '%" . $like . "%'") as
$row) {

array_push($this-
>fifomedicine_id,
$row["fifomedicine_id"]);

array_push($this->medicine_id,
$row["medicine_id"]);

array_push($this->unit_id,
$row["unit_id"]);

array_push($this->supplier_id,
$row["supplier_id"]);

array_push($this-
>manufacture_date,
$row["manufacture_date"]);

array_push($this->expiry_date,
$row["expiry_date"]);

array_push($this->stock,
$row["stock"]);

array_push($this->quantity,
$row["quantity"]);

array_push($this->date_added,
$row["date_added"]);

array_push($this->status,
$row["status"]);
            }
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

    protected function
updatefifomedicine() {
        try {
            $this->db-
>beginTransaction();
            $stmt = $this->db-
>prepare("UPDATE fifomedicine
SET
```

```
                medicine_id =
:medicine_id,
                unit_id =
:unit_id,
                supplier_id =
:supplier_id,

manufacture_date =
:manufacture_date,
                expiry_date =
:expiry_date,
                stock =
:stock,
                quantity =
:quantity,
                date_added =
:date_added,
                status =
:status WHERE fifomedicine_id
= :fifomedicine_id");
            $stmt-
>execute(array("fifomedicine_i
d" => $this->fifomedicine_id,
                "medicine_id"
=> $this->medicine_id,
                "unit_id" =>
$this->unit_id,
                "supplier_id"
=> $this->supplier_id,
"manufacture_date" => $this-
>manufacture_date,
                "expiry_date"
=> $this->expiry_date,
                "stock" =>
$this->stock,
                "quantity" =>
$this->quantity,
                "date_added"
=> $this->date_added,
                "status" =>
$this->status));
                $affected_rows =
$stmt->rowCount();
                $this->db-
>commit();
            }
            catch (PDOException
$ex) {
                $this->db-
>rollBack();
                echo $ex-
>getMessage();
            }
        }
}
```

```
class ReceiveOrder {
    private  $db,
$receiveorder_id,
$medicine_id, $dosage,
$unit_id, $dosecategory_id,
$brand,
    $supplier_id,
$manufacture_date,
$expiry_date,
$date_added,$stock;

    public function
get_receiveorder_credential($t
ype) {

$receiveorder_credential;
        switch($type) {
            case
"receiveorder_id";
$receiveorder_credential =
[$type => $this-
>receiveorder_id];
            break;
            case
"medicine_id";

$receiveorder_credential =
[$type => $this->medicine_id];
            break;
            case "dosage";

$receiveorder_credential =
[$type => $this->dosage];
            break;
            case "unit_id";

$receiveorder_credential =
[$type => $this->unit_id];
```

```
            break;

            case
"dosecategory_id";

$receiveorder_credential =
[$type => $this-
>dosecategory_id];
            break;
            case "brand";

$receiveorder_credential =
[$type => $this->brand];
            break;
            case
"supplier_id";

$receiveorder_credential =
[$type => $this->supplier_id];
            break;
            case
"manufacture_date";

$receiveorder_credential =
[$type => $this-
>manufacture_date];
            break;
            case
"expiry_date";

$receiveorder_credential =
[$type => $this->expiry_date];
            break;
            case "date_added";

$receiveorder_credential =
[$type => $this->date_added];
            break;
            case "stock";

$receiveorder_credential =
[$type => $this->stock];
            break;

            case "all";

$receiveorder_credential =
["receiveorder_id" => $this-
>receiveorder_id,
                    "dosage" =>
$this->dosage,
                    "medicine_id"
=> $this->medicine_id,
                    "unit_id" =>
$this->unit_id,


"dosecategory_id" => $this-
>dosecategory_id,
                    "brand" =>
$this->brand,
                    "supplier_id"
=> $this->supplier_id,

"manufacture_date" => $this-
>manufacture_date,
                    "expiry_date"
=> $this->expiry_date,
                    "date_added"
=> $this->date_added,
                    "stock" =>
$this->stock
                        ];
            break;
            default:

$receiveorder_credential =
null;
        }
        return
$receiveorder_credential;
    }

    public function
set_receiveorder_credential($t
ype,$value) {
        switch($type) {
            case
"receiveorder_id";
                $this-
>receiveorder_id = $value;
            break;
            case
"medicine_id";
                $this-
>medicine_id = $value;
            break;
            case "dosage";
```

```
                $this->dosage
= $value;
            break;
            case "unit_id";
                $this->unit_id
= $value;
            break;

            case
"dosecategory_id";
            $this-
>dosecategory_id = $value;
            break;
            case "brand";
                $this->brand =
$value;
            break;
            case
"supplier_id";
                $this-
>supplier_id = $value;
            break;
            case
"manufacture_date";
                $this-
>manufacture_date = $value;
            break;
            case
"expiry_date";
                $this-
>expiry_date = $value;
            break;
            case "date_added";
                $this-
>date_added = $value;
            break;
             case "stock";
                $this->stock =
$value;
            break;

            case "all";
                $this-
>receiveorder_id =
$value["receiveorder_id"];
                $this-
>medicine_id =
$value["medicine_id"];
                $this->dosage
= $value["dosage"];
                $this->unit_id
= $value["unit_id"];
                $this-
>dosecategory_id =
$value["dosecategory_id"];
                $this->brand =
$value["brand"];
                $this-
>supplier_id =
$value["supplier_id"];
                $this-
>manufacture_date =
$value["manufacture_date"];
                $this-
>expiry_date =
$value["expiry_date"];
                $this-
>date_added =
$value["date_added"];
                $this->stock =
$value["stock"];

                break;
            }
        }

        public function
startcreatereceiveorder() {
            $date_added = true;
            $this-
>createreceiveorder();
            return $date_added;
        }

        public function
startreadreceiveoder() {
            $this->receiveorder_id
= [];
            $this->medicine_id =
[];
            $this->dosage = [];
            $this->unit_id = [];
            $this->dosecategory_id
= [];
            $this->brand = [];
            $this->supplier_id =
[];
            $this-
>manufacture_date = [];
```

```
        $this->expiry_date =
[];
        $this->date_added =
[];
        $this->stock = [];
        $this-
>readreceiveorder();
    }

    public function
startfindreceiveorder($like) {
        $this->receiveorder_id
= [];
        $this->medicine_id =
[];
        $this->dosage = [];
        $this->unit_id = [];
        $this->dosecategory_id
= [];
        $this->brand = [];
        $this->supplier_id =
[];
        $this-
>manufacture_date = [];
        $this->expiry_date =
[];
        $this->date_added =
[];
        $this->stock = [];
        $this-
>findreceiveorder($like);
    }

    public function
startreadreceiveroder1($from,$
to) {
        $this->receiveorder_id
= [];
        $this->medicine_id =
[];
        $this->dosage = [];
        $this->unit_id = [];
        $this->dosecategory_id
= [];
        $this->brand = [];
        $this->supplier_id =
[];
        $this-
>manufacture_date = [];
        $this->expiry_date =
[];
        $this->date_added =
[];
        $this->stock = [];
        $this-
>findreceiveorderbyid($from,$t
o);
     }

    public function
startupdatereceiveorder() {
        $date_added = true;
       $this-
>updatereceiveorder();
        return $date_added;
    }

    public function
startreadreceiveorderbydate1($
from,$to) {
        $this->receiveorder_id
= [];
        $this->medicine_id =
[];
        $this->dosage = [];
        $this->unit_id = [];
        $this->dosecategory_id
= [];
        $this->brand = [];
        $this->supplier_id =
[];
        $this-
>manufacture_date = [];
        $this->expiry_date =
[];
        $this->date_added =
[];
        $this->stock = [];
        $this-
>readreceiveorderbydate2($from
,$to);
    }

    public function
startreadreceiveoderbydate($fr
om,$to) {
        $this->receiveorder_id
= [];
```

```
        $this->medicine_id =
[];
        $this->dosage = [];
        $this->unit_id = [];
        $this->dosecategory_id
= [];
        $this->brand = [];
        $this->supplier_id =
[];
        $this-
>manufacture_date = [];
        $this->expiry_date =
[];
        $this->date_added =
[];
        $this->stock = [];
        $this-
>readreceiveorderbydate($from,
$to);
    }

    public function
__construct() {
        $this->db = new
PDO("mysql:host=localhost;dbna
me=bhc_db;charset=utf8mb4",
"root","");
        $this->db-
>setAttribute(PDO::ATTR_ERRMOD
E, PDO::ERRMODE_EXCEPTION);
        $this->db-
>setAttribute(PDO::ATTR_EMULAT
E_PREPARES, false);
    }

    protected function
readreceiveorderbydate($from,$
to) {
        try {
            $this->db-
>beginTransaction();
            foreach($this->db-
>query("SELECT

ro.receiveorder_id,

ro.medicine_id,
                ro.dosage,
                ro.unit_id,
ro.dosecategory_id,
                ro.brand,

ro.supplier_id,

ro.manufacture_date,

ro.expiry_date,
                ro.date_added,
                ro.stock

                FROM
receiveorder ro join medicine
m on m.dosage = ro.dosage
where ro.supplier_id BETWEEN
'$from' AND '$to' ORDER BY
m.medicine_name AND
ro.supplier_id") as $row) {

array_push($this-
>receiveorder_id,$row["receive
order_id"]);

array_push($this-
>medicine_id,$row["medicine_id
"]);

array_push($this-
>dosage,$row["dosage"]);

array_push($this-
>unit_id,$row["unit_id"]);

array_push($this-
>dosecategory_id,$row["dosecat
egory_id"]);

array_push($this-
>brand,$row["brand"]);

array_push($this-
>supplier_id,$row["supplier_id
"]);

array_push($this-
>manufacture_date,$row["manufa
cture_date"]);

array_push($this-
```

```
>expiry_date,$row["expiry_date
"]);

array_push($this-
>date_added,$row["date_added"]
);

array_push($this-
>stock,$row["stock"]);

            }
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

    protected function
readreceiveorderbydate1($from,
$to) {
        try {
            $this->db-
>beginTransaction();
            foreach($this->db-
>query("SELECT

rco.receiveorder_id,
rco.medicine_id, rco.dosage,
rco.unit_id,
rco.dosecategory_id,
                rco.brand,
rco.supplier_id,
rco.manufacture_date,
rco.expiry_date,
rco.date_added,
                rco.stock
                FROM
receiveorder rco
                join supplier
s on s.supplier_id =
rco.supplier_id
                join medicine
m on m.medicine_id =
rco.medicine_id
                join
dosecategory dc on
dc.dosecategory_id =
rco.dosecategory_id
                join unit u on
u.unit_id = rco.unit_id
                where
rco.date_added BETWEEN '$from'
AND '$to'") as $row) {

array_push($this-
>receiveorder_id,
$row["receiveorder_id"]);

array_push($this->medicine_id,
$row["medicine_id"]);

array_push($this->supplier_id,
$row["supplier_id"]);

array_push($this->dosage,
$row["dosage"]);

array_push($this->unit_id,
$row["unit_id"]);

array_push($this-
>dosecategory_id,
$row["dosecategory_id"]);

array_push($this->brand,
$row["brand"]);

array_push($this->supplier_id,
$row["supplier_id"]);

array_push($this-
>manufacture_date,
$row["manufacture_date"]);

array_push($this->expiry_date,
$row["expiry_date"]);

array_push($this->date_added,
$row["date_added"]);

array_push($this->stock,
$row["stock"]);
            }
```

```
            $this->db-
>commit();
        }
        catch (PDOException
$ex){
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

    protected function
readreceiveorderbydate2($from,
$to) {
        try {
            $this->db-
>beginTransaction();
            foreach($this->db-
>query("SELECT

rco.receiveorder_id,
rco.medicine_id, rco.dosage,
rco.unit_id,
rco.dosecategory_id,
                rco.brand,
rco.supplier_id,
rco.manufacture_date,
rco.expiry_date,
rco.date_added,
                rco.stock
                FROM
receiveorder rco
                join supplier
s on s.supplier_id =
rco.supplier_id
                join medicine
m on m.medicine_id =
rco.medicine_id
                join
dosecategory dc on
dc.dosecategory_id =
rco.dosecategory_id
                join unit u on
u.unit_id = rco.unit_id
                where
rco.date_added BETWEEN '$from'
AND '$to'") as $row) {

array_push($this-
>receiveorder_id,
$row["receiveorder_id"]);

array_push($this->medicine_id,
$row["medicine_id"]);

array_push($this->supplier_id,
$row["supplier_id"]);

array_push($this->dosage,
$row["dosage"]);

array_push($this->unit_id,
$row["unit_id"]);

array_push($this-
>dosecategory_id,
$row["dosecategory_id"]);

array_push($this->brand,
$row["brand"]);

array_push($this->supplier_id,
$row["supplier_id"]);

array_push($this-
>manufacture_date,
$row["manufacture_date"]);

array_push($this->expiry_date,
$row["expiry_date"]);

array_push($this->date_added,
$row["date_added"]);

array_push($this->stock,
$row["stock"]);
            }
            $this->db-
>commit();
        }
        catch (PDOException
$ex){
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }
```

```
    protected function
createreceiveorder() {
        try {
            $this->db-
>beginTransaction();
            $stmt = $this->db-
>prepare("INSERT INTO
receiveorder
(receiveorder_id,medicine_id,
dosage, unit_id,
dosecategory_id, brand,
supplier_id, manufacture_date,
expiry_date, date_added,
stock)
            VALUES ( null,
:medicine_id, :dosage,
:unit_id,:dosecategory_id,
:brand, :supplier_id,
:manufacture_date,
:expiry_date, :date_added,
:stock)");

            $stmt-
>execute(array(
                "medicine_id"
=> $this->medicine_id,
                "dosage" =>
$this->dosage,
                "unit_id" =>
$this->unit_id,

"dosecategory_id" => $this-
>dosecategory_id,
                "brand" =>
$this->brand,
                "supplier_id"
=> $this->supplier_id,

"manufacture_date" => $this-
>manufacture_date,
                "expiry_date"
=> $this->expiry_date,
                "date_added"
=> $this->date_added,
                "stock" =>
$this->stock));
            $affected_rows =
$stmt->rowCount();
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

    protected function
readreceiveorder() {
        try {
            $this->db-
>beginTransaction();
            foreach($this->db-
>query("SELECT * FROM
receiveorder order by
receiveorder_id desc") as
$row) {

array_push($this-
>receiveorder_id,$row["receive
order_id"]);

array_push($this-
>medicine_id,$row["medicine_id
"]);

array_push($this-
>dosage,$row["dosage"]);

array_push($this-
>unit_id,$row["unit_id"]);

array_push($this-
>dosage,$row["dosecategory_id"
]);

array_push($this-
>brand,$row["brand"]);

array_push($this-
>supplier_id,$row["supplier_id
"]);

array_push($this-
>manufacture_date,$row["manufa
cture_date"]);
```

```
array_push($this->expiry_date,$row["expiry_date"]);

array_push($this->date_added,$row["date_added"]);

array_push($this->stock,$row["stock"]);

            }
            $this->db->commit();
        }
        catch (PDOException $ex) {
            $this->db->rollBack();
            echo $ex->getMessage();
        }
    }

    protected function findreceiveorder($like) {
        try {
            $this->db->beginTransaction();
            foreach($this->db->query("SELECT * FROM receiveorder r
                join medicine m on m.dosage=r.dosage
                join supplier s on s.dosage = r.dosage
                join dosecategory d on r.dosecategory_id = r.dosecategory_id
                join medicine u on u.medicine_id = r.medicine_id
                WHERE m.medicine_name LIKE '%" . $like . "%'
                OR s.supplier_name LIKE '%" . $like . "%'
                OR r.medicine_id LIKE '%" . $like . "%'
                OR  r.dosage LIKE '%" . $like . "%'
                OR d.dosecategory_name LIKE '%" . $like . "%'
                OR u.unit_name LIKE '%" . $like . "%'
                OR  r.brand LIKE '%" . $like . "%'
                OR r.supplier_id LIKE '%" . $like . "%'
                OR r.manufacture_date LIKE '%" . $like . "%'
                OR r.expiry_date LIKE '%" . $like . "%'
                OR r.date_added LIKE '%" . $like . "%'
                OR  r.stock LIKE '%" . $like . "%'

                 ") as $row) {


array_push($this->receiveorder_id, $row["receiveorder_id"]);

array_push($this->medicine_id, $row["medicine_id"]);

array_push($this->dosage, $row["dosage"]);

array_push($this->unit_id, $row["unit_id"]);

array_push($this->manufacture_date, $row["dosecategory_id"]);

array_push($this->brand, $row["brand"]);
```

```
array_push($this->supplier_id,
$row["supplier_id"]);

array_push($this-
>manufacture_date,
$row["manufacture_date"]);

array_push($this->expiry_date,
$row["expiry_date"]);

array_push($this->date_added,
$row["date_added"]);

array_push($this->stock,
$row["stock"]);

            }
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

    protected function
findreceiveorderbyid($from,$to
) {
        try {
            $this->db-
>beginTransaction();
            foreach($this->db-
>query("SELECT * FROM
receiveorder where date_added
BETWEEN '$from' AND '$to'") as
$row) {


array_push($this-
>receiveorder_id,
$row["receiveorder_id"]);

array_push($this->medicine_id,
$row["medicine_id"]);

array_push($this->dosage,
$row["dosage"]);

array_push($this->unit_id,
$row["unit_id"]);

array_push($this-
>manufacture_date,
$row["dosecategory_id"]);

array_push($this->brand,
$row["brand"]);

array_push($this->supplier_id,
$row["supplier_id"]);

array_push($this-
>manufacture_date,
$row["manufacture_date"]);

array_push($this->expiry_date,
$row["expiry_date"]);

array_push($this->date_added,
$row["date_added"]);

array_push($this->stock,
$row["stock"]);

            }
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

    protected function
updatereceiveorder() {
        try {
            $this->db-
>beginTransaction();
            $stmt = $this->db-
>prepare("UPDATE receiveorder
```

```
SET dosage = :dosage,  dosage
= :dosage,
                medicine_id =
:medicine_id,
                dosage:dosage,
                  unit_id =
:unit_id,

dosecategory_id:dosecategory_i
d,
                brand =
:brand,
                supplier_id =
:supplier_id,

manufacture_date =
:manufacture_date,
                expiry_date =
:expiry_date,
                date_added =
:date_added,
                stock =
:stock,

                WHERE
receiveorder_id =
:receiveorder_id");

            $stmt-
>execute(array("receiveorder_i
d" => $this->receiveorder_id,
                "medicine_id"
=> $this->medicine_id,
                "dosage" =>
$this->dosage,
                "unit_id" =>
$this->unit_id,

"dosecategory_id" => $this-
>dosecategory_id,
                "brand" =>
$this->brand,
                "supplier_id"
=> $this->supplier_id,

"manufacture_date" => $this-
>manufacture_date,
                "expiry_date"
=> $this->expiry_date,
                "date_added"
=> $this->date_added,
                "stock" =>
$this->stock));
                $affected_rows =
$stmt->rowCount();
                $this->db-
>commit();
            }
            catch (PDOException
$ex) {
                $this->db-
>rollBack();
                echo $ex-
>getMessage();
            }
        }
    }
```

```
  class RequestOrder {
    private  $db,
$requestorder_id,
$supplier_id, $medicine_id,
$dosecategory_id, $unit_id,
    $stock, $dated_req,
$dated_app, $date_added,
$status, $approved,$receive,

$requestorder_genid,$receive_s
tock;

    public function
get_requestorder_credential($t
ype) {

$requestorder_credential;
        switch($type) {
            case
"requestorder_id";

$requestorder_credential =
[$type => $this-
>requestorder_id];
            break;
```

```
            case
"supplier_id";

$requestorder_credential =
[$type => $this->supplier_id];
            break;
            case
"medicine_id";

$requestorder_credential =
[$type => $this->medicine_id];
            break;
            case
"dosecategory_id";

$requestorder_credential =
[$type => $this-
>dosecategory_id];
            break;
            case "unit_id";

$requestorder_credential =
[$type => $this->unit_id];
            break;
            case "stock";

$requestorder_credential =
[$type => $this->stock];
            break;
            case "dated_req";

$requestorder_credential =
[$type => $this->dated_req];
            break;
            case "dated_app";

$requestorder_credential =
[$type => $this->dated_app];
            break;
            case "date_added";

$requestorder_credential =
[$type => $this->date_added];
            break;
            case "status";

$requestorder_credential =
[$type => $this->status];
            break;
            case "approved";

$requestorder_credential =
[$type => $this->approved];
            break;
            case "receive";

$requestorder_credential =
[$type => $this->receive];
            break;
            case
"requestorder_genid";

$requestorder_credential =
[$type => $this-
>requestorder_genid];
            break;
             case
"receive_stock";

$requestorder_credential =
[$type => $this-
>receive_stock];
            break;
            case "all";

$requestorder_credential =
["requestorder_id" => $this-
>requestorder_id,
                    "supplier_id"
=> $this->supplier_id,
                    "medicine_id"
=> $this->medicine_id,

"dosecategory_id" => $this-
>dosecategory_id,
                    "unit_id" =>
$this->unit_id,
                    "stock" =>
$this->stock,
                    "dated_req"
=> $this->dated_req,
                    "dated_app"
=> $this->dated_app,
                    "date_added"
=> $this->date_added,
                    "status" =>
$this->status,
                    "approved" =>
$this->approved,
```

```
                "receive" =>
$this->receive,

"requestorder_genid" => $this-
>requestorder_genid,

"receive_stock" => $this-
>receive_stock
                    ];
            break;
            default:

$requestorder_credential =
null;
        }
        return
$requestorder_credential;
    }

    public function
set_requestorder_credential($t
ype,$value) {
        switch($type) {
            case
"requestorder_id";
                $this-
>requestorder_id = $value;
            break;
            case
"supplier_id";
                $this-
>supplier_id = $value;
            break;
            case
"medicine_id";
                $this-
>medicine_id = $value;
            break;
            case
"dosecategory_id";
                $this-
>dosecategory_id = $value;
            break;
            case "unit_id";
                $this->unit_id
= $value;
            break;
            case "stock";
                $this->stock =
$value;
            break;
            case "dated_req";
                $this-
>dated_req = $value;
            break;
            case "dated_app";
                $this-
>dated_app = $value;
            break;
            case "date_added";
                $this-
>date_added = $value;
            break;
            case "status";
                $this->status
= $value;
            break;
             case "approved";
                $this-
>approved = $value;
            break;
            case "receive";
                $this->receive
= $value;
            break;
            case
"requestorder_genid";
                $this-
>requestorder_genid = $value;
            break;
            case
"receive_stock";
                $this-
>receive_stock = $value;
            break;
            case "all";
                $this-
>requestorder_id =
$value["requestorder_id"];
                $this-
>supplier_id =
$value["supplier_id"];
                $this-
>medicine_id =
$value["medicine_id"];
                $this-
>dosecategory_id =
$value["dosecategory_id"];
                $this->unit_id
= $value["unit_id"];
```

```
                $this->stock =
$value["stock"];
                $this-
>dated_req =
$value["dated_req"];
                $this-
>dated_app =
$value["dated_app"];
                $this-
>date_added =
$value["date_added"];
                $this->status
= $value["status"];
                $this-
>approved =
$value["approved"];
                $this->receive
= $value["receive"];
                $this-
>requestorder_genid =
$value["requestorder_genid"];
                $this-
>receive_stock =
$value["receive_stock"];
                break;
        }
    }

    public function
startcreaterequestorder() {
        $status = true;
     $this-
>createrequestorder();
        return $status;
    }

    public function
startreadrequestorder() {
        $this->requestorder_id
= [];
        $this->supplier_id =
[];
        $this->medicine_id =
[];
        $this->dosecategory_id
= [];
        $this->unit_id = [];
        $this->stock = [];
        $this->dated_req = [];
        $this->dated_app = [];
        $this->date_added =
[];
        $this->status = [];
        $this->approved = [];
        $this->receive = [];
        $this-
>requestorder_genid = [];
        $this->receive_stock =
[];
        $this-
>readrequestorder();
    }

    public function
startfindrequestorder($like) {
        $this->requestorder_id
= [];
        $this->supplier_id =
[];
        $this->medicine_id =
[];
        $this->dosecategory_id
= [];
        $this->unit_id = [];
        $this->stock = [];
        $this->dated_req = [];
        $this->dated_app = [];
        $this->date_added =
[];
        $this->status = [];
        $this->approved = [];
        $this->receive = [];
        $this-
>requestorder_genid = [];
        $this->receive_stock =
[];
        $this-
>findrequestorder($like);
    }

    public function
startreadrequestorderbydate($f
rom,$to) {
        $this->requestorder_id
= [];
        $this->supplier_id =
[];
        $this->medicine_id =
[];
```

```
        $this->dosecategory_id
= [];
        $this->unit_id = [];
        $this->stock = [];
        $this->dated_req = [];
        $this->dated_app = [];
        $this->date_added =
[];
        $this->status = [];
        $this->approved = [];
        $this->receive = [];
        $this-
>requestorder_genid = [];
        $this->receive_stock =
[];
        $this-
>readrequestorderbydate($from,
$to);
        }

    public function
startupdaterequestorder() {
        $status = true;
        $this-
>updaterequestorder();
        return $status;
    }

    public function
__construct() {
        $this->db = new
PDO("mysql:host=localhost;dbna
me=bhc_db;charset=utf8mb4",
"root","");
        $this->db-
>setAttribute(PDO::ATTR_ERRMOD
E, PDO::ERRMODE_EXCEPTION);
        $this->db-
>setAttribute(PDO::ATTR_EMULAT
E_PREPARES, false);
    }

    protected function
createrequestorder() {
        try {
            $this->db-
>beginTransaction();
            $stmt = $this->db-
>prepare("INSERT INTO
requestorder (
requestorder_id,  supplier_id,
medicine_id, dosecategory_id,
unit_id, stock, dated_req,
dated_app,  date_added,
status,
approved,receive,requestorder_
genid, receive_stock) VALUES (
null,  :supplier_id,
:medicine_id,
:dosecategory_id, :unit_id,
:stock, :dated_req,
:dated_app, :date_added,
:status, :approved, :receive,
:requestorder_genid,
:receive_stock)");

            $stmt-
>execute(array(
                "supplier_id"
=> $this->supplier_id,
                "medicine_id"
=> $this->medicine_id,

"dosecategory_id" => $this-
>dosecategory_id,
                "unit_id" =>
$this->unit_id,
                "stock" =>
$this->stock,
                "dated_req" =>
$this->dated_req,
                "dated_app" =>
$this->dated_app,
                "date_added"
=> $this->date_added,
                "status" =>
$this->status,
                "approved" =>
$this->approved,
                "receive" =>
$this->receive,

"requestorder_genid" => $this-
>requestorder_genid,

"receive_stock" => $this-
>receive_stock));
            $affected_rows =
$stmt->rowCount();
```

```
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

    protected function
readrequestorder() {
        try {
            $this->db-
>beginTransaction();
            foreach($this->db-
>query("SELECT * FROM
requestorder ORDER BY
requestorder_id DESC") as
$row) {

array_push($this-
>requestorder_id,$row["request
order_id"]);

array_push($this-
>supplier_id,$row["supplier_id
"]);

array_push($this-
>medicine_id,$row["medicine_id
"]);

array_push($this-
>dosecategory_id,$row["dosecat
egory_id"]);

array_push($this-
>unit_id,$row["unit_id"]);

array_push($this-
>stock,$row["stock"]);

array_push($this-
>dated_req,$row["dated_req"]);

array_push($this-
>dated_app,$row["dated_app"]);

array_push($this-
>date_added,$row["date_added"]
);

array_push($this-
>status,$row["status"]);

array_push($this-
>approved,$row["approved"]);

array_push($this-
>receive,$row["receive"]);

array_push($this-
>requestorder_genid,$row["requ
estorder_genid"]);

array_push($this-
>receive_stock,$row["receive_s
tock"]);
            }
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

    protected function
readrequestorderbydate($from,$
to) {
        try {
            $this->db-
>beginTransaction();
            foreach($this->db-
>query("SELECT

ro.requestorder_id,ro.supplier
_id, ro.medicine_id,
ro.dosecategory_id,
ro.unit_id,
                ro.stock,
ro.dated_req, ro.dated_app,
ro.date_added, ro.status,
```

```
                ro.approved,
ro.receive,
ro.requestorder_genid,
ro.receive_stock
                FROM
requestorder ro
                join supplier
s on s.supplier_id =
ro.supplier_id
                join medicine
m on m.medicine_id =
ro.medicine_id
                join
dosecategory dc on
dc.dosecategory_id =
ro.dosecategory_id
                join unit u on
u.unit_id = ro.unit_id
                where
ro.date_added BETWEEN '$from'
AND '$to'") as $row) {

array_push($this-
>requestorder_id,
$row["requestorder_id"]);

array_push($this->supplier_id,
$row["supplier_id"]);

array_push($this->medicine_id,
$row["medicine_id"]);

array_push($this-
>dosecategory_id,
$row["dosecategory_id"]);

array_push($this->unit_id,
$row["unit_id"]);

array_push($this->stock,
$row["stock"]);

array_push($this-
>dated_req,$row["dated_req"]);

array_push($this->dated_app,
$row["dated_app"]);

array_push($this->date_added,
$row["date_added"]);

array_push($this->status,
$row["status"]);

array_push($this->approved,
$row["approved"]);

array_push($this->receive,
$row["receive"]);

array_push($this-
>requestorder_genid,
$row["requestorder_genid"]);

array_push($this-
>receive_stock,
$row["receive_stock"]);
            }
            $this->db-
>commit();
        }
        catch (PDOException
$ex){
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

    public function
startreadreceivemedicinebydate
($from,$to) {
        $this->requestorder_id
= [];
        $this->supplier_id =
[];
        $this->medicine_id =
[];
        $this->dosecategory_id
= [];
        $this->unit_id = [];
        $this->stock = [];
        $this->dated_req = [];
        $this->dated_app = [];
        $this->date_added =
[];
        $this->status = [];
        $this->approved = [];
        $this->receive = [];
```

```
        $this->receive_stock =
[];
        $this-
>readreceivemedicinebydate($fr
om,$to);
        }

     protected function
readreceivemedicinebydate($fro
m,$to) {
        try {
            $this->db-
>beginTransaction();
            foreach($this->db-
>query("SELECT

ro.requestorder_id,ro.supplier
_id, ro.medicine_id,
ro.dosecategory_id,
ro.unit_id,
                ro.stock,
ro.dated_req, ro.dated_app,
ro.date_added, ro.status,
                ro.approved,
ro.receive, ro.receive_stock
                FROM
requestorder ro
                join supplier
s on s.supplier_id =
ro.supplier_id
                join medicine
m on m.medicine_id =
ro.medicine_id
                join
dosecategory dc on
dc.dosecategory_id =
ro.dosecategory_id
                join unit u on
u.unit_id = ro.unit_id
                where
ro.date_added BETWEEN '$from'
AND '$to'") as $row) {

array_push($this-
>requestorder_id,
$row["requestorder_id"]);

array_push($this->supplier_id,
$row["supplier_id"]);

array_push($this->medicine_id,
$row["medicine_id"]);

array_push($this-
>dosecategory_id,
$row["dosecategory_id"]);

array_push($this->unit_id,
$row["unit_id"]);

array_push($this->stock,
$row["stock"]);

array_push($this-
>dated_req,$row["dated_req"]);

array_push($this->dated_app,
$row["dated_app"]);

array_push($this->date_added,
$row["date_added"]);

array_push($this->status,
$row["status"]);

array_push($this->approved,
$row["approved"]);

array_push($this->receive,
$row["receive"]);

array_push($this-
>receive_stock,
$row["receive_stock"]);
            }
            $this->db-
>commit();
        }
        catch (PDOException
$ex){
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

    protected function
findrequestorder($like) {
```

```
        try {
            $this->db->beginTransaction();
            foreach($this->db->query("SELECT
r.requestorder_id,
             r.supplier_id,
             r.medicine_id,
r.dosecategory_id,
             r.unit_id,
             r.stock,
             r.dated_req,
             r.dated_app,
             r.date_added,
             r.status,
             r.approved,
             r.receive,
r.requestorder_genid,
             r.receive_stock
             FROM requestorder r
                join medicine m on
m.medicine_id=r.medicine_id
                join supplier s on s.supplier_id =
r.supplier_id
                join dosecategory d on
d.dosecategory_id =
r.dosecategory_id
                join unit u on
u.unit_id = r.unit_id
                WHERE
m.medicine_name LIKE '%" . $like . "%'
                OR
s.supplier_name LIKE '%" . $like . "%'
                OR
m.medicine_name LIKE '%" . $like . "%'
                OR
d.dosecategory_name LIKE '%" . $like . "%'
                OR
u.unit_name LIKE '%" . $like . "%'
                OR  r.stock
LIKE '%" . $like . "%'
                OR
r.dated_req LIKE '%" . $like . "%'
                OR
r.dated_app LIKE '%" . $like . "%'
                OR
r.date_added LIKE '%" . $like . "%'
                OR  r.status
LIKE '%" . $like . "%'
                OR  r.approved
LIKE '%" . $like . "%'
                OR
r.requestorder_genid LIKE '%" . $like . "%'
                OR
r.receive_stock LIKE '%" . $like . "%'
                 ") as $row) {

array_push($this->requestorder_id,
$row["requestorder_id"]);

array_push($this->supplier_id,
$row["supplier_id"]);

array_push($this->medicine_id,
$row["medicine_id"]);

array_push($this->dosecategory_id,
$row["dosecategory_id"]);

array_push($this->unit_id,
$row["unit_id"]);

array_push($this->stock,
$row["stock"]);

array_push($this->dated_req,
$row["dated_req"]);
```

```
array_push($this->dated_app,
$row["dated_app"]);

array_push($this->date_added,
$row["date_added"]);

array_push($this->status,
$row["status"]);

array_push($this->approved,
$row["approved"]);

array_push($this->receive,
$row["receive"]);

array_push($this-
>requestorder_genid,
$row["requestorder_genid"]);

array_push($this-
>receive_stock,
$row["receive_stock"]);
            }
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

    protected function
updaterequestorder() {
        try {
            $this->db-
>beginTransaction();
            $stmt = $this->db-
>prepare("UPDATE requestorder
SET supplier_id =
:supplier_id,  medicine_id =
:medicine_id,

dosecategory_id =
:dosecategory_id,
                unit_id =
:unit_id,
                stock =
:stock,
                dated_req =
:dated_req,
                dated_app =
:dated_app,
                date_added =
:date_added,
                status =
:status,
                approved =
:approved,
                receive =
:receive,
                receive_stock
= :receive_stock
                WHERE
requestorder_id =
:requestorder_id");

            $stmt-
>execute(array("requestorder_i
d" => $this->requestorder_id,
                "supplier_id"
=> $this->supplier_id,
                "medicine_id"
=> $this->medicine_id,

"dosecategory_id" => $this-
>dosecategory_id,
                "unit_id" =>
$this->unit_id,
                "stock" =>
$this->stock,
                "dated_req" =>
$this->dated_req,
                "dated_app" =>
$this->dated_app,
                "date_added"
=> $this->date_added,
                "status" =>
$this->status,
                "approved" =>
$this->approved,
                "receive" =>
$this->receive,
```

```
"receive_stock" => $this-
>receive_stock));
            $affected_rows =
$stmt->rowCount();
            $this->db-
>commit();
        }


class Record{
    private $db, $record_id,
$patient_id, $doctor_id,
$introfinding, $prescription,
$date_added, $status,
$medicine_id, $quantity;

  public function
get_record_credential($type) {
        $record_credential;
        switch($type) {
              case "record_id";

  $record_credential = [$type
=> $this->record_id];
              break;
              case
"patient_id";

  $record_credential = [$type
=> $this->patient_id];
              break;
              case "doctor_id";

  $record_credential = [$type
=> $this->doctor_id];
              break;
              case
"introfinding";

  $record_credential = [$type
=> $this->introfinding];
              break;
              case
"prescription";

  $record_credential = [$type
=> $this->prescription];
              break;
            catch (PDOException
$ex) {
                $this->db-
>rollBack();
                echo $ex-
>getMessage();
            }
        }
    }

              case
"date_added";

  $record_credential = [$type
=> $this->date_added];
              break;
              case "status";

  $record_credential = [$type
=> $this->status];
              break;
              case
"medicine_id";

  $record_credential = [$type
=> $this->medicine_id];
              break;
              case "quantity";

  $record_credential = [$type
=> $this->quantity];
              break;
              case "all";

  $record_credential =
["record_id" => $this-
>record_id, "patient_id" =>
$this->patient_id, "doctor_id"
=> $this->doctor_id,
"introfinding" => $this-
>introfinding, "prescription"
=> $this->prescription,
"date_added" => $this-
>date_added, "status" =>
$this->status,"medicine_id" =>
$this->medicine_id,"quantity"
=> $this->quantity];
              break;
              default:
```

```
  $record_credential = null;
        }
        return
$record_credential;
  }

  public function
set_record_credential($type,$v
alue) {
        switch($type) {
              case "record_id";
                    $this-
>record_id = $value;
              break;
              case
"patient_id";
                    $this-
>patient_id = $value;
              break;
              case "doctor_id";
                    $this-
>doctor_id = $value;
              break;
              case
"introfinding";
                    $this-
>introfinding = $value;
              break;
              case
"prescription";
                    $this-
>prescription = $value;
              break;
              case
"date_added";
                    $this-
>date_added = $value;
              break;
              case "status";
                    $this-
>status = $value;
              break;
              case
"medicine_id";
                    $this-
>medicine_id = $value;
              break;
              case "quantity";
                    $this-
>quantity = $value;
              break;
              case "all";
                    $this-
>record_id =
$value["record_id"];
                    $this-
>patient_id =
$value["patient_id"];
                    $this-
>doctor_id =
$value["doctor_id"];
                    $this-
>introfinding =
$value["introfinding"];
                    $this-
>prescription =
$value["prescription"];
                    $this-
>date_added =
$value["date_added"];
                    $this-
>status = $value["status"];
                    $this-
>medicine_id =
$value["medicine_id"];
                    $this-
>quantity =
$value["quantity"];
              break;
        }
  }

  public function
startcreaterecord() {
        $status = true;
        if ($this->patient_id
== null) $status = false;
        else if ($this-
>doctor_id == null) $status =
false;
        else if ($this-
>introfinding == null) $status
= false;
        else if ($this-
>prescription == null) $status
= false;
```

```
        else if ($this-
>date_added == null) $status =
false;
        else if ($this->status
== null) $status = false;
        else if ($this-
>medicine_id == null) $status
= false;
        else if ($this-
>quantity == null) $status =
false;
        else $this-
>createrecord();
        return $status;
  }

  public function
startreadrecord() {
        $this->record_id = [];
        $this->patient_id = [];
        $this->doctor_id = [];
        $this->introfinding =
[];
        $this->prescription =
[];
        $this->date_added = [];
        $this->status = [];
        $this->medicine_id =
[];
        $this->quantity = [];
        $this->readrecord();
  }

    public function
startreadrecordbydate($from,$t
o) {
        $this->record_id = [];
        $this->patient_id =
[];
        $this->doctor_id = [];
        $this->introfinding =
[];
        $this->prescription =
[];
        $this->date_added =
[];
        $this->status = [];
        $this->medicine_id =
[];
        $this->quantity = [];
        $this-
>readrecordbydate($from,$to);
    }

    public function
startreadpatientrecordbydate($
from,$to) {
        $this->record_id = [];
        $this->patient_id =
[];
        $this->doctor_id = [];
        $this->introfinding =
[];
        $this->prescription =
[];
        $this->date_added =
[];
        $this->status = [];
        $this->medicine_id =
[];
        $this->quantity = [];
        $this-
>readpatientrecordbydate($from
,$to);
    }

  public function
startfindrecord($like) {
        $this->record_id = [];
        $this->patient_id = [];
        $this->doctor_id = [];
        $this->introfinding =
[];
        $this->prescription =
[];
        $this->date_added = [];
        $this->status = [];
        $this->medicine_id =
[];
        $this->quantity = [];
        $this-
>findrecord($like);
  }

    public function
startfindrecordbyacc($like) {
        $this->record_id = [];
        $this->patient_id =
[];
        $this->doctor_id = [];
```

```
        $this->introfinding =
[];
        $this->prescription =
[];
        $this->date_added =
[];
        $this->status = [];
        $this->medicine_id =
[];
        $this->quantity = [];
        $this-
>findrecordbyacc($like);
    }

    public function
startreadrecordbydateconsult($
from,$to) {
        $this->record_id = [];
        $this->patient_id =
[];
        $this->doctor_id = [];
        $this->introfinding =
[];
        $this->prescription =
[];
        $this->date_added =
[];
        $this->status = [];
        $this->medicine_id =
[];
        $this->quantity = [];
        $this-
>readrecordbydateconsult($from
,$to);
    }

  public function
startfindrecordbyname($like) {
        $this->record_id = [];
        $this->patient_id = [];
        $this->doctor_id = [];
        $this->introfinding =
[];
        $this->prescription =
[];
        $this->date_added = [];
        $this->status = [];
        $this->medicine_id =
[];
        $this->quantity = [];
        $this-
>findrecordbyname($like);
  }

  public function
startupdaterecord() {
        $status = true;
        if ($this->record_id ==
null) $status = false;
        else if ($this-
>patient_id == null) $status =
false;
        else if ($this-
>doctor_id == null) $status =
false;
        else if ($this-
>introfinding == null) $status
= false;
        else if ($this-
>prescription == null) $status
= false;
        else if ($this-
>date_added == null) $status =
false;
        else if ($this->status
== null) $status = false;
        else if ($this-
>medicine_id == null) $status
= false;
        else if ($this-
>quantity == null) $status =
false;
        else $this-
>updaterecord();
        $this->updaterecord();
        return $status;
    }

    public function
startupdaterecordbymedrabies()
{
        $status = true;
        if ($this->record_id ==
null) $status = false;
        else $this-
>updaterecordbymedrabies();
     $this-
>updaterecordbymedrabies();
        return $status;
    }
```

```
  public function __construct()
{
        $this->db = new
PDO("mysql:host=localhost;dbna
me=bhc_db;charset=utf8mb4",
"root","");
        $this->db-
>setAttribute(PDO::ATTR_ERRMOD
E, PDO::ERRMODE_EXCEPTION);
        $this->db-
>setAttribute(PDO::ATTR_EMULAT
E_PREPARES, false);
  }

    protected function
findrecordbyacc($like) {
        try {
            $this->db-
>beginTransaction();
            foreach($this->db-
>query("SELECT * FROM record r
                join patient p
on p.patient_id = r.patient_id
                join doctor dt
on dt.doctor_id = r.doctor_id
                join medicine
m on m.medicine_id =
r.medicine_id
                WHERE
record_id LIKE '%".
                $like . "%'
OR  p.lname LIKE '%" .
                $like . "%'
OR  p.fname LIKE '%" .
                $like . "%'
OR  p.mname LIKE '%" .
                $like . "%'
OR  dt.lname LIKE '%" .
                $like . "%'
OR  dt.fname LIKE '%" .
                $like . "%'
OR  dt.mname LIKE '%" .
                $like . "%'
OR  r.introfinding LIKE '%" .
                $like . "%'
OR  r.prescription LIKE '%" .
                $like . "%'
OR  r.date_added LIKE '%" .
                $like . "%'
OR  r.status LIKE '%" .
                $like . "%'
OR  m.medicine_name LIKE '%" .
                $like . "%'
OR  r.quantity LIKE '%" .$like
. "%'") as $row) {

array_push($this->record_id,
$row["record_id"]);

array_push($this->patient_id,
$row["patient_id"]);

array_push($this->doctor_id,
$row["doctor_id"]);

array_push($this-
>introfinding,
$row["introfinding"]);

array_push($this-
>prescription,
$row["prescription"]);

array_push($this->date_added,
$row["date_added"]);

array_push($this->status,
$row["status"]);

array_push($this->medicine_id,
$row["medicine_id"]);

array_push($this->quantity,
$row["quantity"]);
            }
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }
```

```
    protected function
readrecordbydateconsult($from,
$to) {
        try {
            $this->db-
>beginTransaction();
            foreach($this->db-
>query("SELECT
                r.record_id,
                r.patient_id,
                r.doctor_id,

r.introfinding,

r.prescription,
                r.date_added,
                r.status,
                r.medicine_id,
                r.quantity
                FROM record r
                join patient p
on p.patient_id =
r.patient_id,
                join medicine
mdc on mdc.medicine_id =
r.medicine_id,
                join doctor d
on d.doctor_id =  r.doctor_id
                where
r.date_added BETWEEN '$from'
AND '$to' ORDER BY d.doctor_id
AND r.date_added") as $row) {

array_push($this->record_id,
$row["record_id"]);

array_push($this->patient_id,
$row["patient_id"]);

array_push($this->doctor_id,
$row["doctor_id"]);

array_push($this-
>introfinding,
$row["introfinding"]);

array_push($this-
>prescription,
$row["prescription"]);

array_push($this->date_added,
$row["date_added"]);

array_push($this->status,
$row["status"]);

array_push($this->medicine_id,
$row["medicine_id"]);

array_push($this->quantity,
$row["quantity"]);
            }
            $this->db-
>commit();
        }
        catch (PDOException
$ex){
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

    protected function
findrecordbyname($like) {
        try {
            $this->db-
>beginTransaction();
            foreach($this-
>db->query("SELECT DISTINCT *
FROM record INNER JOIN patient
on record.patient_id =
patient.patient_id WHERE
patient.fname LIKE '%" . $like
. "%' OR patient.mname LIKE
'%" . $like . "%' OR
patient.lname LIKE '%" . $like
. "%' "."order by
record.date_added") as $row) {

 array_push($this->record_id,
$row["record_id"]);

 array_push($this->patient_id,
$row["patient_id"]);

 array_push($this->doctor_id,
$row["doctor_id"]);
```

```
  array_push($this-
>introfinding,
$row["introfinding"]);

  array_push($this-
>prescription,
$row["prescription"]);

  array_push($this->date_added,
$row["date_added"]);

  array_push($this->status,
$row["status"]);

  array_push($this-
>medicine_id,
$row["medicine_id"]);

  array_push($this->quantity,
$row["quantity"]);
              }
              $this->db-
>commit();
        }
        catch (PDOException
$ex) {
              $this->db-
>rollBack();
              echo $ex-
>getMessage();
        }
  }

    protected function
readpatientrecordbydate($from,
$to) {
         try {
             $this->db-
>beginTransaction();
             foreach($this->db-
>query("SELECT
                  record_id,
                  patient_id,
                  doctor_id,
                  introfinding,
                  prescription,
                  date_added,
                  status,
                  medicine_id,
                  quantity FROM
record
                    where
date_added BETWEEN '$from' AND
'$to'") as $row) {

array_push($this->record_id,
$row["record_id"]);

array_push($this->patient_id,
$row["patient_id"]);

array_push($this->doctor_id,
$row["doctor_id"]);

array_push($this-
>introfinding,
$row["introfinding"]);

array_push($this-
>prescription,
$row["prescription"]);

array_push($this->date_added,
$row["date_added"]);

array_push($this->status,
$row["status"]);

array_push($this->medicine_id,
$row["medicine_id"]);

array_push($this->quantity,
$row["quantity"]);
             }
             $this->db-
>commit();
         }
         catch (PDOException
$ex){
             $this->db-
>rollBack();
             echo $ex-
>getMessage();
         }
     }

    protected function
readrecordbydate($from,$to) {
         try {
```

```
            $this->db-
>beginTransaction();
            foreach($this->db-
>query("SELECT
                record_id,
                patient_id,
                doctor_id,
                introfinding,
                prescription,
                date_added,
                status,
                medicine_id,
                quantity FROM
record where date_added
BETWEEN '$from' AND '$to'") as
$row) {

array_push($this->record_id,
$row["record_id"]);

array_push($this->patient_id,
$row["patient_id"]);

array_push($this->doctor_id,
$row["doctor_id"]);

array_push($this-
>introfinding,
$row["introfinding"]);

array_push($this-
>prescription,
$row["prescription"]);

array_push($this->date_added,
$row["date_added"]);

array_push($this->status,
$row["status"]);

array_push($this->medicine_id,
$row["medicine_id"]);

array_push($this->quantity,
$row["quantity"]);
            }
            $this->db-
>commit();
        }
        catch (PDOException
$ex){
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
    }

  protected function
createrecord() {
        try {
            $this->db-
>beginTransaction();
            $stmt = $this-
>db->prepare("INSERT INTO
record ( record_id,
patient_id,  doctor_id,
introfinding,  prescription,
date_added,  status,
medicine_id, quantity) VALUES
( null,  :patient_id,
:doctor_id,  :introfinding,
:prescription,  :date_added,
:status, :medicine_id,
:quantity)");
            $stmt-
>execute(array("patient_id" =>
$this->patient_id, "doctor_id"
=> $this->doctor_id,
"introfinding" => $this-
>introfinding, "prescription"
=> $this->prescription,
"date_added" => $this-
>date_added, "status" =>
$this->status,

"medicine_id" => $this-
>medicine_id,  "quantity" =>
$this->quantity));
            $affected_rows =
$stmt->rowCount();
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
```

```
            echo $ex-
>getMessage();
        }
  }

  protected function
readrecord() {
        try {
            $this->db-
>beginTransaction();
            foreach($this-
>db->query("SELECT * FROM
record order by record_id
asc") as $row) {

  array_push($this->record_id,
$row["record_id"]);

  array_push($this->patient_id,
$row["patient_id"]);

  array_push($this->doctor_id,
$row["doctor_id"]);

  array_push($this-
>introfinding,
$row["introfinding"]);

  array_push($this-
>prescription,
$row["prescription"]);

  array_push($this->date_added,
$row["date_added"]);

  array_push($this->status,
$row["status"]);

  array_push($this-
>medicine_id,
$row["medicine_id"]);

  array_push($this->quantity,
$row["quantity"]);
            }
            $this->db-
>commit();
        }
        catch (PDOException
$ex) {
            $this->db-
>rollBack();
            echo $ex-
>getMessage();
        }
  }

  protected function
findrecord($like) {
        try {
            $this->db-
>beginTransaction();
            foreach($this-
>db->query("SELECT * FROM
record r
                join patient p
on p.patient_id = r.patient_id
                join doctor dt
on dt.doctor_id = r.doctor_id
                join medicine
m on m.medicine_id =
r.medicine_id
                WHERE
r.record_id LIKE '%".
                $like . "%'
OR  p.lname LIKE '%" .
                $like . "%'
OR  p.fname LIKE '%" .
                $like . "%'
OR  p.mname LIKE '%" .
                $like . "%'
OR  dt.lname LIKE '%" .
                $like . "%'
OR  dt.fname LIKE '%" .
                $like . "%'
OR  dt.mname LIKE '%" ) as
$row) {

  array_push($this->record_id,
$row["record_id"]);

  array_push($this->patient_id,
$row["patient_id"]);

  array_push($this->doctor_id,
$row["doctor_id"]);

  array_push($this-
>introfinding,
$row["introfinding"]);
```

```
  array_push($this-
>prescription,
$row["prescription"]);

  array_push($this->date_added,
$row["date_added"]);

  array_push($this->status,
$row["status"]);

  array_push($this-
>medicine_id,
$row["medicine_id"]);

  array_push($this->quantity,
$row["quantity"]);
                }
                $this->db-
>commit();
        }
        catch (PDOException
$ex) {
                $this->db-
>rollBack();
                echo $ex-
>getMessage();
        }
  }

  protected function
updaterecord() {
        try {
                $this->db-
>beginTransaction();
                $stmt = $this-
>db->prepare("UPDATE record
SET patient_id = :patient_id,
doctor_id = :doctor_id,
introfinding = :introfinding,
prescription = :prescription,
date_added = :date_added,
status = :status, medicine_id
= :medicine_id, quantity =
:quantity WHERE record_id =
:record_id");
                $stmt-
>execute(array("record_id" =>
$this->record_id, "patient_id"
=> $this->patient_id,
"doctor_id" => $this-
>doctor_id, "introfinding" =>
$this->introfinding,
"prescription" => $this-
>prescription, "date_added" =>
$this->date_added, "status" =>
$this->status, "medicine_id"
=> $this->medicine_id,
"quantity" =>$this-
>quantity));
                $affected_rows =
$stmt->rowCount();
                $this->db-
>commit();
        }
        catch (PDOException
$ex) {
                $this->db-
>rollBack();
                echo $ex-
>getMessage();
        }
    }

    protected function
updaterecordbymedrabies() {
         try {
                $this->db-
>beginTransaction();
                $stmt = $this->db-
>prepare("UPDATE record SET
medicine_id = :medicine_id
WHERE record_id =
:record_id");
                $stmt-
>execute(array("record_id" =>
$this->record_id,
"medicine_id" => $this-
>medicine_id));
                $affected_rows =
$stmt->rowCount();
                $this->db-
>commit();
        }
        catch (PDOException
$ex) {
                $this->db-
>rollBack();
                echo $ex-
>getMessage();
        }
```

Appendix D – Curriculum Vitae

ALEGRE, RYAN JOSEPH L.
4th Year College
B 27 L 53 Birmingham Village
Brgy. Pulo City of Cabuyao, Laguna
rjalegreciara@gmail.com
09198973425

PERSONAL DATA

Age:	22
Gender:	Male
Date of Birth:	May 22, 1997
Place of Birth:	Sta. Cruz Manila
Father's name:	Alegre, Romy S.
Mother's name:	Alegre, Eugene L.

EDUCATIONAL ATTAINMENT

Tertiary

Pamantasan ng Cabuyao
Bachelor of Science in Information Technology
Major in Web Development
(2013 – present)

Secondary

Pulo National High School

Brgy. Pulo City of Cabuyao Laguna

2009 – 2013

Primary

San Pedro Central Elementary School

San Pedro, Laguna

2003 – 2009

SKILLS

Software:

Microsoft Office Applications (word, excel, access, publisher, power point), Adobe Photoshop, Visual Studio

Programming:
C#, JAVA, MySQL, HTML5, JAVASCRIPT, PHP, CSS

SEMINAR AND TRAININGS ATTENDED

Y4IT (17TH Youth Congress on Information Technology)
September 25-26, 2019
University of the Philippines Diliman, Quezon City

AWS Siklab Pilipinas 2019 Cloud Computing Bootcamp
September 26, 2019
University of the Philippines Diliman, Quezon City

BASALLOTE, JOHN JORDAN M.

4th Year College

557 Purok 6 Brgy. Marinig

City of Cabuyao, Laguna

jordanbasallote30@gmail.com

09551486442

PERSONAL DATA

Age:	20
Gender:	Male
Date of Birth:	June 24, 1999
Place of Birth:	Brgy Gulod Cabuyao Laguna
Father's name:	Basallote, Moises A.
Mother's name:	Basallote, Rowena M.

EDUCATIONAL ATTAINMENT

Tertiary

Pamantasan ng Cabuyao

Bachelor of Science in Information Technology

Major in Web Development

(2015 – present)

Secondary

Gulod National High School

Brgy Gulod Cabuyao, Laguna

2014 – 2015

Primary

Marinig South Elementary School

Brgy Marinig Cabuyao, Laguna

2010 – 2011

SKILLS

Software:

Microsoft Office Applications (word, excel, access, publisher, power point), Adobe Photoshop, Visual Studio

Programming:
C#, JAVA, MySQL, HTML5, JAVASCRIPT, PHP, CSS

SEMINAR AND TRAININGS ATTENDED

Y4IT (17TH Youth Congress on Information Technology)
September 25-26, 2019
University of the Philippines Diliman, Quezon City

AWS Siklab Pilipinas 2019 Cloud Computing Bootcamp
September 26, 2019
University of the Philippines Diliman, Quezon City

VALDUEZA, APRIL LOU C.
4th Year College
Bagong Silang, Sala Cabuyao, Laguna
aprilouvaldueza@gmail.com
0923-698-5310

PERSONAL DATA

Age:	20
Gender:	Female
Date of Birth:	April 14, 1999
Place of Birth:	Sta. Cruz Provincial Hospital
Father's name:	Valdueza, Larry C.
Mother's name:	Valdueza, Liza C.

EDUCATIONAL ATTAINMENT

Tertiary

Pamantasan ng Cabuyao
Bachelor of Science in Information Technology
Major in Web Development
(2015 – present)

Secondary

Cabuyao National High School

Poblacion Tres Cabuyao, Laguna

2014 – 2015

Primary

Cabuyao Elementary School

Poblacion dos Cabuyao, Laguna

2010 – 2011

SKILLS

Software:

Microsoft Office Applications (word, excel, access, publisher, power point), Adobe Photoshop, Visual Studio

Programming:
C#, JAVA, MySQL, HTML5, JAVASCRIPT, PHP, CSS

SEMINAR AND TRAININGS ATTENDED

AWS Siklab Pilipinas 2019 Cloud Computing Bootcamp
September 26, 2019
University of the Philippines Diliman, Quezon City

Y4IT (17TH Youth Congress on Information Technology)
September 25-26, 2019
University of the Philippines Diliman, Quezon City