Pharmacy Management Software For New Popular Hospital.

Prepared by, B.M. Ashik Mahmud

A practicum report submitted in partial fulfillment of the requirements for the degree of Bachelor of Computer Science and Engineering (BCSE)

The practicum has been examined and approved,

Prof Dr Md Abdul Haque

Chair and Professor

Dept. of Computer Science and Engineering

IUBAT – International University of Business Agriculture and Technology

----

Prof Dr Utpal Kanti Das

Coordinator and Professor

Dept. of Computer Science and Engineering

IUBAT – International University of Business Agriculture and Technology

----

Rubayea Ferdows

Dept. of Computer Science and Engineering

IUBAT – International University of Business Agriculture and Technology Department of Computer Science and Engineering

College of Engineering and Technology

IUBAT – International University of Business Agriculture and Technology

Spring 2019

Abstract

This report is for practicum defense. The primary objective of this report is to learn how to conduct a project and work in real field and write it down in a formal and specific way. The secondary objective of this report is to learn about how this Pharmacy Management Software can manage inventory and sales process for owner more effective. The Pharmacy Management Software is developed for storing purchase information, sales information and loss/ profit balance data. Where admin can manage everything of the system and staff can sale only medicine. Different login system for both. Admin have to add medicine name, medicine presentation , generic name and supplier name then they can insert purchase information by date with auto calculation also retrieve the data date to date. Also sales system do the same and print the invoice of the sale. At the end of the day or anytime owner can generate the purchase and sales balance report and print the daily report. We wish to make it flawless in near future.

LETTER OF TRANSMITAL

18th April 2019

The chairman,
Practicum and Placement committee

College of Engineering and Technology (CEAT)

IUBAT– International University of Business Agriculture and Technology
4 Embankment Drive Road, Sector 10, Uttara Model Town
Dhaka 1230, Bangladesh

Subject: Letter of Transmittal.

Dear Sir,

I am pleased to present to you my practicum report titled "**Pharmacy Management Software For New Popular Hospital**" as required by International University of Business Agriculture and Technology for the partial fulfillment of the requirements for the award of Bachelor of Computer Science and Engineering. It was indeed a great opportunity for me to work on this project to actualize my theoretical knowledge into practice. Now I am looking forward for your kind appraisal of my report.

Finally, I would like to thank you for giving me the opportunity to pursue my studies in your renowned university.

Sincerely yours,

………………………
B.M Ashik Mahmud
ID: 15103052
Program: BCSE

STUDENTS'S DECLARATION

I, hereby, declare that this practicum report titled, "**Pharmacy Management Software For New Popular Hospital**" is my original work. It has never been presented previously or concurrently for any other purpose, reward or degree at IUBAT University or any other institutions either by me or by any other student. I also declare that there is no plagiarism or data falsification and materials used in this report from various sources have duly cited.

……………………………..

B.M. Ashik Mahmud

ID No. 15103052

Program: BCSE

Acknowledgements

In the name of ALLAH, who is the most merciful and the most graceful.

It‘s my pleasure to take this occasion to thank a few people, who have, assisted, encouraged, directed and supported me throughout my practicum program.

First of all, I want to thank my parents, who have endowed their immeasurable-innumerable support and encouragement to attain this exquisite event of my life.

My sincere thanks to our Vice Chancellor **Dr. Abdur Rab** to give me an opportunity to submit this report.

My outmost and sincere gratitude goes to **Prof. Dr. Utpal Kanti Das**, Coordinator of Department of Computer Science and Engineering, IUBAT– International University of Business Agriculture and Technology for allowing me to work on the project.

I would like to pay my gratitude to my faculty advisor **Rubayea Ferdows**, Faculty of Computer Science & Engineering Department, who has given me the opportunity to make such a report for not only in this semester but also throughout my education life at IUBAT– International University of Business Agriculture and Technology by giving his valuable suggestions and advices at any time, at any situation. I would able to make this report effectively and properly only for his right direction.

Last but not least, I sincerely would like to thank **Yousuf Mahmud** (CEO, Green Software and Technology) for giving me the opportunity to complete my internship and project at Green Software and Technology.

Their continuous encouragement and contribution gave me the courage and determination needed to complete the internship and project properly.

DEDICATION

This humble effort, the fruit of my thoughts and study is dedicated to the people who have always been there to encourage and support me and especially to my beloved parents whose affection, love, and prays of day and night make me able to get this project done.

I would also like to dedicate this to my friends, who have inspired me throughout my university life.

SUPERVISOR'S CLARIFICATION

I certify that the student B.M. Ashik Mahmud (ID No. 15103052) carried out her practicum work "**Pharmacy Management Software For New Popular Hospital"** between January 01, 2019 and April 18, 2019. During this period she consulted me on regular basis as required by the department. I therefore recommend that her practicum report accepted for oral examination.

____

Rubayea Ferdows
Lecturer
Dept. of Computer Science and Engineering
IUBAT – International University of Business Agriculture and Technology

DEPARTMENTAL CERTIFICATION

On behalf of the Department of Computer Science and Engineering of International University of Business Agriculture and Technology (IUBAT) we, the undersigned, certify that this practicum report "**Pharmacy Management Software For New Popular Hospital**" for the award of Bachelor of Computer Science and Engineering (BCSE) degree was duly presented by B.M. Ashik Mahmud (ID No. 15103052) and accepted by the department.

Rubayea Ferdows

Supervisor

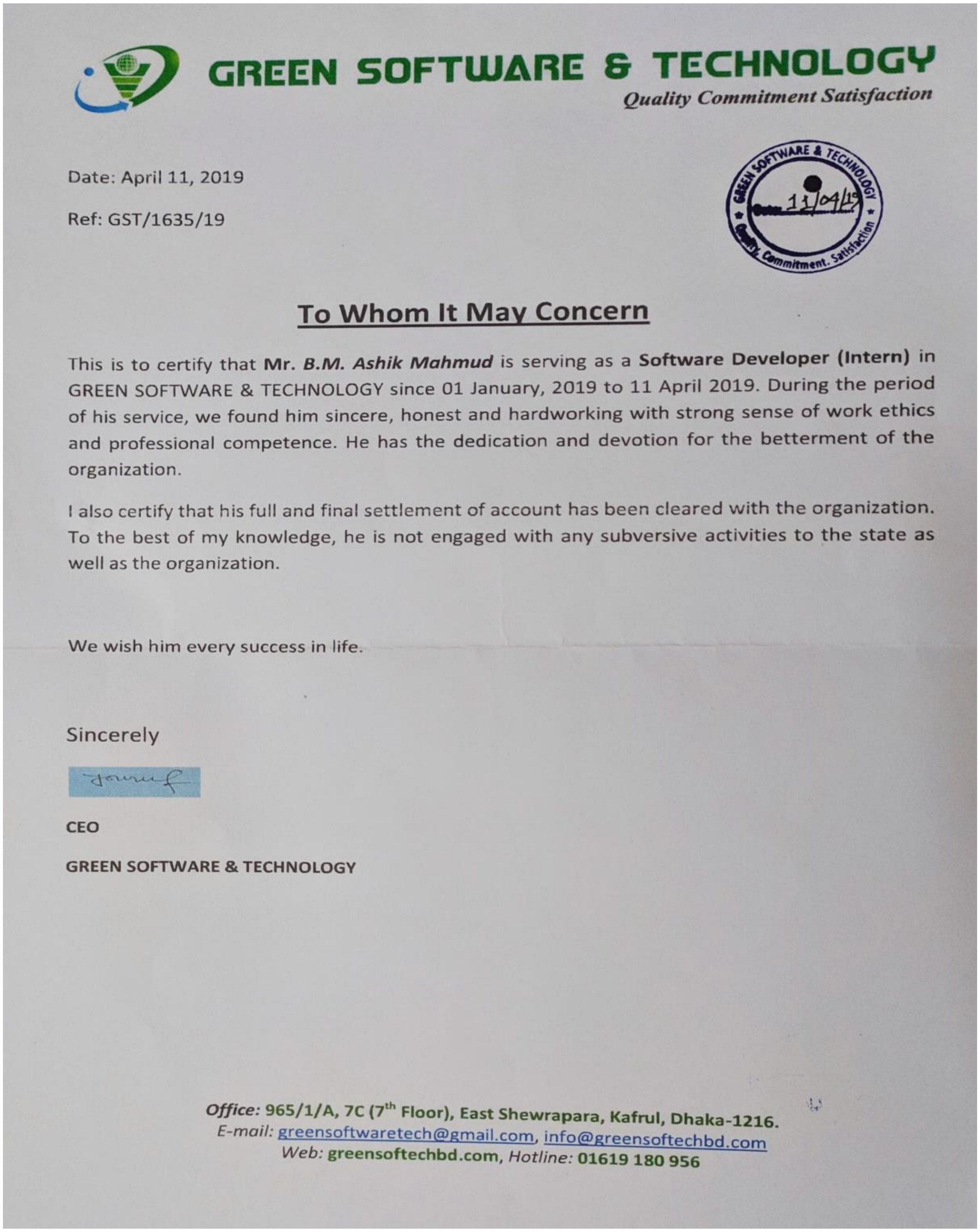
Prof Dr Utpal Kanti Das

Coordinator and Professor

Prof Dr Md Abdul Haque

Chairperson and Professor

Internship Certification



GREEN SOFTWARE & TECHNOLOGY
Quality Commitment Satisfaction

Date: April 11, 2019

Ref: GST/1635/19

GREEN SOFTWARE & TECHNOLOGY
Quality. Commitment. Satisfaction
11/04/19

To Whom It May Concern

This is to certify that **Mr. *B.M. Ashik Mahmud*** is serving as a **Software Developer (Intern)** in GREEN SOFTWARE & TECHNOLOGY since 01 January, 2019 to 11 April 2019. During the period of his service, we found him sincere, honest and hardworking with strong sense of work ethics and professional competence. He has the dedication and devotion for the betterment of the organization.

I also certify that his full and final settlement of account has been cleared with the organization. To the best of my knowledge, he is not engaged with any subversive activities to the state as well as the organization.

We wish him every success in life.

Sincerely

CEO

GREEN SOFTWARE & TECHNOLOGY

Office: 965/1/A, 7C (7th Floor), East Shewrapara, Kafrul, Dhaka-1216.
E-mail: greensoftwaretech@gmail.com, info@greensoftechbd.com
Web: greensoftechbd.com, *Hotline:* 01619 180 956

Contents

List of Figures

List of Tables

Chapter 1 | Organization

The organization chapter will describe the organizational overview, and the mission, vision of the company where I have completed my internship. It will also show the organizational hierarchy and my position there as an intern.

1.1 Organization Overview

GREEN SOFTWARE & TECHNOLOGY is a leading solution provider for internet and desktop based applications, established in 2014. The company has been promoted by some highly experienced, professional and dedicated team to provide total IT solutions under one roof. It possesses not only the latest technology but also the most knowledgeable and secured hands to offer most user friendly customized solutions. Within these years of its operations, GREEN SOFTWARE & TECHNOLOGY has carved a niche for itself in the IT industry and has increased its business by acquiring some major international & domestic projects. No doubt the company has been able to make a name for itself in a relatively short span of time because of its ability and commitments. We are ensuring customer satisfaction by rewarding quality work on right time and in a right manner.

1.2 The Mission

"Quality Commitment Satisfaction" is our slogan. Our commitment to our valued customer has multiplied our responsibilities. On the basis of that commitment, we are always trying to make their lives comfortable through our specialized services. We are working tirelessly to introduce more new innovations in purpose of giving them high quality solutions that add value and reliable competitive advantage for our client. We strongly believe that customer's satisfaction will help us to acquire our target.

1.3 The Vision

GREEN SOFTWARE & TECHNOLOGY envisions to grow as a major recognized software company providing cutting-edge technology with complete web & desktop based software solution in both local and global marketplace.

We want to build an environment where employees can unleash their full potential to leave a mark on the tech industry. Thankfully, we have a great and talented crew of professionals working here with complete sincerity and commitment to achieve our goal.

1.4 My position in this Organization

I am an Software Developer Intern of this organization. I am guided by a supervisor in this organization. He is very helpful and informative. I have learned so many new things from him. I have successfully completed my project in time. It was only possible under the guidance of my supervisor. It was also a great experience to maintain the office time for me. I have also maintained the other rules and regulation of this organization. I am really happy to work with this office. It‘s really a great opportunity for me doing great in my future career.

1.5 Organizational Structure:

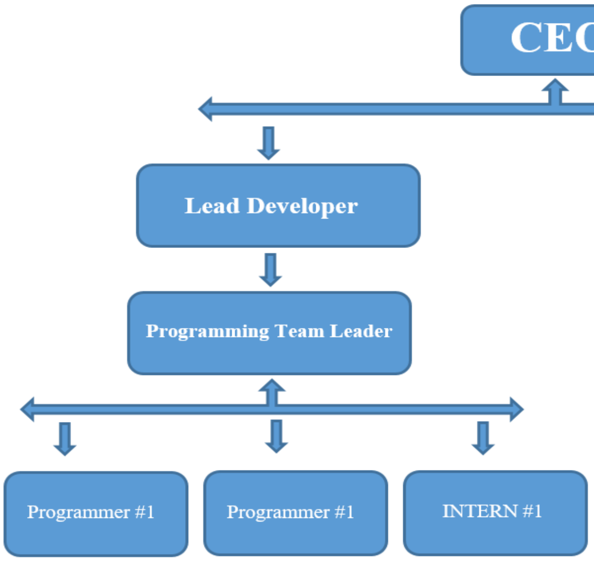


Figure 1.1: *Organizational Structure of GREEN SOFTWARE & TECHNOLOGY*

Chapter 2 | Introduction

2.1 Introduction

Internship is a practical experience of theoretically gained knowledge and can measure as a groundwork trial to be aware with any organization and to make oneself confident enough to enter into service life and start building career. And also wonderful and effective way to connect academic experience with the professional work arena. It allows gaining valuable experience to the workplace, provides the opportunity for skill development, and gives a competitive edge in the job search. This chapter attempts to describe the objectives, scope and all topics of initialization period of this project.

2.2 Project Overview

The "**Pharmacy Management Software for New Popular Hospital**" is a system that will record product purchase rate, sales rate and loss-profit calculation. It will show stock product, sales report and purchase and sales difference, so that owner can maintain loss and profit. Reports can be export as PDF, can be printable and also excel format. Existing manual system is very hard to generate and calculate all the parts. This system will help to reduce all the existing problems.

Using the system user can insert all stock product, can sell product with discount and print system and at the end user can check their balance status. Developing a system that is error free. Developing a system that will save time then manual system.

2.3 Objective

Currently the pharmacy handle Inventory, Sales and Account balance process manually which is a very time consuming process. Dealing with different supplier with purchase products and Sales medicine for many customer in same time, so it becomes a very tedious job for the staff and the owner to look after all these particulars to maintain the task at right time with accurate balance. **The Pharmacy Management Software** not only deals with Inventory but also deals with the Sales medicine process. The software will store all purchased medicine date to date also there purchase price, selling price and expire date. In Sales system the Software will auto calculate total price by multiplying unit price and quantity and can be added as many medicine as the sales man want, also can give discount on total sales and at the end can generate invoice and can print the invoice. In Account system owner can check date to date purchase and sales amount and balance. Owner and Staff login system will be separate to make the system more secure.

The objective of this system is to make the Inventory and sales system easy, reliable, user friendly, and corrective. Moreover less time consuming as compared to manual work. The system also generate purchase report date to date, sales and account report with print system. The main objective of this system is to automate inventory, sales and account process like- purchasing medicine, selling them and find the balance of loss-profit and so on.

2.4 System Benefits

The system benefit of this project has been described below-

2.4.1 Easy Access into the Software

New Populate hospital is a new hospital with modern equipment and technological system. Pharmacy Staffs send busy time to maintain there inventory and selling process. They need faster and accurate process to maintain their daily job. This system made their work more easy and faster than manual system and all data will be accurate.

2.4.2 Faster Transaction

The sales staff send very busy time to sale the medicine and they need a automatic system to know the medicine name, generic and selling price of their sales time. Also, need to calculate all medicine prices and there invoice for the client. The system do all this things an easy and faster way.

2.4.3 Accurate Record keeping

The data that admin or Staff insert into the system will be accurate by the system. Numeric and text data will check by system and also important input fields can’t be empty.

2.4.4 To automate the whole sales process.

In busy sales system staff need to just select the medicine name all information related to the medicine will auto comes up and staff will just provide the quantity of medicine and as many medicine as client want and can print the invoice.

2.4.5 Get all reports by date to date easily.

Using the system owner can easily generate and check the purchase statement, sales statement and account balance by date to date also can print them. Only admin can do this. Staff only can sell products.

2.5 Methodologies:

To develop this project first is to make the requirement analysis. Depending on the user requirements choose the Incremental Process Model to develop the system and follow this model in entire project of development.

2.6 Software Process model:

The process model for this software is selected as Waterfall process model.

The reasons for choosing this model are given below:

- This project is a small project.
- This model is simple and easy to understand and use.
- Its process activities are clearly separated and organized.
- Waterfall model works well of smaller projects where requirements are very well understood.
- All the requirements for this project are already known.
- Technology is understood.

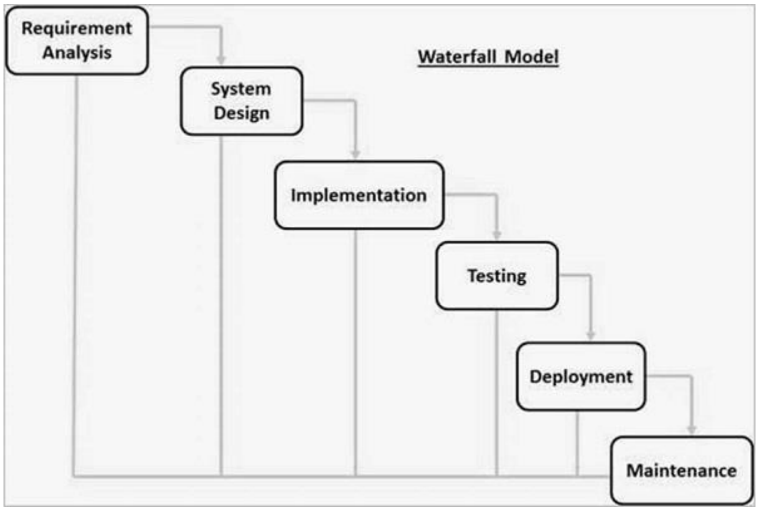


Figure 2.1 *Water Fall Model*

2.7 Feasibility Study:

Feasibility study determines whether that solution is feasible or achievable for the organization. There are three major areas of feasibility study.

- Technical feasibility
- Economic feasibility
- Operational feasibility

2.7.1 Technical Feasibility:

Technical feasibility addresses concerns about hardware capability, reliability and availability and the skills of the development team. This study looks at the hardware and software available to perform the necessary steps for the proposed system. I have identified several software and hardware requirements for the implementation of our system. The hardware and software requirements are as follows:

- MYSQL Server
- PhpStrom or any other Text Editor
- Power backup for electricity failure

2.7.2 Economic Feasibility:

Economic feasibility determines to what extent a new system is cost effective. We are developed this system for Pharmacy Management Software for New Popular Hospital. Store records of purchase product and sales info is costly, but suing this system all process will be aquatic.

2.7.3 Operational Feasibility:

Operational feasibility addresses concerns about staff acceptance, management support, and the requirements of entities. The proposed system will be desirable with the existing way of their Online entry system. As the system will cover all the requirements of user it will become very easy for them to sell and inventory. During our conversation with them we found that they are very much interested and eager to use this system.

Chapter 3 | Requirement Engineering

The process to gather the software requirements from client, analyze and document them is known as requirement engineering. [RE:1] Requirements engineering is closely related to software engineering, which focuses more on the process of designing the system that Staffs want.

We should try to understand what sort of requirements may arise in the requirement elicitation phase and what kinds of requirements are expected from the software system. [RE:1]

Broadly software requirements should be categorized in two categories:

- User requirements
- System requirements
- Functional requirements
- No - Functional requirements

3.1 User Requirements:

A. Admin

a) Admin will be able to add Staff, who can only Sales.

b) Staff will be able to Sign In their own software part to use.

c) Admin will be able to Create Medicine Info.

d) Admin will be able to Create Supplier Info.

e) Admin will be able to manage Inventory part.

f) Admin will be able to manage Sales part.

g) Admin will be able to manage Accounted Part.

B. Staff

a) Staff will be able to Sales Medicine.

b) Staff will be able to check Medicine Quantity.

c) Staff will be able to Print Invoice.

3.2 System Requirements:

A. Admin will be able to add Staff, who can only Sales.

- Admin have to first login into the system.
- System will check is it admin or not.
- Admin will Insert Staff information with Staff username and password.

- Form Validation will work.
- Admin will be able to remove Staff Information.

B. Staff will be able to Sign In their own software part to use.

- Staff will sign in and use their specific part permitted by Admin.
- Staff can't modify their Staff username and password.

C. Admin will be able to Create Medicine Info.

- Admin will be able to insert Medicine Presentation, Generic Name, Medicine Name.
- Admin will be able to delete Medicine Presentation, Generic Name, Medicine Name.

D. Admin will be able to Create Supplier Info.

- Admin will be able to insert supplier Name, Phone Number, Address, Previous Due.
- Admin will be able to delete supplier Info.

E. Admin man will be able to manage Inventory part.

- Store man will have to Login to system.
- System will check the login.
- Then, Store man will be able Insert purchase product with price, amount and selling price and check stock storage and product expire date.

F. Admin will be able to manage Sales part.

- Admin will need to Login to system.
- System will check the login.
- Admin will be able to sales product, check product storage and daily sales information.

G. Admin will be able to manage Accounted part.

- Admin will have to Login to system.
- System will check the login.
- Will only check date to date purchase and selling statement.

H. Staff will be able to Sales Medicine.

- Staff or Sales man will be able to sales medicine to new or old customer.
- Staff will be able to check medicine stock.
- Staff will be able to see sales list.

I. Staff will be able to check Medicine Quantity.

- Staff will be able to check medicine stock at sales time.
- Also will be able to check medicine expire date.

J. Staff will be able to Print Invoice.

- Staff will be able to print sales invoice.
- In invoice there will be invoice number, client email, medicine list and total amount & discount.

3.3 Functional Requirements:

- Admin can maintain whole system.
- Admin can add, delete and edit Staff.
- Admin will able to print date to date loss-profit statement.
- Admin will able to search sales report by date, client id, medicine id.
- Staff will be able to print sales invoice for the client.
- Admin will be able to export excel file for owner.

3.4 Non-Functional Requirements:

- Responsive Design.
- Will be able to use the system from any browser with any device like, mobile, tab, computer.
- Total secured system.
- Easy interface.
- Logging system with valid Staff username and password.
- Mush have Internet connection to access.

3.5 Use Case Diagram

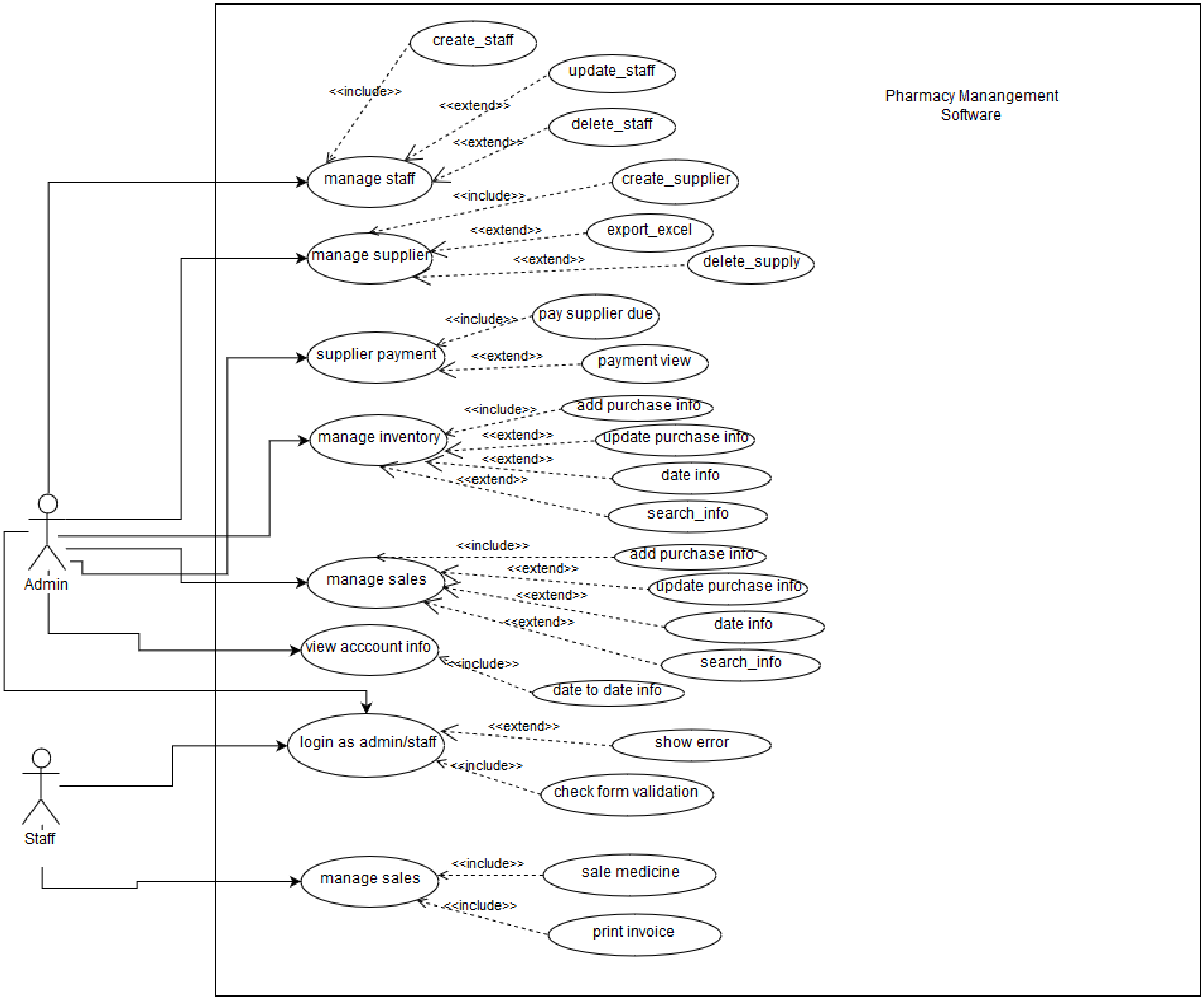


Figure 3.1: Use Case Diagram

3.6 Use Case Text

3.6.1 Admin

Use case Title: Login

Actor: Admin

Description: Admin can login with their username and password

Use case Title: Manage Stuff

Actor: Admin

Description: Admin can add new staff and edit username and password, also can delete staff account.

Use case Title: Manage Supplier

Actor: Admin

Description: Admin can add supplier with information and can delete info. And export data as CSV.

Use case Title: Manage Inventory

Actor: Admin

Description: Admin can add purchase information and can delete or update information.

Use case Title: Manage Sales

Actor: Admin

Description: Admin can sell medicine with all privilege and can print invoice. Also can check Sales statement.

Use case Title: Manage Account

Actor: Admin

Description: Admin can check date to date purchase and sales information and also print.

3.6.2 Staff

Use case Title: Login

Actor: Staff

Description: Staff can login with their username and password

Use case Title: Manage Sales

Actor: Staff

Description: Staff can only sale medicine.

Chapter 4 | System Planning

4.1 Functions of Proposed System:

Login in to the system	F1
Add Staff	F2
Delete Stuff	F3
Add Supplier	F4
Delete Supplier	F5
Pay supplier	F6
Add medicine presentation name	F7
Delete medicine presentation name	F8
Add generic name	F9
Delete generic name	F10
Add medicine name	F11
Delete medicine name	F12
Add purchase info by date	F13
Delete purchase info	F15
Update purchase info	F16
Search date to date purchase	F17
Insert sales info by date	F18
Delete sales info	F19
Search date to date sales info	F20
Report generate	F21
Print sales invoice	F23
Print account info	F24
Date to date account data search	F25
View purchase info	F26
View Sales info	F27
View Account info	F28

4.2 System Project Planning

Before starting any project, it is compulsory to estimate the work to be done, the resources that will be required, the time that will elapse from start to finish and to analyze the project to determine whether it is feasible or not.

The following activities of software project planning that have followed in this project are:

- System Project Estimation
- Function Oriented Metrics
- Process Based Estimation
- Effort Distribution
- Task Scheduling
- Project Schedule Chart
- Cost Estimation

4.2.1 System Project Estimation

The accuracy of a software project estimate predicated based on a number of things:

- Properly estimated the size of the product to build.
- The ability to translate the size estimation into human effort, calendar time and money.
- The degree to which the project plan reflects the abilities of the software team or engineer.
- The stability of the product requirements and the environment that supports the software engineering effort.

Software size estimation is the most important matter that I have to consider during the software project. If the software size not calculate properly, then this will cause various problems such as scheduling problems, budget problem etc. As the project goes on before estimating the software size, I have to confirm that software scope is bounded.

4.2.2 Function Oriented Metrics

Function point-based estimation focuses on information domain values rather that software values. Function points are computed by comparing five information domain characteristics. The information domain values are as follows

Number of external inputs (EI) – Each user input that provides distinct application-oriented data to the software is counted inputs should be distinguished from inquires.

Number of external outputs (EO) – Each user output that provides application-oriented information to the user is counted.

Number of external inquires (EQ) – An inquiry defined as an on-line input those results in the generation of some immediate software response in the form of an on-line output. Each distinct inquiry counted.

Number of Internal logical files (ILF) – Each logical master file counted. Database table wherefrom input goes for modified by application.

Numbers of external interfaces files (EIF) – All machine-readable interfaces that used to transmit information to another system counted.

The weights of the domains are fixes, which are provided in appropriate table location. Weights can be divided into three categories according to the functionality of the system. They are simple, average and complex. The total system is a complex system but the part of the total system. Once these data has collected, a complexity value is associated with each count. To find out the FP count the following formula is used,

Value Adjustment Factor (VAF) = (0.65+ (.01X TDI)) UFP = UFP (Data Fn) + UFP (Transaction Fn) Adjusted Function Point Count (AFP) = UFP X VAF Effort for PHP = AFP x Productivity. **[RE:4]**

4.2.3 Function Point Estimation:

4.2.3.1 Identifying Complexity (Transition Function): Admin

Table 4.1: Identifying complexity (Admin)

Transition Function	Fields/ File Involve	FTRs	DETs
Login in to the system(EI)	**Fields-** id, username, password, identity **File name**- admin	1	3
Add Staff(EI)	**Fields**- stuff_id, username, password, identity **File name**- stuff	1	4
Delete Stuff (EI)	**Fields-** stuff_id, username, password, identity	1	4

	File name- stuff		
Add Supplier (EI)	**Fields**- supplier_id, supplier_name, supplier_phone, supplier_address, previous_due **File name**- create_supplier, insert_purchase_info	2	5
Delete Supplier (EI)	**Fields**- supplier_id, supplier_name, supplier_phone, supplier_address,previous_due **File name**- create_supplier	1	5
Pay supplier (EI)	**Fields**- purchase_id, date, invoice_id, particular, purchase _paid **File name**- insert_purchase_info	1	5
Add medicine presentation name (EI)	**Fields**- medicine_presentation_id, medicine_presentation **File name**-create_medicine_presentation	1	2
Delete medicine presentation name(EI)	**Fields**- medicine_presentation_id, medicine_presentation **Filename**-create_medicine_presentation	1	2
Add generic name(EI)	**Fields**- generic_id, generic_name **File name**- create_generic	1	2
Delete generic name(EI)	**Fields**- generic_id, generic_name **File name**- create_generic	1	2
Add medicine name(EI)	**Fields**- medicine_name_id, medicine_name,generic_id,generic_name **File name**- create_medicine_name, create_generic	4	2
Delete medicine name(EI)	**Fields**- medicine_name_id, medicine_name,generic_id,generic_name **File name**- create_medicine_name	1	2

Add purchase info by date(EI)	**Fields-** purchase_id, date, invoice_id, particular, medicine_presentation_id, medicine_presentation, medicine_name_id, medicine_name, generic_id, generic_name, supplier_id, supplier_name, unit_price,qty, purchase_price, unit_sales_price, unit, purchase _paid, purchase _due, expiredate **File name-** insert_purchase_info, create_medicine_name, create_generic, create_supplier,create_medicine_presentation	5	20
Delete purchase info (EI)	**Fields-** purchase_id, dateTime, invoice_id, particular, medicine_presentation_id, medicine_presentation, medicine_name_id, medicine_name, generic_id, generic_name, supplier_id, supplier_name, unit_price,qty, purchase_price, unit_sales_price, unit, purchase _paid, purchase _due, expiredate **File name-** insert_purchase_info	1	20
Update purchase info(EI)	**Fields-** purchase_id, dateTime, invoice_id, particular, medicine_presentation_id, medicine_presentation, medicine_name_id, medicine_name, generic_id, generic_name, supplier_id, supplier_name, unit_price,qty, purchase_price, unit_sales_price, unit, purchase _paid, purchase _due, expiredate **File name-** insert_purchase_info	1	20
Search date to date purchase(EQ)	**Fields-** purchase_id, dateTime, invoice_id, particular, medicine_presentation_id, medicine_presentation, medicine_name_id, medicine_name, generic_id, generic_name, supplier_id, supplier_name, unit_price,qty,	1	20

	purchase_price, unit_sales_price, unit, purchase _paid, purchase _due, expiredate **File name-** insert_purchase_info		
Insert sales info by date(EI)	**Fields-** sales_id, invoice_id, date, particular, customer_emai, medicine_name_id, medicine_name, generic_id, generic_name, medicine_presentation_id, medicine_presentation,unit_sales_price, qty, price, total_price, total_discount, discount_price, sales_paid, sales_due **File name-** sales_product, insert_purchase_info	2	19
Delete sales info(EI)	**Fields-** sales_id, invoice_id, date, particular, customer_emai, medicine_name_id,medicine_name, generic_id, generic_name, medicine_presentation_id, medicine_presentation,unit_sales_price, qty,price, total_price,total_discount, discount_price,sales_paid,sales_due **File name-** sales_product	1	19
Search date to date sales info(EQ)	**Fields-** sales_id, invoice_id, date, particular, customer_emai, medicine_name_id,medicine_name, generic_id, generic_name, medicine_presentation_id, medicine_presentation,unit_sales_price, qty,price, total_price,total_discount, discount_price,sales_paid,sales_due **File name-** sales_product	1	19

Report generate(EQ)	**Fields-** sales_id, invoice_id, date, particular, customer_emai, medicine_name_id,medicine_name, generic_id, generic_name, medicine_presentation_id, medicine_presentation,unit_sales_price, qty,price, total_price,total_discount, discount_price,sales_paid,sales_due **File name-** sales_product	1	19
Print sales invoice(EO)	**Fields-** sales_id, invoice_id, date, particular, customer_emai, medicine_name_id,medicine_name, generic_id, generic_name, medicine_presentation_id, medicine_presentation,unit_sales_price, qty,price, total_price,total_discount, discount_price,sales_paid,sales_due **File name-** sales_product	1	19
Print account info(EO)	**Fields-** Purchase:date,qty,purchase_amount Sales: date, qty, discount_price **File name-** sales_product, insert_purchase_info	2	6
Date to date account data search (EQ)	**Fields-** Purchase: date,qty,purchase_amount Sales: date, qty, discount_price **File name-** sales_product, insert_purchase_info	2	6
View purchase info(EO)	**Fields-** sales_id, invoice_id, date, particular, customer_emai, medicine_name_id,medicine_name,	1	19

	generic_id, generic_name, medicine_presentation_id, medicine_presentation,unit_sales_price, qty,price, total_price,total_discount, discount_price,sales_paid,sales_due **File name-** sales_product		
View Sales info(EO)	**Fields-** sales_id, invoice_id, date, particular, customer_emai, medicine_name_id,medicine_name, generic_id, generic_name, medicine_presentation_id, medicine_presentation,unit_sales_price, qty,price, total_price,total_discount, discount_price,sales_paid,sales_due **File name-** sales_product	1	19
View Account info(EO)	**Fields-** Purchase:date,qty,purchase_amount Sales: date, qty, discount_price **File name-** sales_product, insert_purchase_info	2	6

4.2.3.2 Identifying Complexity (Transition Function): Staff

Table 4.2: Identifying complexity (Staff)

Transition Function	Fields/ File Involve	FTRs	DETs
Login in to the system(EI)	**Fields-** staff_id, username, password, identity **File name-** admin	1	3
Insert sales info by date(EI)	**Fields-** sales_id, invoice_id, date, particular, customer_emai, medicine_name_id, medicine_name, generic_id, generic_name, medicine_presentation_id, medicine_presentation,unit_sales_price, qty, price, total_price, total_discount, discount_price, sales_paid, sales_due **File name-** sales_product, insert_purchase_info	2	19

4.2.3.3 Identifying Complexity (Data Function):

Table 4.3 Identifying complexity

Data Function	Fields/ File Involve	RETs	DETs
Create_medicine_name (ILF)	**Fields-** medicine_name_id, medicine_name,generic_id,generic_name **File name-** create_medicine_name, create_generic	2	4
Sales_product (ILF)	**Fields-** sales_id, invoice_id, date, particular, customer_emai, medicine_name_id, medicine_name, generic_id, generic_name, medicine_presentation_id,	2	19

	medicine_presentation,unit_sales_price, qty, price, total_price, total_discount, discount_price, sales_paid, sales_due **File name-** sales_product, insert_purchase_info		
Insert_purchase_info(ILF)	**Fields-** purchase_id, date, invoice_id, particular, medicine_presentation_id, medicine_presentation, medicine_name_id, medicine_name, generic_id, generic_name, supplier_id, supplier_name, unit_price,qty, purchase_price, unit_sales_price, unit, purchase _paid, purchase _due, expiredate **File name-** insert_purchase_info, create_medicine_name, create_generic, create_supplier,create_medicine_presentation	5	20

4.2.3.4 Unadjusted function point contribution

Table 4.4: Unadjusted function point contribution

SL	Transition Function	FTRs	DETs	Complexity	UFP
1	Login in to the system(EI)	1	3	**Low**	3
2	Add Staff(EI)	1	4	**Low**	3
3	Delete Stuff (EI)	1	4	**Low**	3
4	Add Supplier (EI)	2	5	**Average**	4
5	Delete Supplier (EI)	1	5	**Low**	3
6	Pay supplier (EI)	1	5	**Low**	3
7	Add medicine presentation name (EI)	1	2	**Low**	3
8	Delete medicine presentation name(EI)	1	2	**Low**	3
9	Add generic name(EI)	1	2	**Low**	3

10	Delete generic name(EI)	1	2	**Low**	3
11	Add medicine name(EI)	4	2	**Average**	4
12	Delete medicine name(EI)	1	2	**Low**	3
13	Add purchase info by date(EI)	5	20	**High**	6
14	Delete purchase info (EI)	1	20	**Average**	4
15	Update purchase info(EI)	1	20	**Average**	4
16	Search date to date purchase(EQ)	1	20	**Average**	4
17	Insert sales info by date(EI)	2	19	**High**	6
18	Delete sales info(EI)	1	19	**High**	6
19	Search date to date sales info(EQ)	1	19	**Low**	3
20	Report generate(EQ)	1	19	**Low**	3
21	Print sales invoice(EO)	1	19	**Low**	3
22	Print account info(EO)	2	6	**Average**	4
23	Date to date account data search (EQ)	2	6	**Average**	4
24	View purchase info(EO)	1	19	**Low**	3
25	View Sales info(EO)	1	19	**Low**	3
26	View Account info(EO)	2	6	**Low**	3
				Total	238

4.2.3.5 Unadjusted Function Point Contribution (Data Function):

Table 4.5: Unadjusted function point contribution

Data Function	RETs	DETs	Complexity	UFP
Create_medicine_name (ILF)	2	4	**Low**	7
Sales_product (ILF)	2	19	**Low**	7
Insert_purchase_info(ILF)	5	20	**Low**	7
			Total	21

4.2.3.6 Performance and Environmental Impact

Table 6.6: Performance and Environmental Impact

SL	**GSC**	**TDI**
	Data Communication	1
	Distributed Data Processing	0
	Distributed Data Processing	1
	Heavily Used Configuration	1
	Transaction Rate	2
	Online Data Entry	3
	End-user Efficiency	4
	Online Update	2
	Complex Processing	2
	Reusability	3
	Installation Ease	3
	Operational Ease	3
	Multiple Sites	3
	Facilitate Change	3
Total Degree of Influence (TDI) (Range 0 to 70->influence size by +-32%)		**32**

Value adjustment factor (VAF) = (0.65+ (0.01* TDI))

= (0.65+ (0.01* 32))

= 0.97

UFP= UFP (Data function) + UFP (Transaction function)

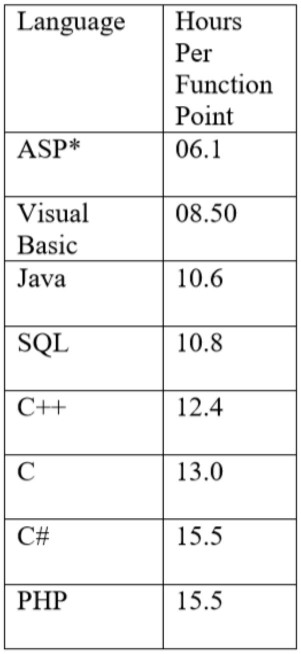
= 21 + 238 = 259

Total time calculation frame = 259 * 15.5

= 4014 person hours / 9 hours

= 446 person days / 22 days

= 20 person months / 2 persons



Language	Hours Per Function Point
ASP*	06.1
Visual Basic	08.50
Java	10.6
SQL	10.8
C++	12.4
C	13.0
C#	15.5
PHP	15.5

10 months for two persons, Approximately 4 months required for five persons to finish the project.

4.2.7. Project Schedule Chart

Total system development is a combination of set of tasks. These set of tasks should done sequentially and timely. Project schedule works as the guideline of the system developer. The following is the schedule chart of this project:

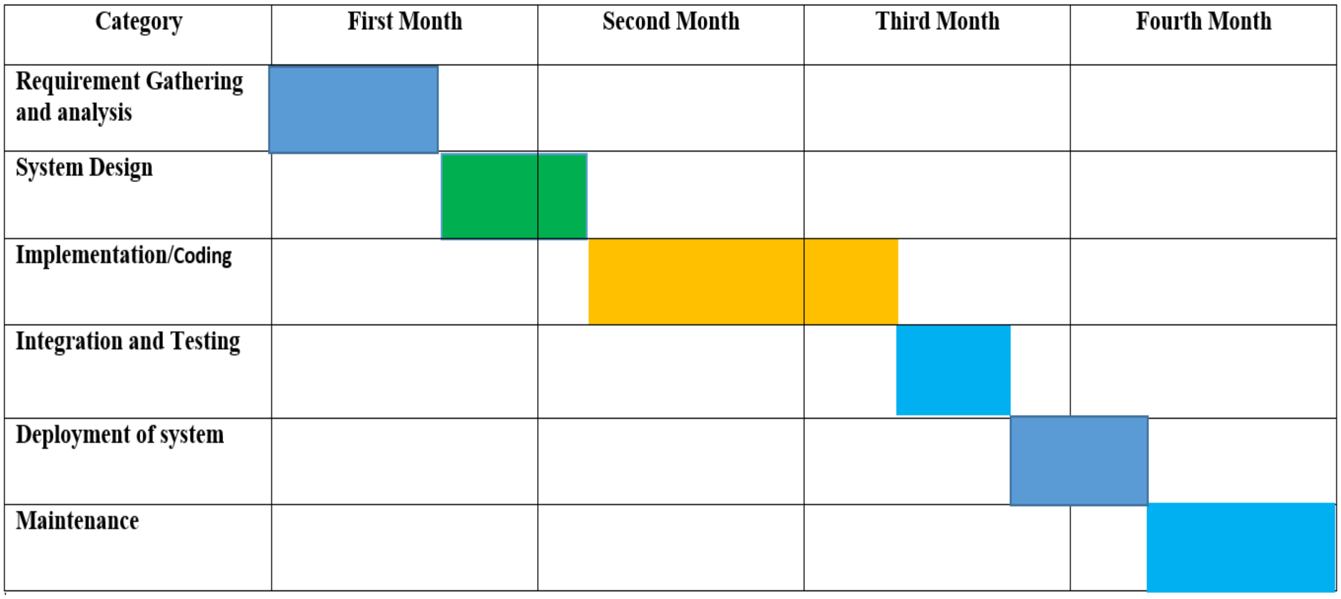


Figure 4.1: *Project scheduling chart*

4.2.8. Cost Estimation

The approximation of the cost of a program is cost estimation. In this project, there are five factors to analyze and calculate the cost. Given bellow,

- Personnel cost
- Software cost
- Hardware cost
- Other cost

Personnel cost

• Number of days in a year = 365

• Number of government holidays in a year =24

• Number of weekly holidays in a year =52

- Total number of working days to develop the project =365-(52+24) =289 days

- Total number of working days per months to develop the project =289/12 =24.083 days

- Organization working hours per day = 9 hours

- Organization working hours per month = 24.083*9 = 216.747 hours

Table 4.7: Personnel cost

Type	No. of Members	Months	Salary
System Analyst & Customer Communicator	2	1	20,000.00
Coder, Tester & Designer	2	1	7000.00
Total			27,000.00

Hardware Cost

Cost of the computer that used to complete the project.

Table 4.8: Personnel cost

Name	Number	Price	Description	Total
Lenovo 310(laptop Computer)	1	38,500	38,500 / 24*4=	401

Total Hardware Cost = 401.00 TK

Software Cost

It is the cost of the software is which used in this project.

Table 4.9: Software Cost

SL	Software	Number	Price
1	Windows 10	1	Student pack (free)
2	XAMPP control panel	1	Free
3	PhpStrom 2018	1	Student Lenience(free)
4	Google Chrome	1	Free

Accounts table

Table 4.10: Total Cost

Particular	Tk
Salary System Analyst Designer & Customer Communicator Coder &Tester	27,000 tk
Hardware Cost	401tk
Software Cost	0 tk
Other – pen, paper, printout, transport	1000tk
Total	28,401tk

Chapter 5 | Risk Management

5.1 Risk Analysis

Risk analysis and management are a series of works that help a system development team to understand and manage uncertainty. Many problems can arise while developing a system. A risk is a potential problem – it may happen may not. There are several steps to analyze and manage risks. The first step is risk identification. Next each risk is analyzed to determine the likelihood that it will occur and the damage that it will do if it does occur. Once this information is established risks are remarked. Finally, a plan is developed to manage those risks with high probability and impact.

There are different Stages of risks. These are:

1. Risk identification: Risk identification is the process of detecting potential risks or hazards through data collection. A range of data collection and manipulation tools and techniques exists. The team is using both automated and manual techniques to collect data and begin to characterize potential risks to Web resources. Web crawling is one effective way to collect information about the state of Web pages and sites.

2. Risk classification: Risk classification is the process of developing a structured model to categorize risk and fitting observable risk attributes and events into the model. The team combines quantitative and qualitative methods to characterize.

3. Risk assessment: Risk assessment is the process of defining relevant risk scenarios or sequences of events that could result in damage or loss and the probability of these events. Many sources focus on risk assessment. Rosenthal describes the characteristics of a generic standard for risk assessment as "transparent, coherent, consistent, complete, comprehensive, impartial, uniform, balanced, defensible, sustainable, flexible, and accompanied by suitable and sufficient guidance.

4. Risk analysis: Risk analysis determines the potential impact of risk patterns or scenarios, the possible extent of loss, and the direct and indirect costs of recovery. This step identifies vulnerabilities, considers the willingness of the organization to accept risk given potential consequences, and develops mitigation responses.

5. Risk management implementation: defines policies, procedures, and mechanisms to manage and respond to identifiable risks. The implemented program should balance the value of assets and the direct and indirect costs of preventing or recovering from damage or loss.

To take comprehensive care of a web based system we must consider the following points:

A. Hardware and software environment including any upgrades to the operating system and Web server, the installation of security patches, the removal of insecure services, use of firewalls, etc.
B. Administrative procedures such as contracting with reputable service providers, renewing domain name registration, etc.
C. Network configuration and maintenance including load balancing, traffic management, and usage monitoring.
D. Backup and archiving policies and procedures including the choice of backup media, media replacement interval, number of backups made and storage location.
E. Physical location of the server and its vulnerability to fire, flood, earthquake, electric power anomalies, power interruption, temperature fluctuations, theft, and vandalism.

There are different categories of risks that should be considered in any software project. The following categories of risks have been considered in this software project.

01. Project risks: These risks threaten the project plan. If these risks become real, it is likely that the project schedule will slip and that costs will increase. Project risks identify potential budgetary, schedule, personnel, resource, customer and requirement problems and their impact on the software project.

02. Technical risks: These risks threaten the quality and timeliness of the software to be produced. If a technical risk becomes a reality, implementation may become difficult or impossible. Technical risks identify potential design, implementation, interface, verification and maintenance problems. Moreover, specification ambiguity, technical uncertainty, technical obsolescence are also risk factors.

03. Business risks: These risks threaten the viability of the software to be built. The business risks can be market risks, building a system that no one really wants. Strategic risks, building a system that no longer fits into the overall business strategy for the company. Management risks, losing the support of senior management due to a change in focus or a change in people. Budget risks, losing budgetary or personnel commitment.

Type of Impact:

1. Catastrophic
2. Critical
3. Marginal
4. Tolerable

5.2 The RMMM Plan:

Risk Mitigation: Proactive planning for risk avoidance.

Risk Monitoring: Assessing whether predicted risks occur or not, ensuring preventive steps are being properly applied, collect information for future risk analysis, attempt to determine which risks caused which problem.

Risk Management: Actions to be taken in the event that mitigation steps have failed and the risk has become a live problem.

Type of Impact: Catastrophic (1), Marginal (2), Tolerable (3), Critical (4).

Type of Probability: very low (<10%), low (10–25%), moderate (25–50%), high (50–75%), very high (>75%).

Project Risks: Threaten the project plan. In my system, the bellow mentioned projects risks I needed manage.

ID	Name of the Risk	Type	Probability	Impact	RMMM
B01	Insufficient Budget	B01	12%	3	Plan 1
B02	End Users Resist System	B02	9%	1	Plan 2
P03	Changes in Requirements	P03	15%	3	Plan 3
T04	Lack of Development Experience	T04	25%	2	Plan 4
T05	Poor Comments in Code	T05	5%	4	Plan 5
T06	Server Security Risk	T06	18%	1	Plan 6
T07	Data Redundancy Risk	T07	30%	1	Plan 7

Now the RMMM Plan for the previously mentioned problems:

Table 5. 1: RMMM Plan 1

Business Risk(B01)	Insufficient Budget	05 – March -2019
Probability	12%	**Impact:** Low
Description:	If the budget is low project may not complete.	
Refinement:	The project need streaming server that is costly to set-up	
Mitigation/monitoring:	We find several alternative streaming services to reduce the budget risk.	
Management/Contingency Plan:	Refinement in project goal. A new plan for regulate the budget.	
Current Status	Problem resolved	

Table 5.2: RMMM Plan 2

Business Risk(B01)	End Users Resist System	10 – March -2019
Probability	9%	**Impact:** Very Low
Description:	The system fails to gain user's faith.	
Refinement:	The users do not like the new system or the new system is not performing up to its expectations.	
Mitigation/monitoring:	In order to prevent this from happening, the software will develop with the end user in mind. The user-interface will design in a way to make use of the program convenient and pleasurable.	
Management/Contingency Plan:	Training the users to familiarize them with the new system. Releasing patches/bug fixes for greater user satisfaction.	
Current Status	The risk has not been arisen yet	

Table 5.3: RMMM Plan 3

Business Risk(B01)	End Users Resist System	14 – March -2019
Probability	15%	**Impact:** Low
Description:	The company may change their requirements.	
Refinement:	Requirements are redefined by the company due to time or business needs.	
Mitigation/monitoring:	Meeting will be held with the company regularly. This insures that the product we are producing solves a problem.	
Management/Contingency Plan:	Emergency meeting between both parties to identify new project requirements and goals.	
Current Status	Mitigation steps initiated.	

Table 5.4: RMMM Plan 4

Business Risk(B01)	End Users Resist System	17 – March -2019
Probability	25%	**Impact:** Low
Description:	Stuff inexperience.	
Refinement:	Problems in analysis, problems in design, and problems in testing these problems may occur due to inexperience.	
Mitigation/monitoring:	The development cost of the software may increase by 20%. Consult with the System Analyst during the system analysis, design and testing phase of the software project. Management	
Management/Contingency Plan:	Though the development cost is increased by 20%, the project is still feasible. Set appointment for formal meeting with the System Analyst to solve different problems of each of the phases.	
Current Status	The risk has not been arisen yet.	

Table 5.5: RMMM Plan 5

Business Risk(T05)	End Users Resist System	23 – March -2019
Probability	5	**Impact:** Very Low
Description:	Poor commenting in the coding phase.	
Refinement:	Codes are hard to reuse and study at later stages.	
Mitigation/monitoring:	Proper coding grammar is followed to make sure that the codes are easily understandable and reusable.	
Management/Contingency Plan:	In emergency, a refinement in code commenting should be done. This may slow down the development process but it will help the developers in long run.	
Current Status	The risk has not been arisen yet.	

Table 5.6: RMMM Plan 6

Business Risk(T05)	End Users Resist System	25 – March -2019
Probability	18	**Impact:** Low
Description:	Server Security Risk	
Refinement:	The server is hacked, breached the security by hacker.	
Mitigation/monitoring:	We should follow the proper server configuration and install firewall.	
Management/Contingency Plan:	If the server security compromised, we can restore from backup and ensure the security.	
Current Status	The risk has not been arisen yet.	

Table 5.7: RMMM Plan 7

Business Risk(T07)	End Users Resist System	28 – March -2019
Probability	30	**Impact:** Moderate
Description:	Data Redundancy Risk.	
Refinement:	For any reason, some data are lost.	
Mitigation/monitoring:	We should follow the RAID method to get backup if any data is erase.	
Management/Contingency Plan:	Though any data of any process will be lost we can recover or take backup by RAID method. So, that no harm will occur in the process.	
Current Status	The risk has not been arisen yet.	

Chapter – 6 | Analysis Modeling

Analysis modeling uses a combination of text and diagrammatic forms to depict requirements for data, function, and behavior in a way that is relatively easy to understand, and more important, straightforward to review for correctness, completeness and consistency. This section presents resources for conventional and object-oriented analysis (OOA) methods as well as resources for UML.

6.1 Analysis Modeling Objectives of analysis model

- Domain Analysis
- Describe what the client requires
- Establish a basis for the creation of a software design
- Define a set of requirements that can be validated once the software is built.

6.2 Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes. Activity diagrams show the overall flow of control.

6.2.1 Activity Diagram of Admin

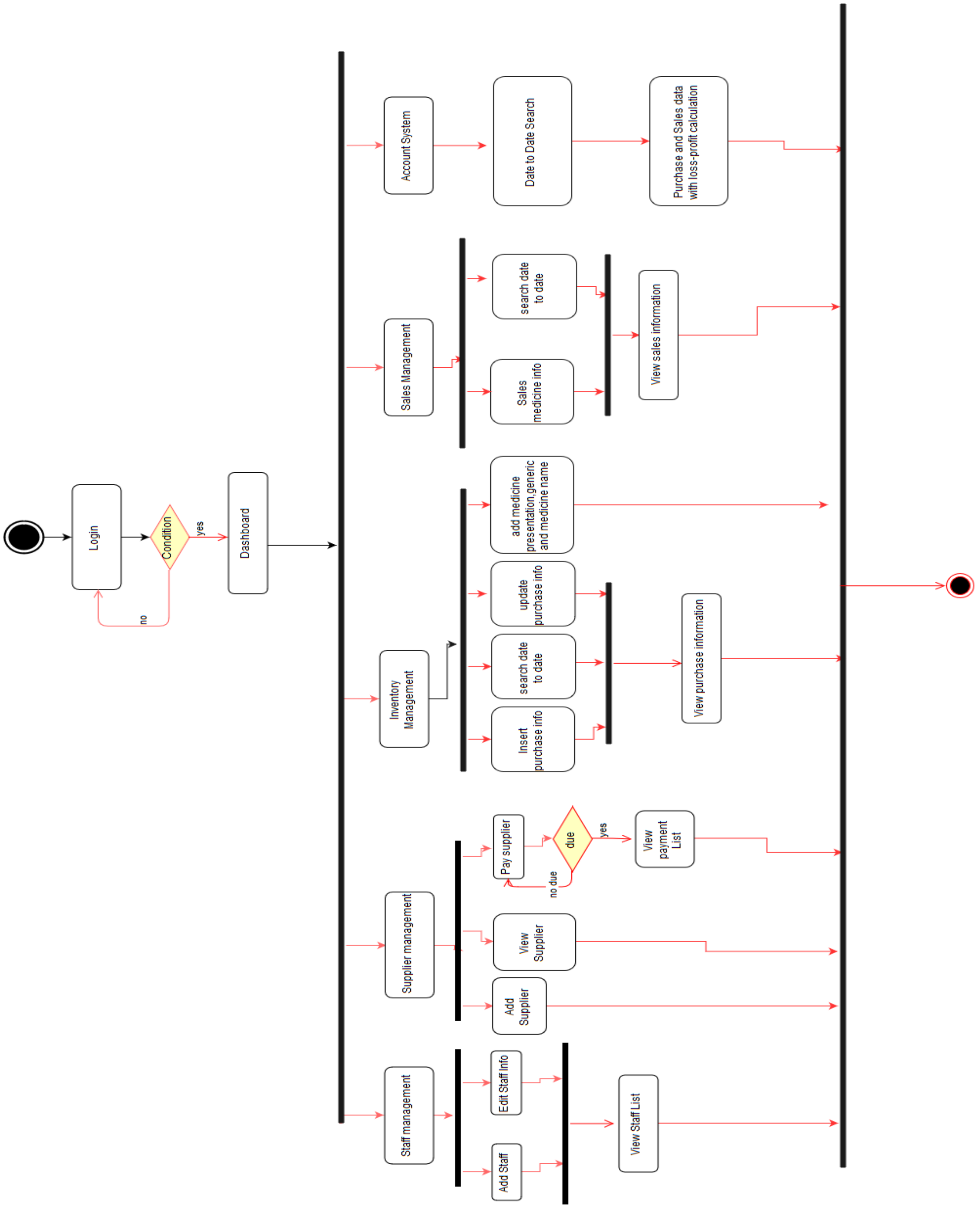


Figure 6.1 : Activity Diagram of Admin

6.2.2 Activity Diagram of Staff

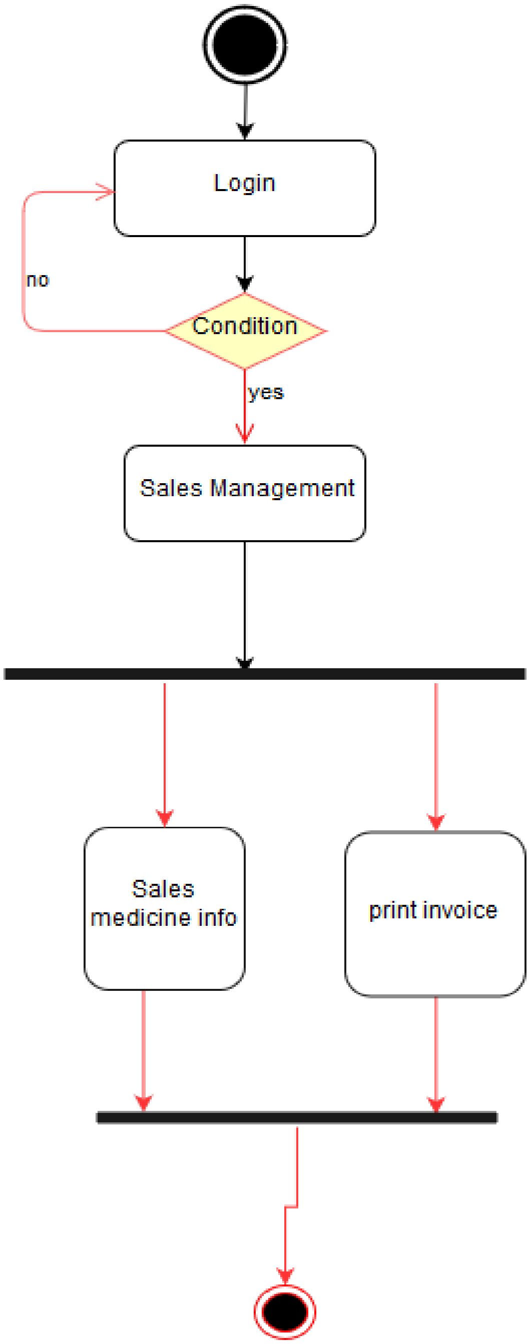


***Figure 6.2:** Activity Diagram of Staff*

6.3 Swim lane Diagram

Swim lane is a visual element used in process flow diagrams, or flowcharts that visually distinguishes job sharing and responsibilities for sub-processes of a business process. Swim lanes may be arranged either horizontally or vertically. In the accompanying example, the swim lanes are named Admin, Faculty, and System timetable and are arranged vertically.

6.3.1 Swim lane Diagram of Admin & Staff:

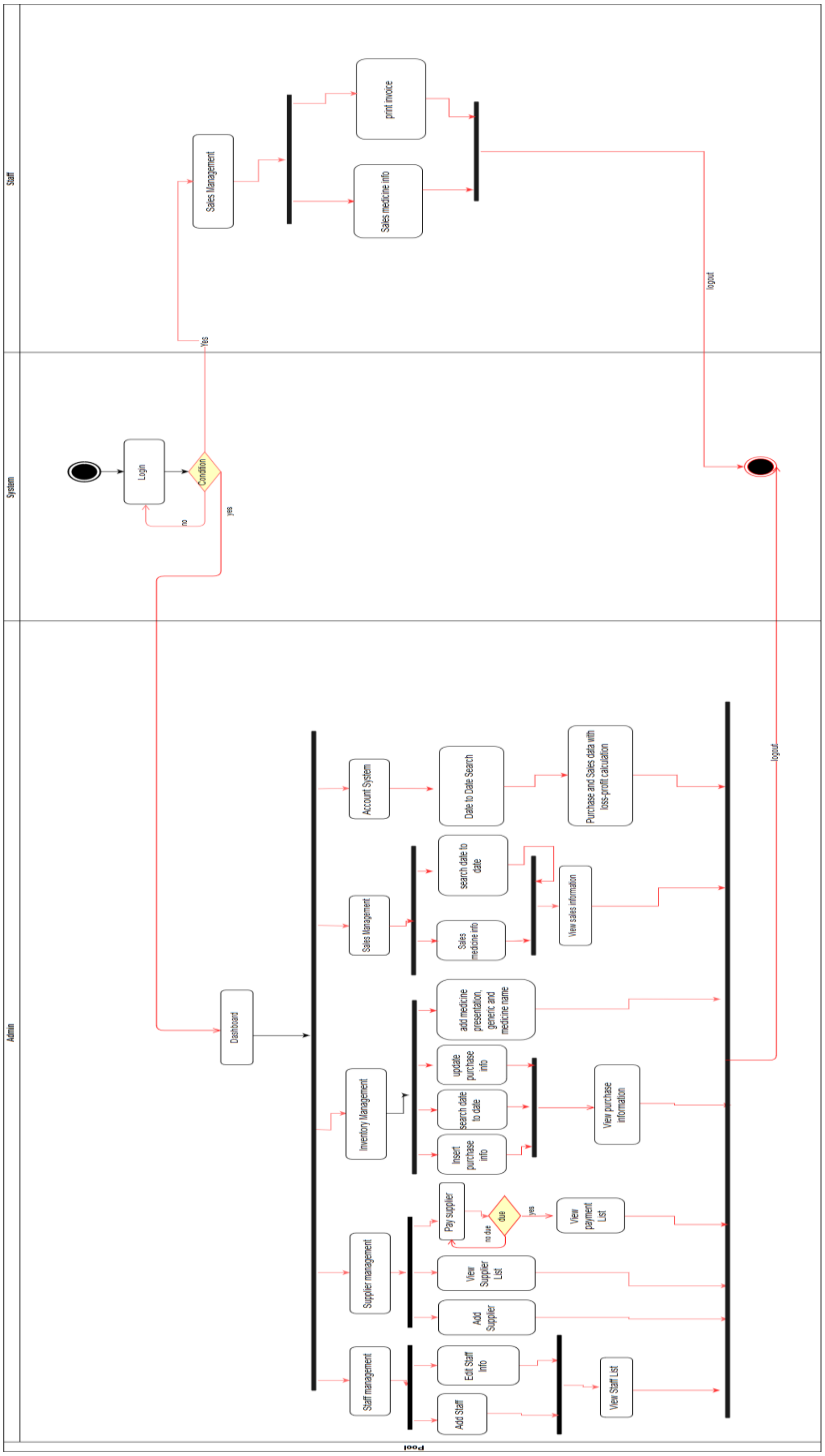


Figure 6.3: Swim lane Diagram

6.4 ER Diagram

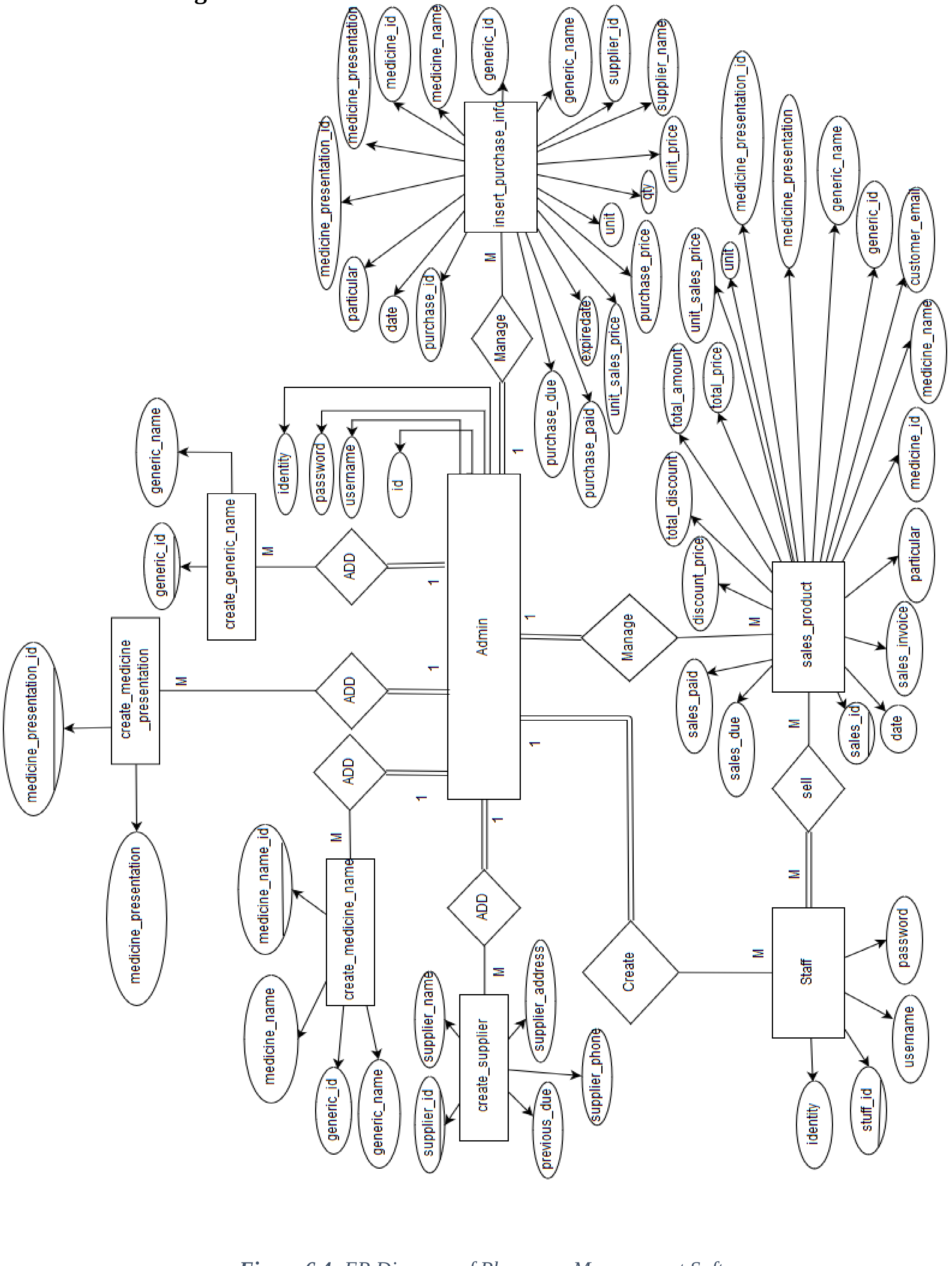


Figure 6.4: ER Diagram of Pharmacy Management Software

6.5 Data Flow Diagram (DFD)

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated DFDs can also, be used for the visualization of data processing. A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel.

6.5.1 Context Level Diagram

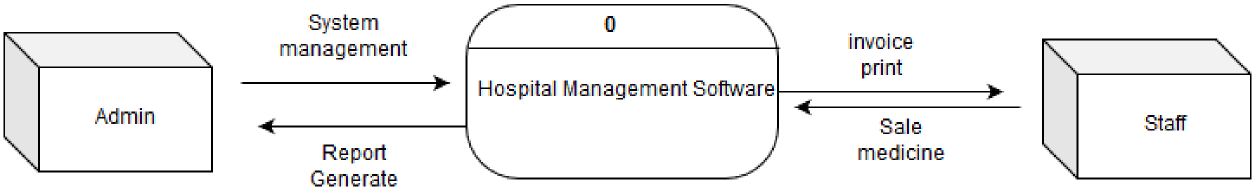


Figure 6.5: Context Level Diagram of Transport Ticket Booking System

6.5.2 Level 1 DFD

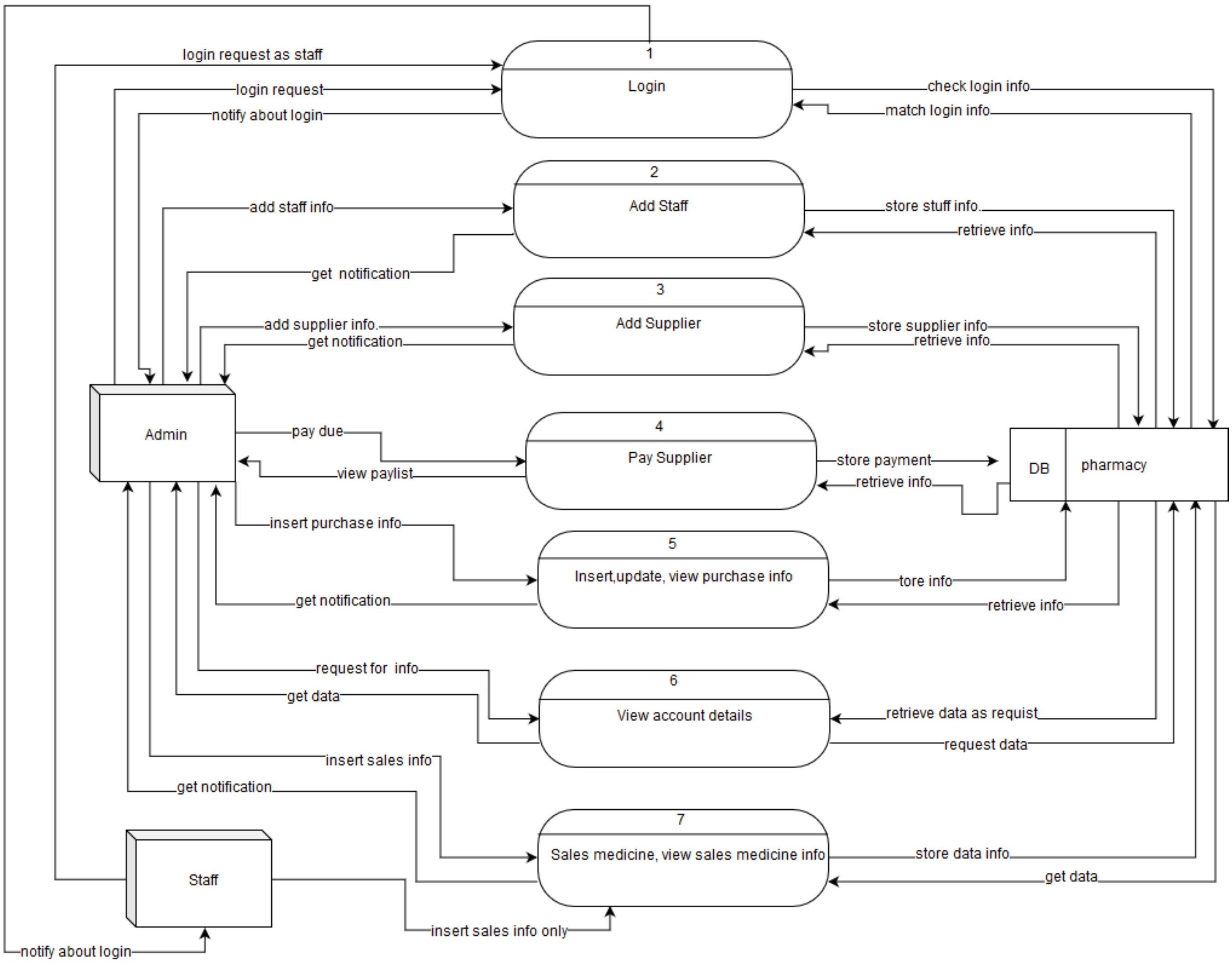


Figure 6.6: DFD Level-1

6.5.3 Level 2 DFD (Login Process)

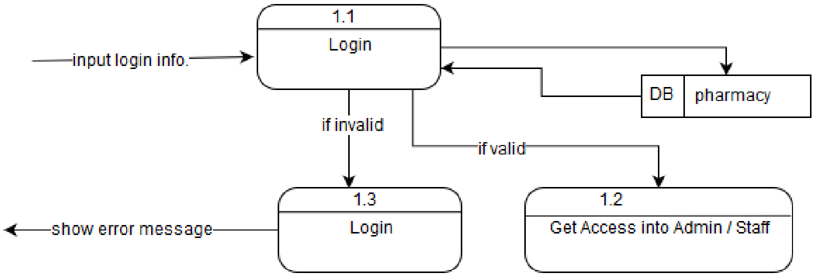


Figure 6.7: Level 2 DFD (Login Process)

6.5.4 Level 2 DFD (Add Stuff)

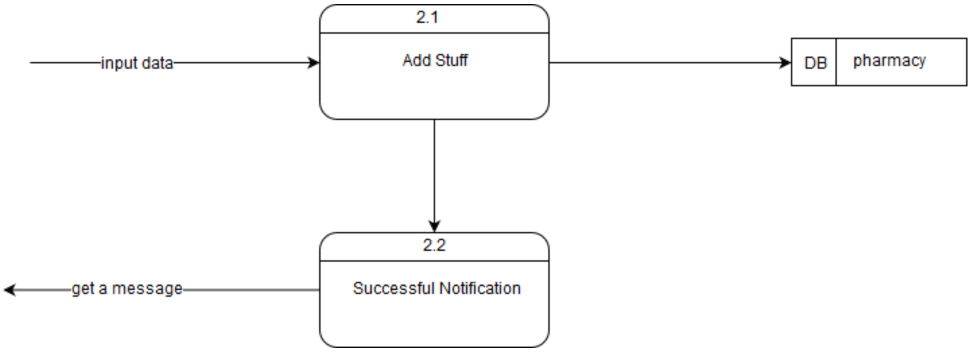


***Figure 6.8:** Level 2 DFD (Add Stuff)*

6.5.5 Level 2 DFD (Add Supplier)

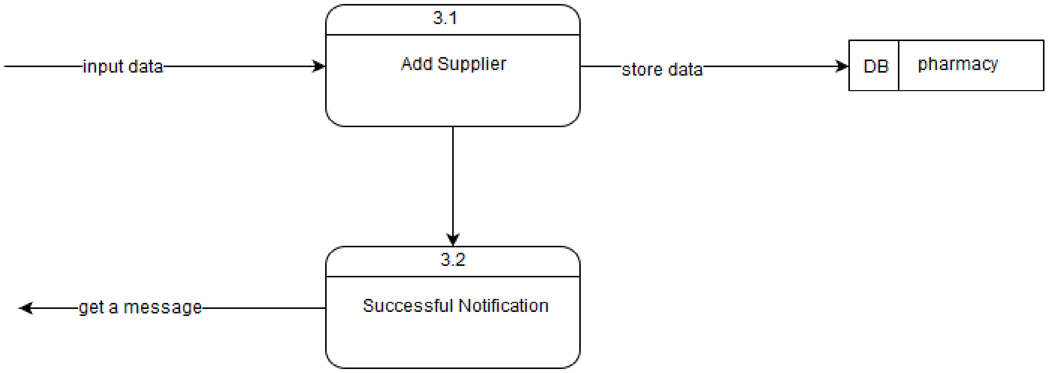


***Figure 6.9**: Level 2 DFD (Add Supplier)*

6.5.6 Level 2 DFD (pay Supplier)

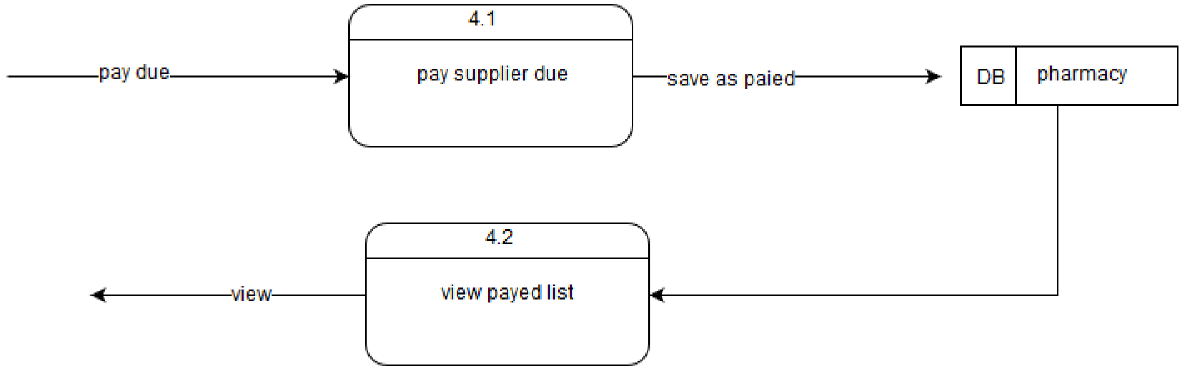


Figure 6.10: Level 2 DFD (pay Supplier)

6.5.7 Level 2 DFD (Insert, Update, view Purchase Info)

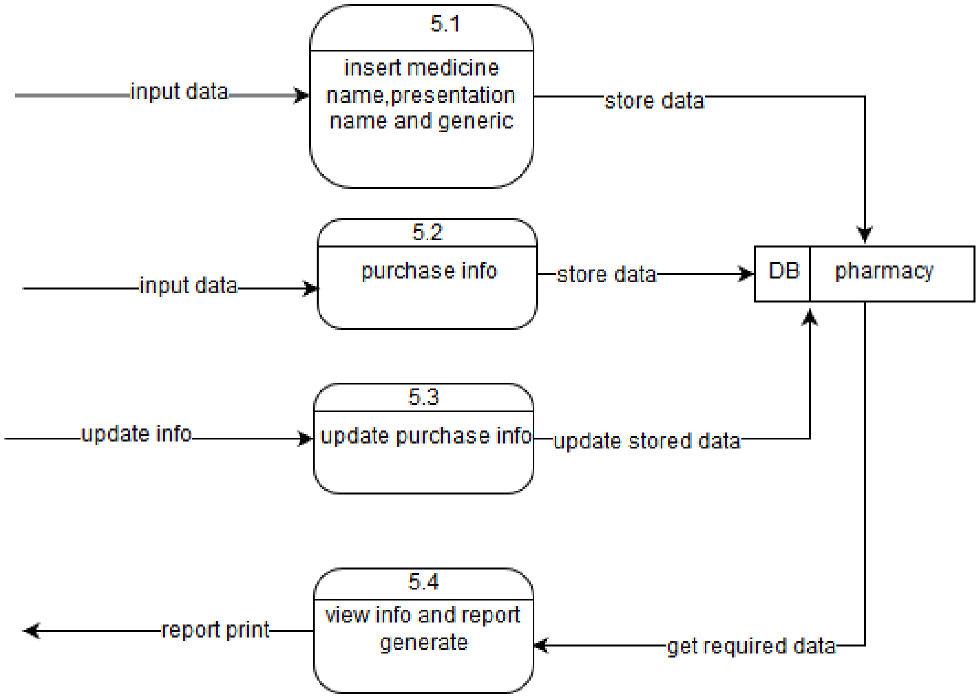


Figure 6.11: Level 2 DFD (Insert, Update, view Purchase Info)

6.5.8 Level 2 DFD (Sales Medicine and view sales info by Admin / staff)

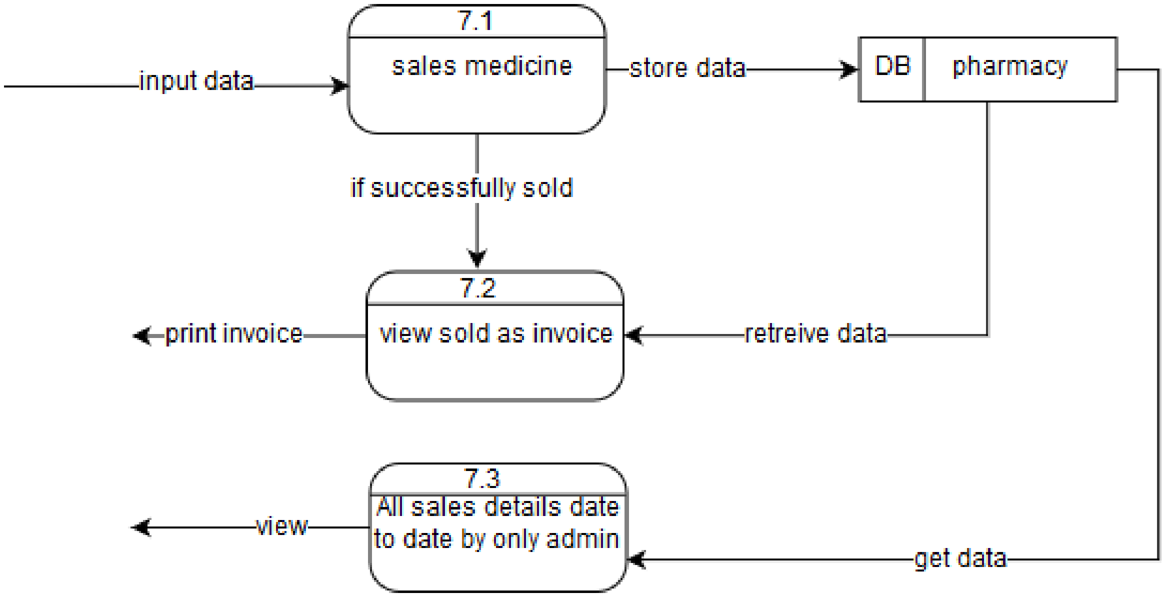
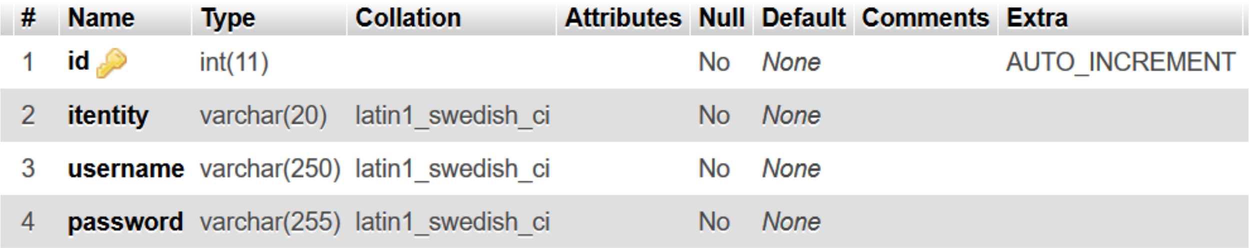


Figure 6.12: Level 2 DFD (sales medicine and View info)

Chapter 7 | Designing

7.1 Database Field Design

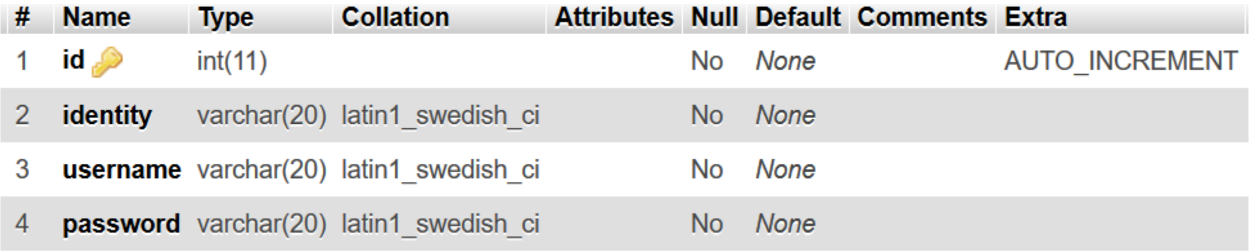
Admin



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id**	int(11)			No	*None*		AUTO_INCREMENT
2	**itentity**	varchar(20)	latin1_swedish_ci		No	*None*		
3	**username**	varchar(250)	latin1_swedish_ci		No	*None*		
4	**password**	varchar(255)	latin1_swedish_ci		No	*None*		

Figure 7.2 *Admin Login Table Structure*

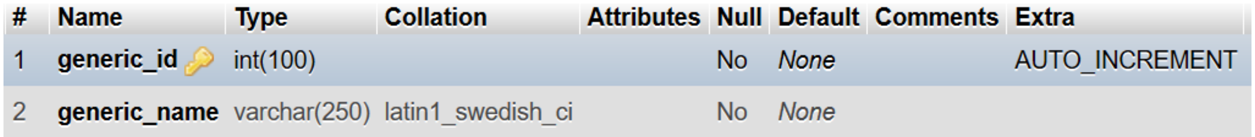
Stuff



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id**	int(11)			No	*None*		AUTO_INCREMENT
2	**identity**	varchar(20)	latin1_swedish_ci		No	*None*		
3	**username**	varchar(20)	latin1_swedish_ci		No	*None*		
4	**password**	varchar(20)	latin1_swedish_ci		No	*None*		

Figure 7.3 *Stuff Login Table Structure*

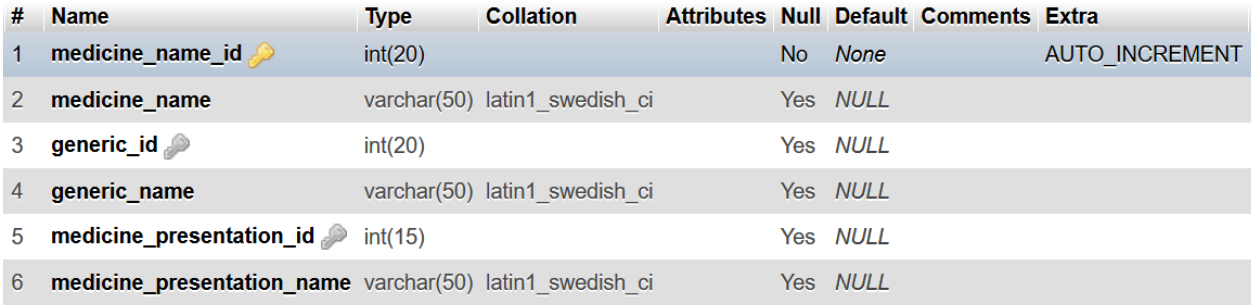
create_generic_name



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**generic_id**	int(100)			No	*None*		AUTO_INCREMENT
2	**generic_name**	varchar(250)	latin1_swedish_ci		No	*None*		

Figure 7.4 *create generic name Table Structure*

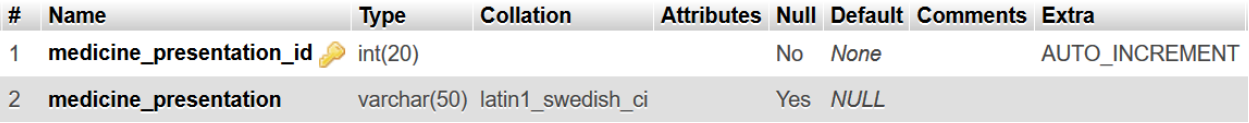
create_medicine_name



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**medicine_name_id**	int(20)			No	*None*		AUTO_INCREMENT
2	**medicine_name**	varchar(50)	latin1_swedish_ci		Yes	*NULL*		
3	**generic_id**	int(20)			Yes	*NULL*		
4	**generic_name**	varchar(50)	latin1_swedish_ci		Yes	*NULL*		
5	**medicine_presentation_id**	int(15)			Yes	*NULL*		
6	**medicine_presentation_name**	varchar(50)	latin1_swedish_ci		Yes	*NULL*		

Figure 7.5 *create medicine name Table Structure*

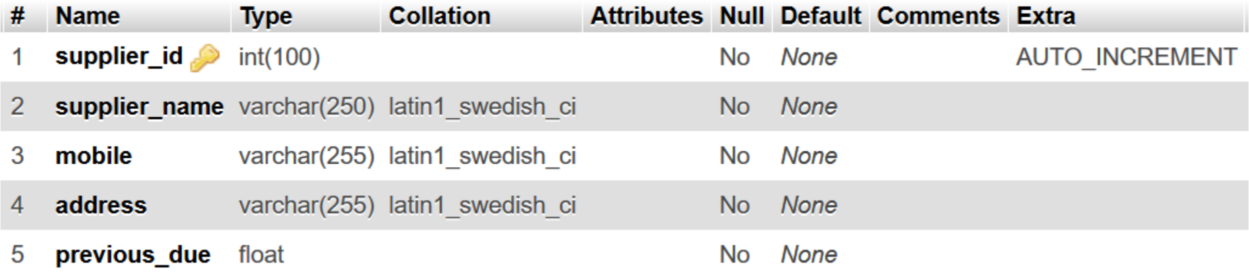
create_medicine_presentation



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**medicine_presentation_id** 🔑	int(20)			No	*None*		AUTO_INCREMENT
2	**medicine_presentation**	varchar(50)	latin1_swedish_ci		Yes	*NULL*		

Figure 7.6 *create medicine name Table Structure*

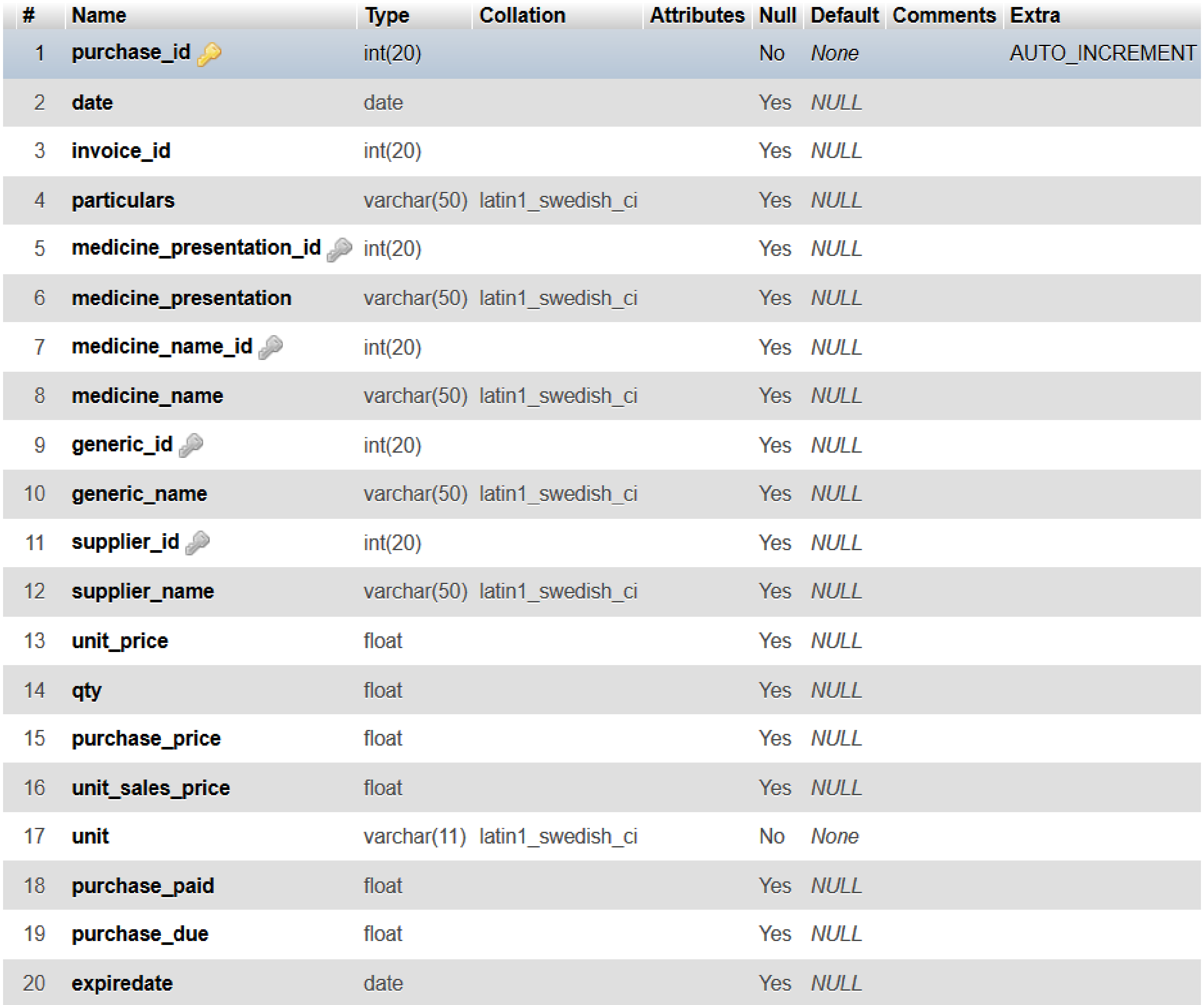
create_supplier



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**supplier_id** 🔑	int(100)			No	*None*		AUTO_INCREMENT
2	**supplier_name**	varchar(250)	latin1_swedish_ci		No	*None*		
3	**mobile**	varchar(255)	latin1_swedish_ci		No	*None*		
4	**address**	varchar(255)	latin1_swedish_ci		No	*None*		
5	**previous_due**	float			No	*None*		

Figure 7.7 *create supplier Table Structure*

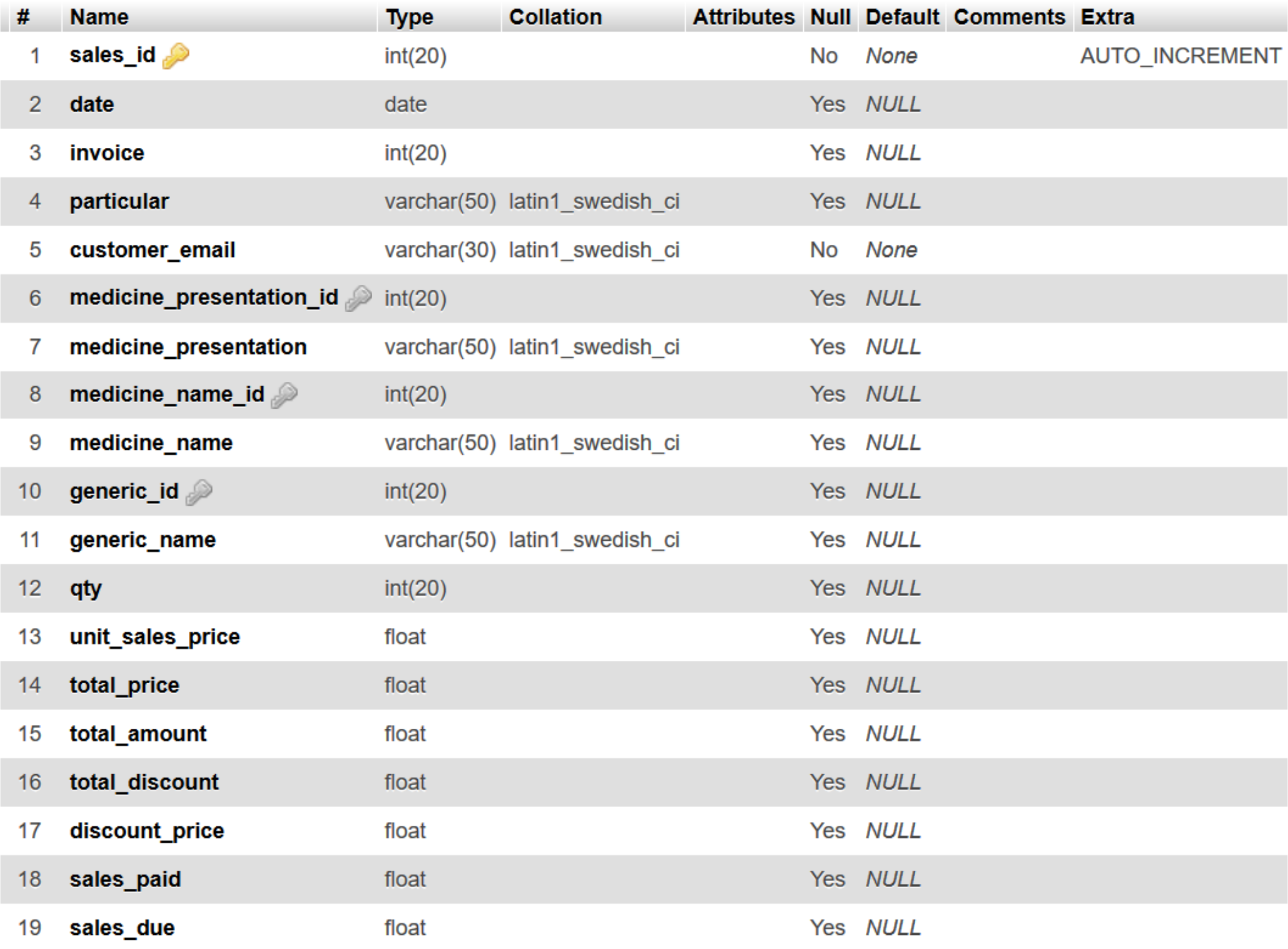
insert_purchase_info



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**purchase_id** 🔑	int(20)			No	*None*		AUTO_INCREMENT
2	**date**	date			Yes	*NULL*		
3	**invoice_id**	int(20)			Yes	*NULL*		
4	**particulars**	varchar(50)	latin1_swedish_ci		Yes	*NULL*		
5	**medicine_presentation_id** 🔑	int(20)			Yes	*NULL*		
6	**medicine_presentation**	varchar(50)	latin1_swedish_ci		Yes	*NULL*		
7	**medicine_name_id** 🔑	int(20)			Yes	*NULL*		
8	**medicine_name**	varchar(50)	latin1_swedish_ci		Yes	*NULL*		
9	**generic_id** 🔑	int(20)			Yes	*NULL*		
10	**generic_name**	varchar(50)	latin1_swedish_ci		Yes	*NULL*		
11	**supplier_id** 🔑	int(20)			Yes	*NULL*		
12	**supplier_name**	varchar(50)	latin1_swedish_ci		Yes	*NULL*		
13	**unit_price**	float			Yes	*NULL*		
14	**qty**	float			Yes	*NULL*		
15	**purchase_price**	float			Yes	*NULL*		
16	**unit_sales_price**	float			Yes	*NULL*		
17	**unit**	varchar(11)	latin1_swedish_ci		No	*None*		
18	**purchase_paid**	float			Yes	*NULL*		
19	**purchase_due**	float			Yes	*NULL*		
20	**expiredate**	date			Yes	*NULL*		

Figure 7.8 *insert purchase info Table Structure*

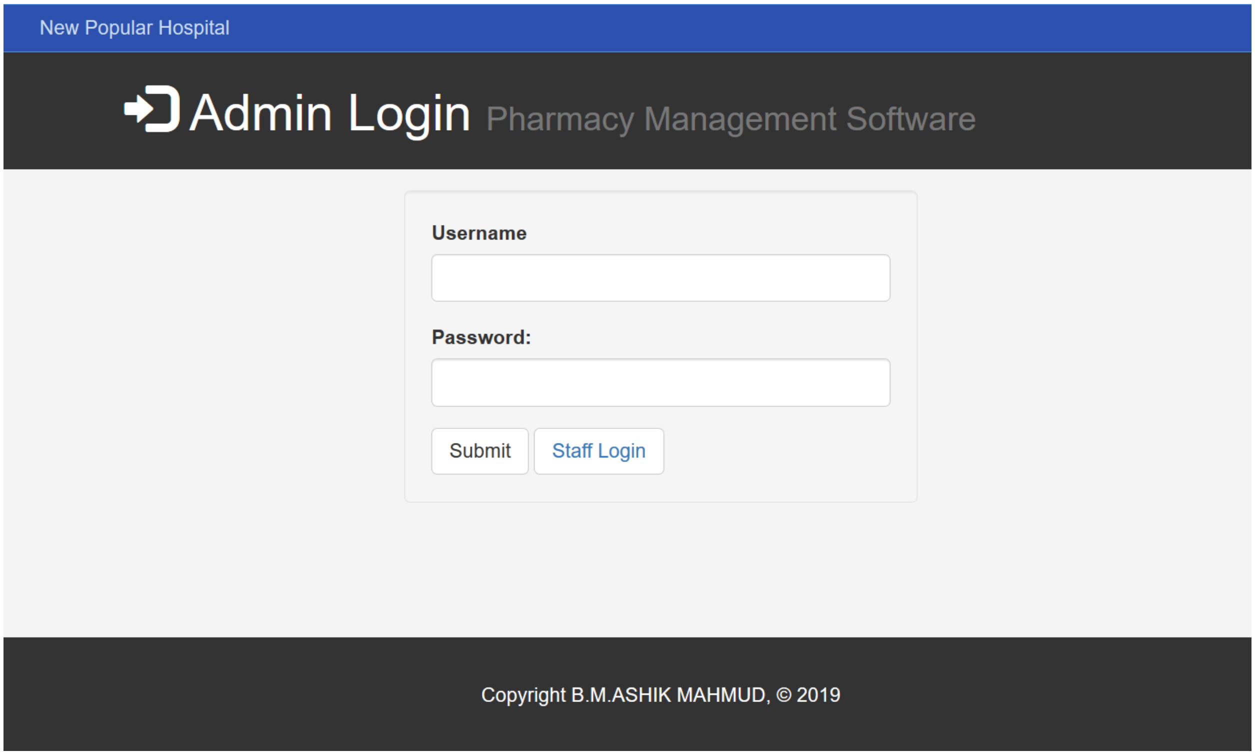
sales_product



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**sales_id** 🔑	int(20)			No	*None*		AUTO_INCREMENT
2	**date**	date			Yes	*NULL*		
3	**invoice**	int(20)			Yes	*NULL*		
4	**particular**	varchar(50)	latin1_swedish_ci		Yes	*NULL*		
5	**customer_email**	varchar(30)	latin1_swedish_ci		No	*None*		
6	**medicine_presentation_id** 🔑	int(20)			Yes	*NULL*		
7	**medicine_presentation**	varchar(50)	latin1_swedish_ci		Yes	*NULL*		
8	**medicine_name_id** 🔑	int(20)			Yes	*NULL*		
9	**medicine_name**	varchar(50)	latin1_swedish_ci		Yes	*NULL*		
10	**generic_id** 🔑	int(20)			Yes	*NULL*		
11	**generic_name**	varchar(50)	latin1_swedish_ci		Yes	*NULL*		
12	**qty**	int(20)			Yes	*NULL*		
13	**unit_sales_price**	float			Yes	*NULL*		
14	**total_price**	float			Yes	*NULL*		
15	**total_amount**	float			Yes	*NULL*		
16	**total_discount**	float			Yes	*NULL*		
17	**discount_price**	float			Yes	*NULL*		
18	**sales_paid**	float			Yes	*NULL*		
19	**sales_due**	float			Yes	*NULL*		

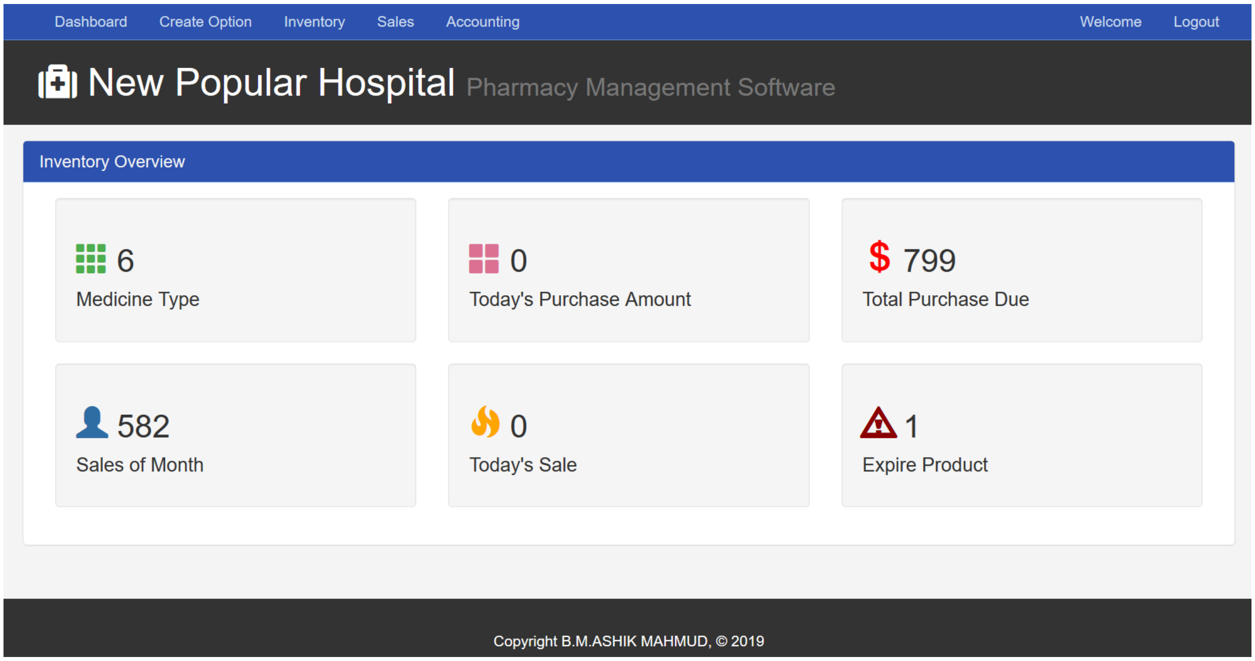
Figure 7.9 *sales product Table Structure*

7.2 Interface Design: Admin



Copyright B.M.ASHIK MAHMUD, © 2019

Figure 7.10 *Interface design of Admin Login*



Dashboard Create Option Inventory Sales Accounting Welcome Logout

New Popular Hospital Pharmacy Management Software

Inventory Overview

6
Medicine Type

0
Today's Purchase Amount

$ 799
Total Purchase Due

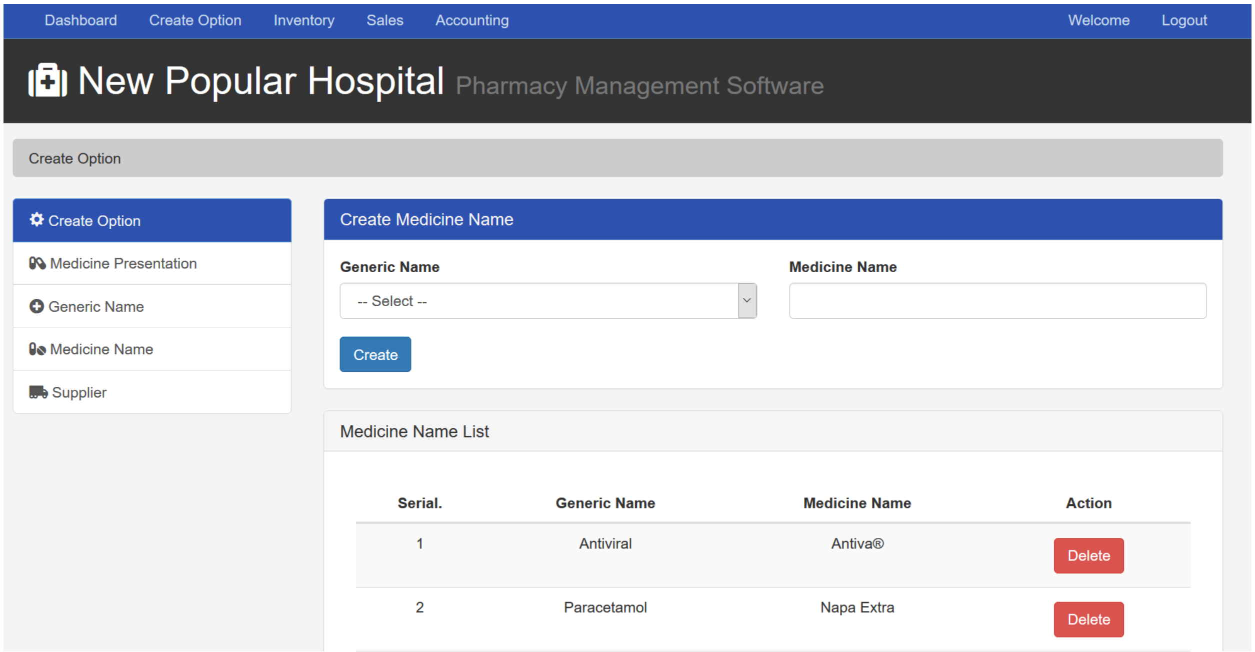
582
Sales of Month

0
Today's Sale

1
Expire Product

Copyright B.M.ASHIK MAHMUD, © 2019

Figure 7.11 Interface design of Admin Dashboard



Dashboard Create Option Inventory Sales Accounting Welcome Logout

New Popular Hospital Pharmacy Management Software

Create Option

- Create Option
- Medicine Presentation
- Generic Name
- Medicine Name
- Supplier

Create Medicine Name

Generic Name: -- Select --

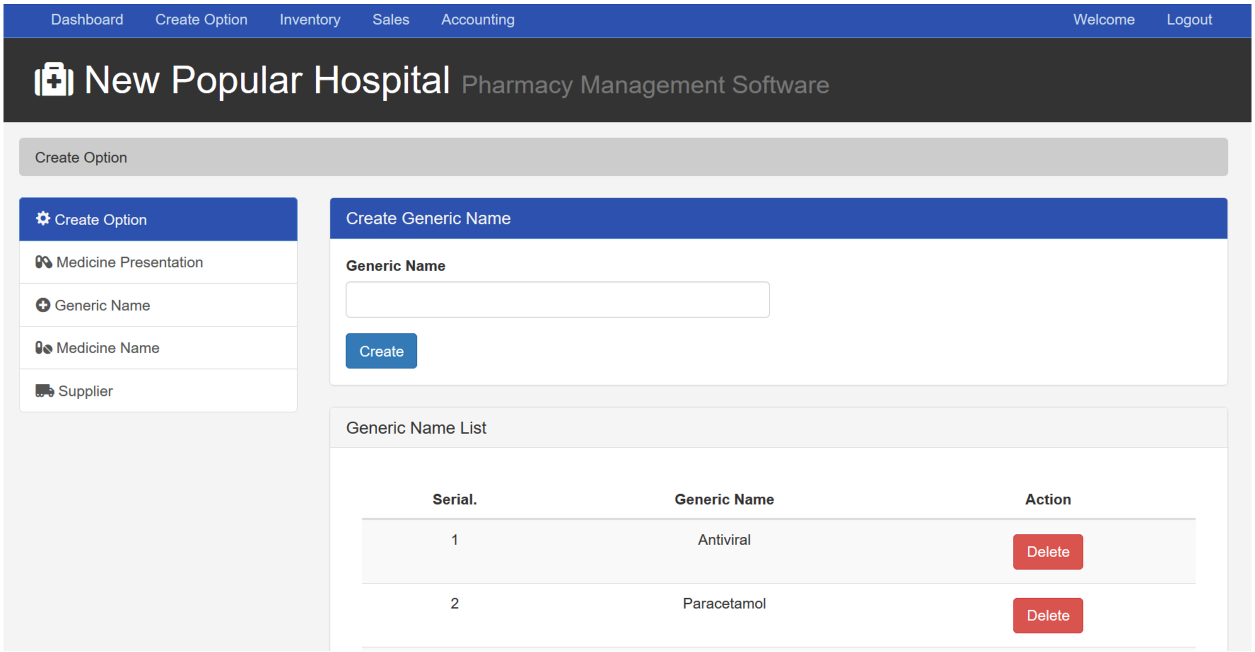
Medicine Name:

Create

Medicine Name List

Serial.	Generic Name	Medicine Name	Action
1	Antiviral	Antiva®	Delete
2	Paracetamol	Napa Extra	Delete

Figure 7.12 Interface design of Create Medicine name



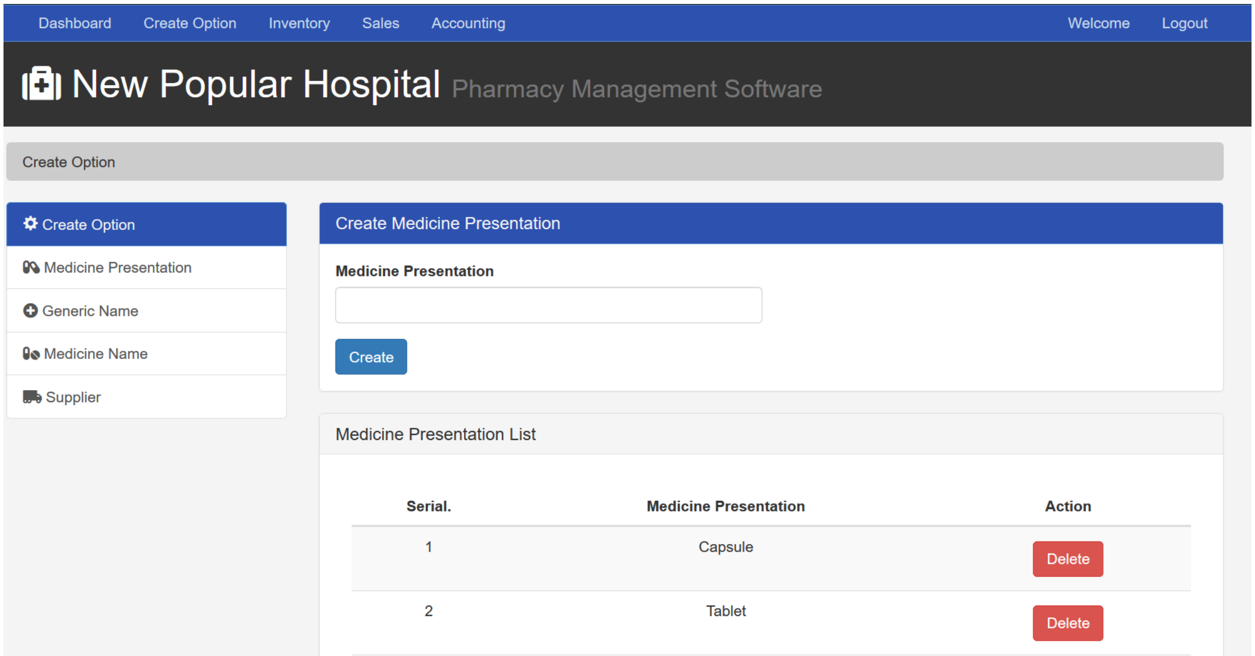
Create Generic Name

Generic Name

Generic Name List

Serial.	Generic Name	Action
1	Antiviral	Delete
2	Paracetamol	Delete

Figure 7.13 *Interface design of Create Generic name*



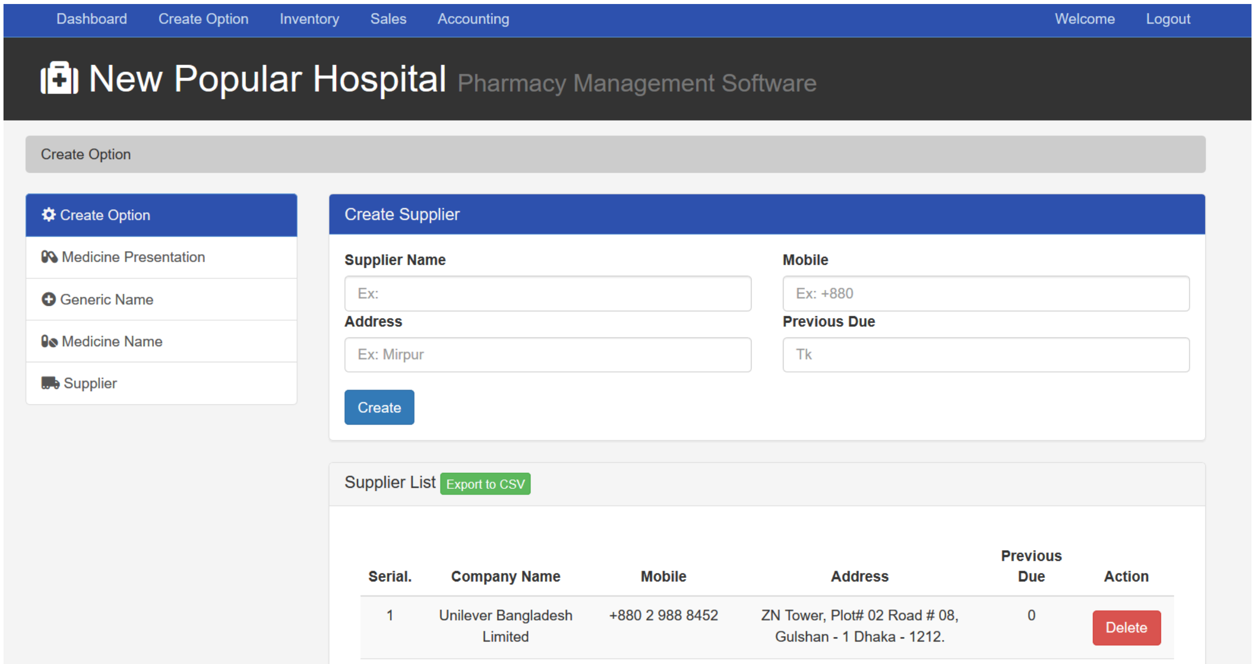
Create Medicine Presentation

Medicine Presentation

Medicine Presentation List

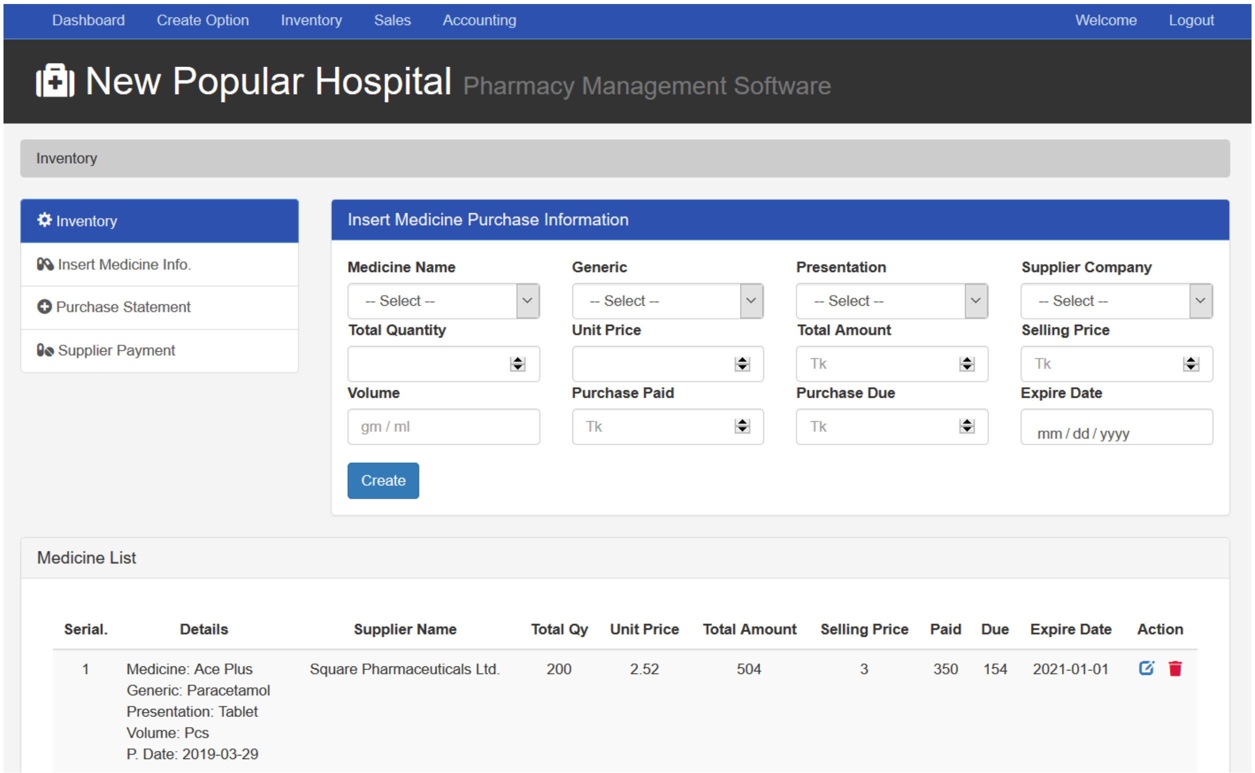
Serial.	Medicine Presentation	Action
1	Capsule	Delete
2	Tablet	Delete

Figure 7.14 *Interface design of Create Medicine Presentation*



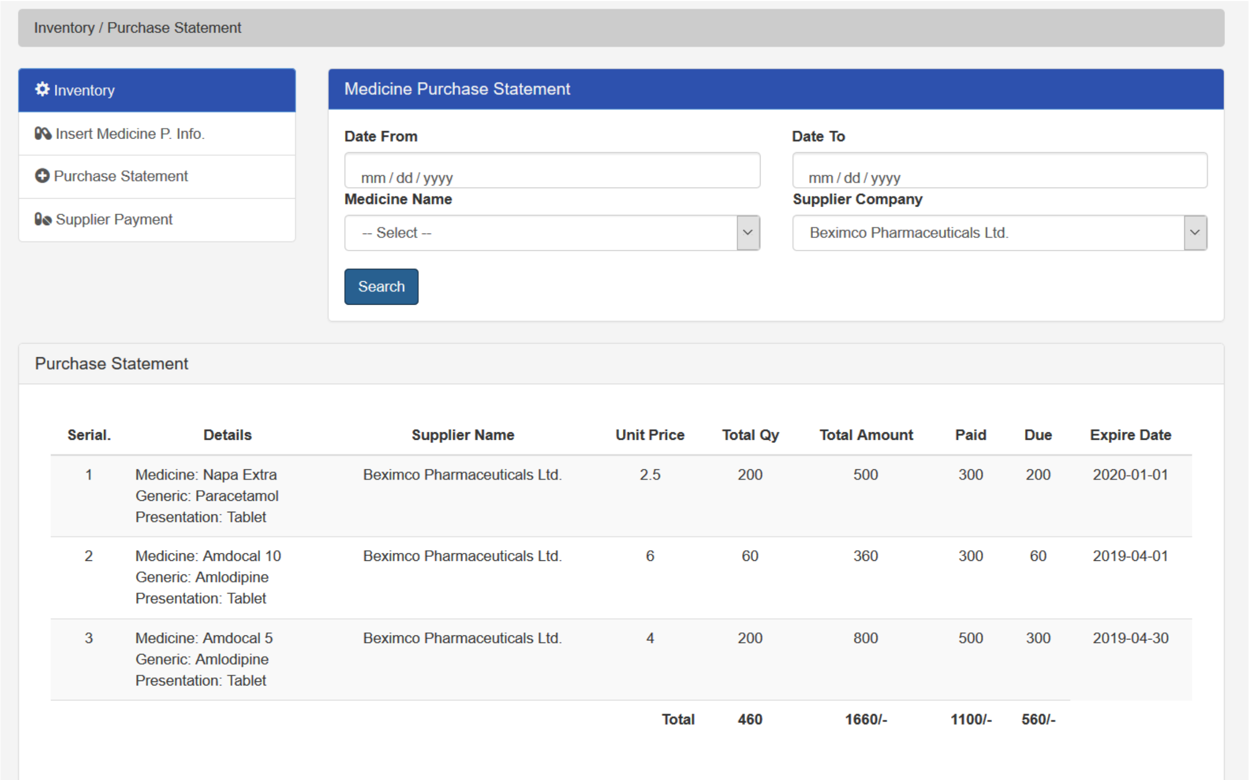
Serial.	Company Name	Mobile	Address	Previous Due	Action
1	Unilever Bangladesh Limited	+880 2 988 8452	ZN Tower, Plot# 02 Road # 08, Gulshan - 1 Dhaka - 1212.	0	Delete

Figure 7.15 *Interface design of Create Supplier*



Serial.	Details	Supplier Name	Total Qy	Unit Price	Total Amount	Selling Price	Paid	Due	Expire Date	Action
1	Medicine: Ace Plus Generic: Paracetamol Presentation: Tablet Volume: Pcs P. Date: 2019-03-29	Square Pharmaceuticals Ltd.	200	2.52	504	3	350	154	2021-01-01	✎ 🗑

Figure 7.16 *Interface design of Purchase Medicine*



Date From: mm / dd / yyyy

Date To: mm / dd / yyyy

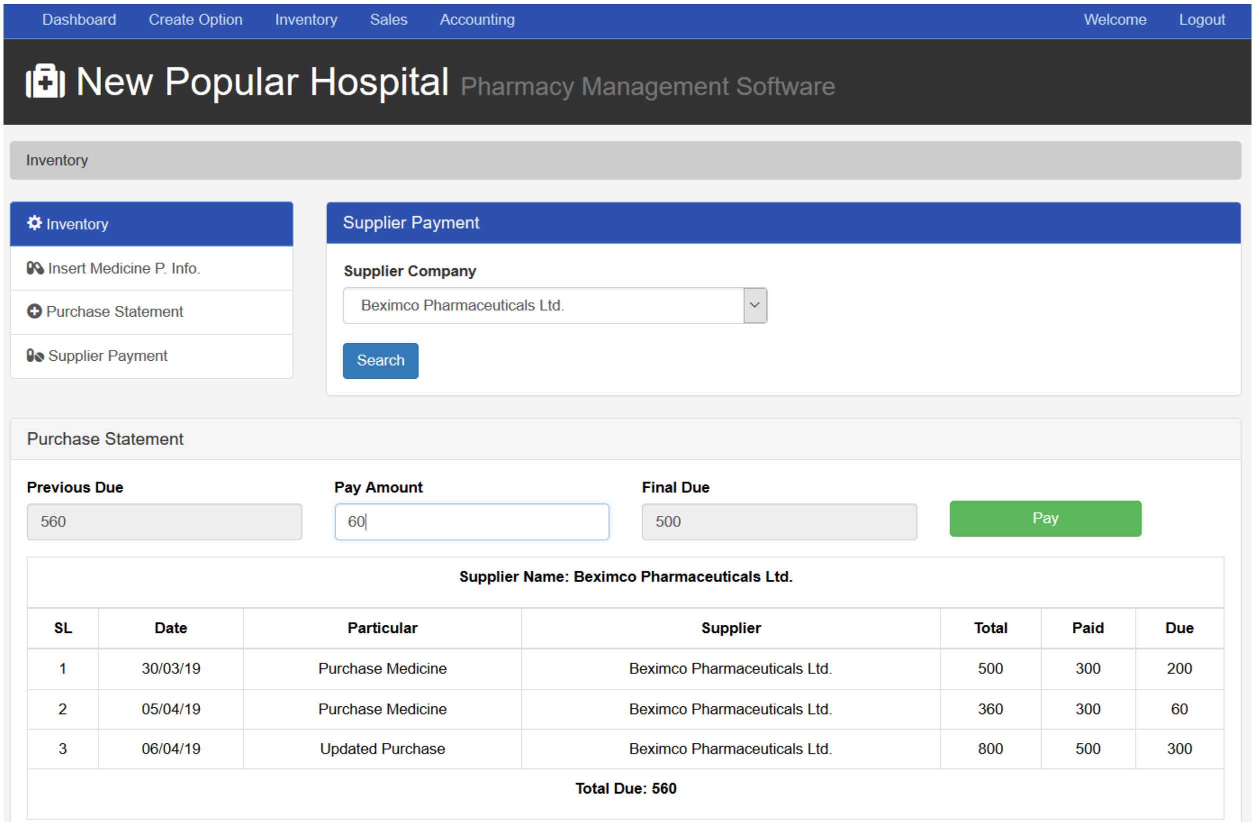
Medicine Name: -- Select --

Supplier Company: Beximco Pharmaceuticals Ltd.

Purchase Statement

Serial.	Details	Supplier Name	Unit Price	Total Qy	Total Amount	Paid	Due	Expire Date
1	Medicine: Napa Extra Generic: Paracetamol Presentation: Tablet	Beximco Pharmaceuticals Ltd.	2.5	200	500	300	200	2020-01-01
2	Medicine: Amdocal 10 Generic: Amlodipine Presentation: Tablet	Beximco Pharmaceuticals Ltd.	6	60	360	300	60	2019-04-01
3	Medicine: Amdocal 5 Generic: Amlodipine Presentation: Tablet	Beximco Pharmaceuticals Ltd.	4	200	800	500	300	2019-04-30
			Total	**460**	**1660/-**	**1100/-**	**560/-**	

Figure 7.17 Interface design of Purchase Medicine statement



Supplier Company: Beximco Pharmaceuticals Ltd.

Purchase Statement

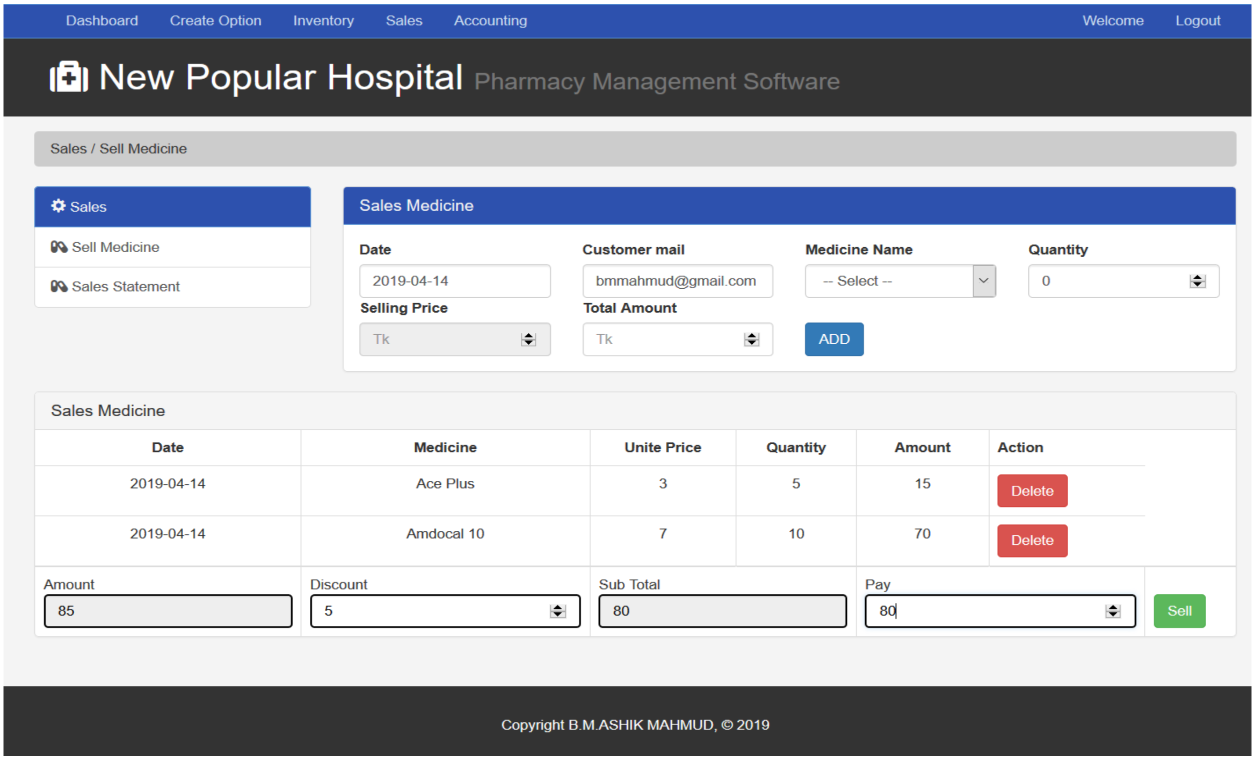
Previous Due	Pay Amount	Final Due
560	60	500

Supplier Name: Beximco Pharmaceuticals Ltd.

SL	Date	Particular	Supplier	Total	Paid	Due
1	30/03/19	Purchase Medicine	Beximco Pharmaceuticals Ltd.	500	300	200
2	05/04/19	Purchase Medicine	Beximco Pharmaceuticals Ltd.	360	300	60
3	06/04/19	Updated Purchase	Beximco Pharmaceuticals Ltd.	800	500	300

Total Due: 560

Figure 7.18 *Interface design of Supplier payment*



Sales Medicine

Date: 2019-04-14
Customer mail: bmmahmud@gmail.com
Medicine Name: -- Select --
Quantity: 0
Selling Price: Tk
Total Amount: Tk

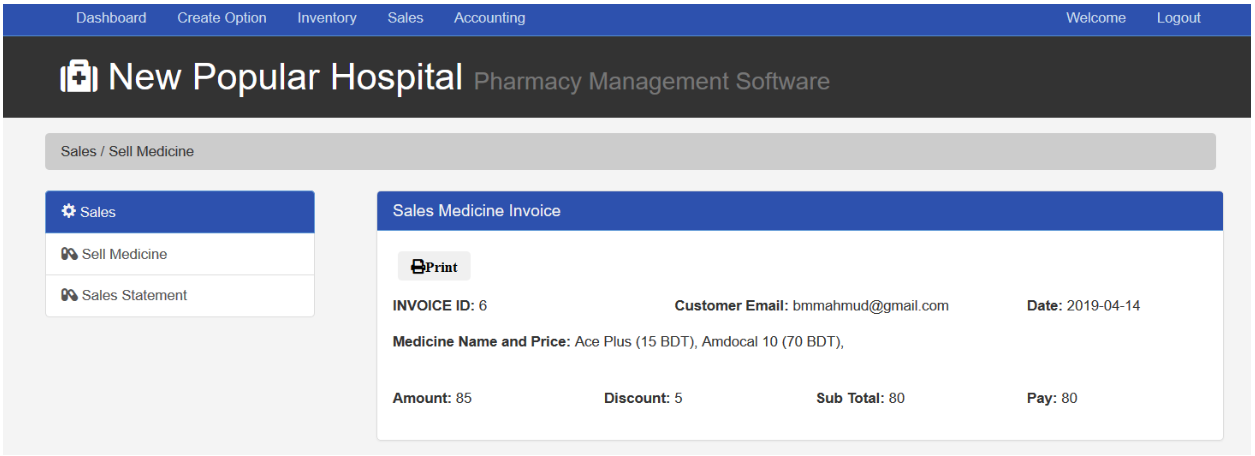
Sales Medicine

Date	Medicine	Unite Price	Quantity	Amount	Action
2019-04-14	Ace Plus	3	5	15	Delete
2019-04-14	Amdocal 10	7	10	70	Delete

Amount: 85
Discount: 5
Sub Total: 80
Pay: 80

Copyright B.M.ASHIK MAHMUD, © 2019

Figure 7.19 Interface design of Sales Medicine



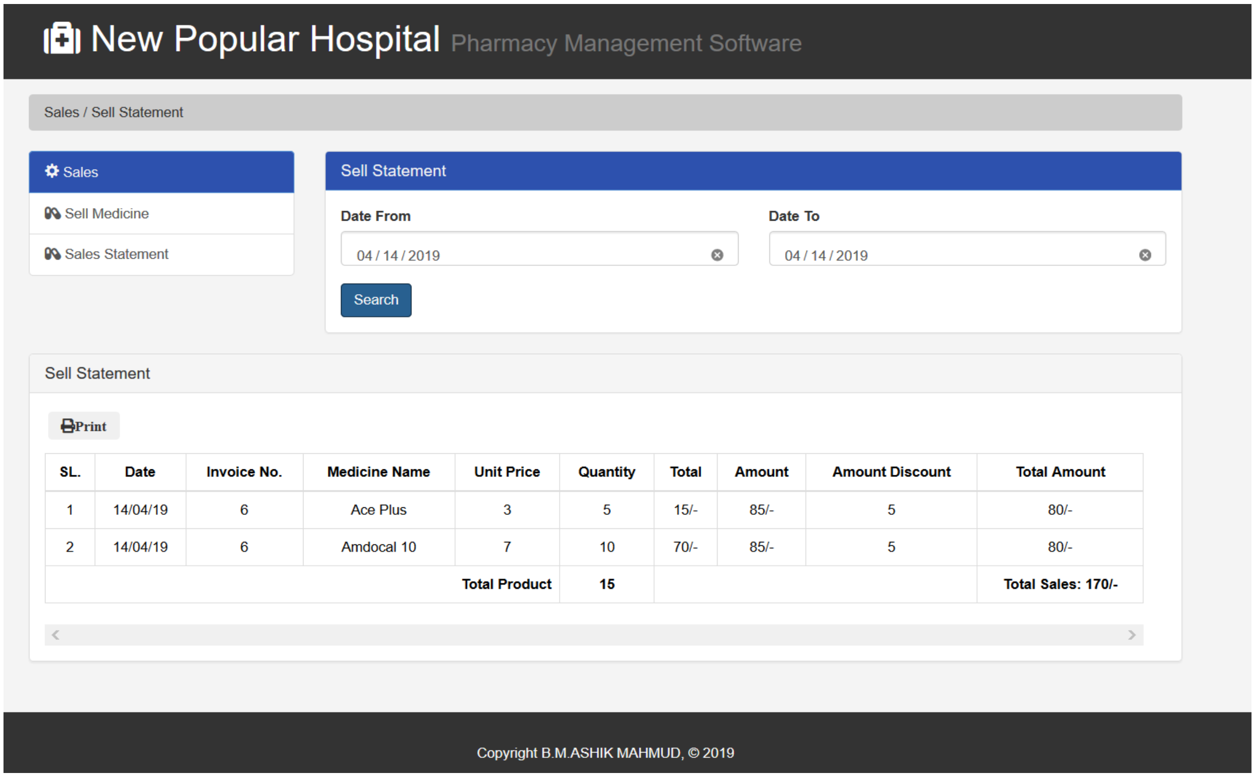
Sales Medicine Invoice

INVOICE ID: 6 **Customer Email:** bmmahmud@gmail.com **Date:** 2019-04-14

Medicine Name and Price: Ace Plus (15 BDT), Amdocal 10 (70 BDT),

Amount: 85 **Discount:** 5 **Sub Total:** 80 **Pay:** 80

Figure 7.20 Interface design of Sales Invoice preview


New Popular Hospital Pharmacy Management Software
Sales / Sell Statement
Sales
Sell Medicine
Sales Statement
Sell Statement
Search
Print
Copyright B.M.ASHIK MAHMUD, © 2019

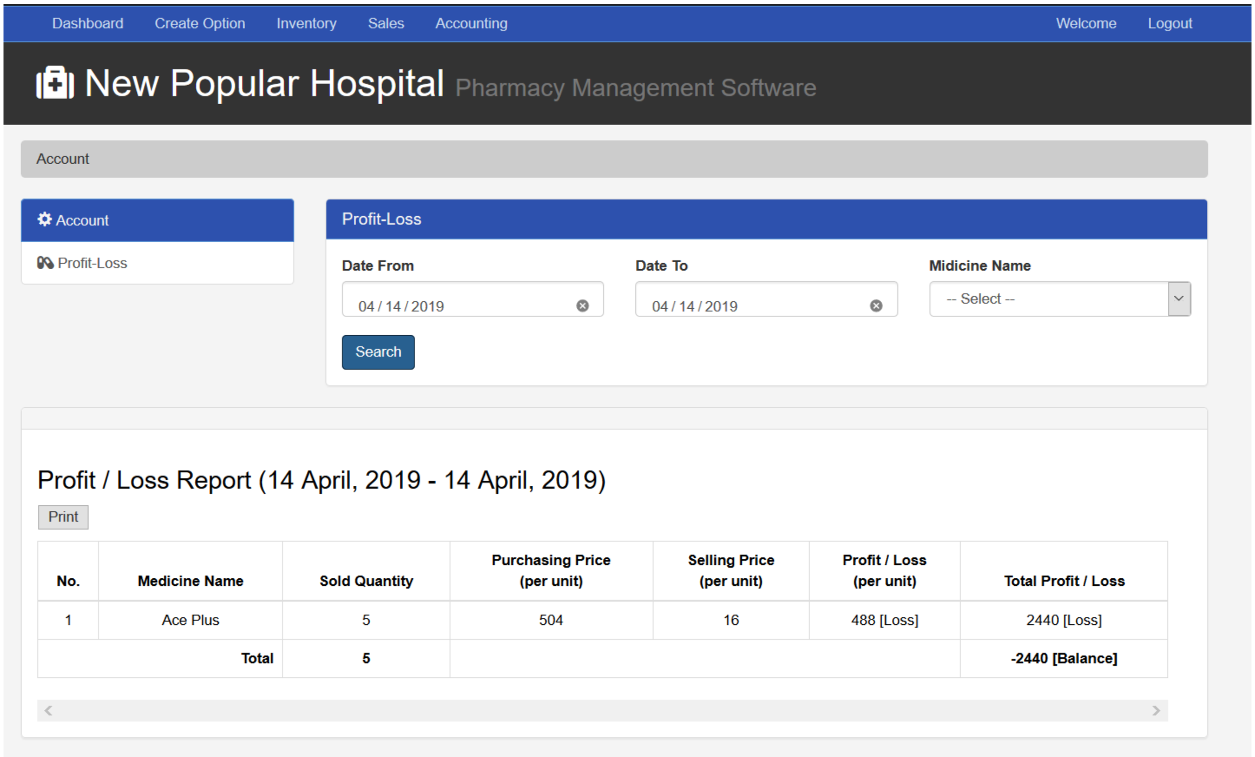
Date From: 04 / 14 / 2019

Date To: 04 / 14 / 2019

Sell Statement

SL.	Date	Invoice No.	Medicine Name	Unit Price	Quantity	Total	Amount	Amount Discount	Total Amount
1	14/04/19	6	Ace Plus	3	5	15/-	85/-	5	80/-
2	14/04/19	6	Amdocal 10	7	10	70/-	85/-	5	80/-
				Total Product	**15**				**Total Sales: 170/-**

Figure 7.21 *Interface design of Sales medicine statement*



Date From: 04 / 14 / 2019

Date To: 04 / 14 / 2019

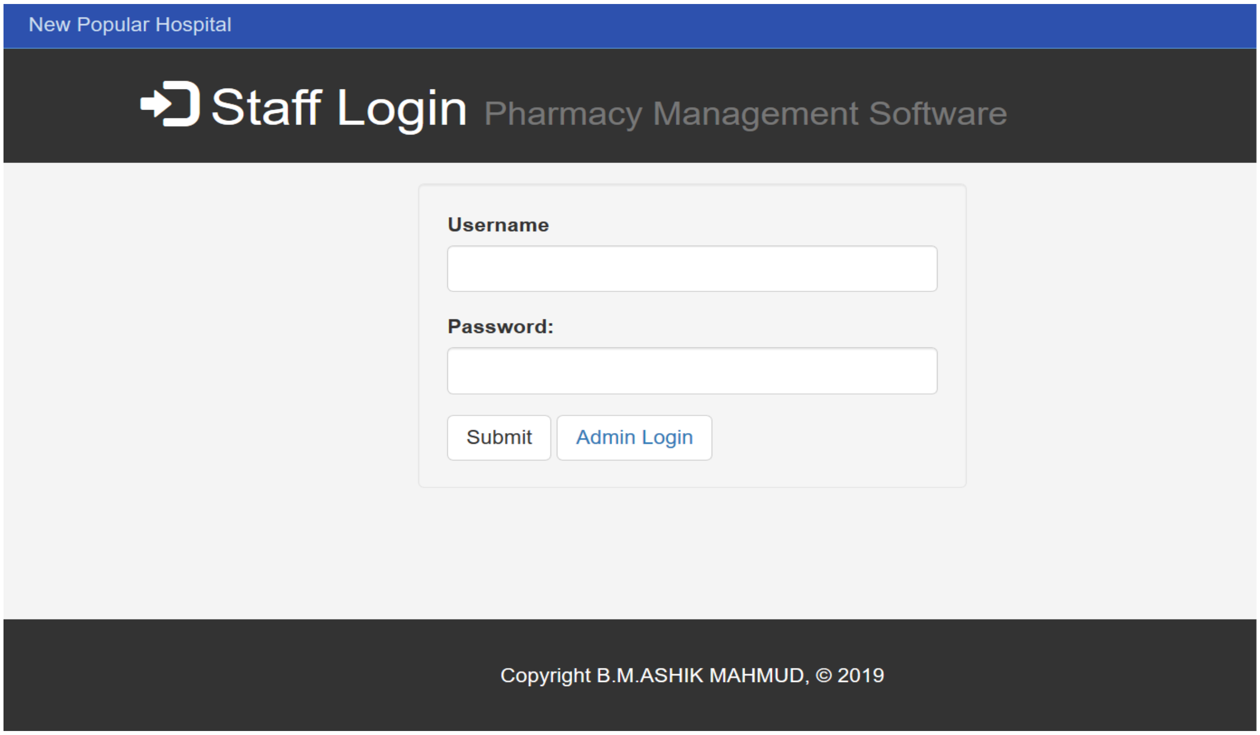
Midicine Name: -- Select --

Profit / Loss Report (14 April, 2019 - 14 April, 2019)

No.	Medicine Name	Sold Quantity	Purchasing Price (per unit)	Selling Price (per unit)	Profit / Loss (per unit)	Total Profit / Loss
1	Ace Plus	5	504	16	488 [Loss]	2440 [Loss]
	Total	**5**				**-2440 [Balance]**

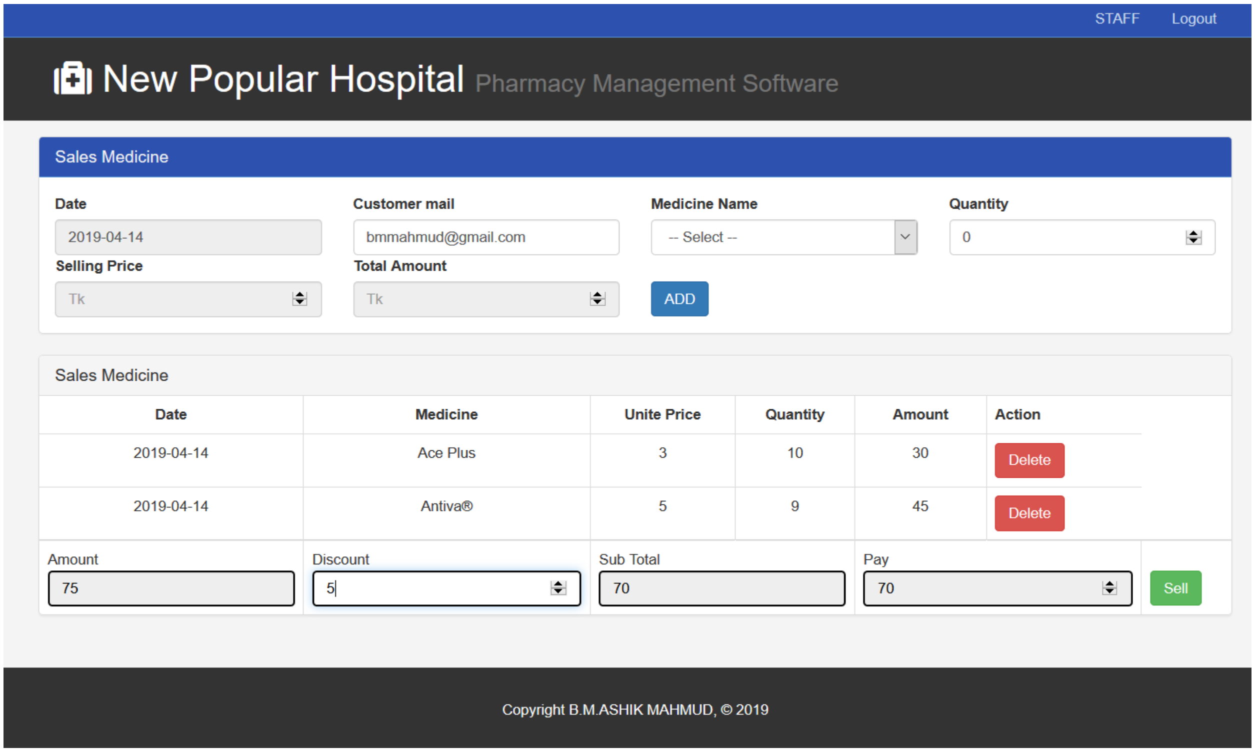
Figure 7.22 *Interface design of Account*

7.3 Interface Design: Stuff



Copyright B.M.ASHIK MAHMUD, © 2019

Figure 7.23 Interface design of Staff login



Date	Customer mail	Medicine Name	Quantity
2019-04-14	bmmahmud@gmail.com	-- Select --	0
Selling Price	Total Amount		
Tk	Tk	ADD	

Sales Medicine

Date	Medicine	Unite Price	Quantity	Amount	Action
2019-04-14	Ace Plus	3	10	30	Delete
2019-04-14	Antiva®	5	9	45	Delete

Amount	Discount	Sub Total	Pay	
75	5	70	70	Sell

Copyright B.M.ASHIK MAHMUD, © 2019

Figure 7.24 Interface design of Sales Medicine

Chapter 8 | Quality Assurance

8.1 System testing

Software testing is the process of evaluation a software item to detect differences between given input and expected output. Also, to assess the feature of A software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words, software testing is a verification and validation process.

Verification: Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

Validation: Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

The objectives of software testing are:

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has a high probability of finding an as-yet-undiscovered error.
- A successful test is one that uncovers an as-yet-undiscovered error.

The design of tests for software can be challenging as the initial design of the product itself. Software can be tested in one of two ways:

- Knowing the specified function that the software has been designed to perform, tests can be conducted that demonstrate each function fully while at the same time searching for errors in each function. This approach is known as black-box testing.
- Knowing the internal workings of software, tests can be conducted to ensure that internal operations are performed according to specifications and all internal components have been adequately exercised. This approach is known as white-box testing.

8.1.1 Software Testing Strategy

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of a software. The strategy provides a road map that describes the steps to be conducted as part of testing.

Testing strategy that will be followed in this software project –

- Unit testing
- Integration testing
- Validation testing

The first step in software testing is unit testing. Unit testing concentrates on each unit of the software as implemented in source code. Unit testing focuses on each component individually. The unit test is white-box oriented. Thus, unit testing of this library software will be done after completion of every module or component. The next step is integration testing. Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective of integration testing is to take unit tested components and build a program structure that has been dictated by design.The integration testing strategy that has been chosen for this project is top down testing.Black-box testing method is the most prevalent for integration testing. Top down integration strategy will be used to perform integration testing. Top down integration will be done by breadth-first manner. Breadth-first integration incorporates all components directly subordinate at each level, moving across the structure horizontally. After the software has been integrated, a set of high order tests am conducted. Hence, the validation criteria that have been mentioned in requirements engineering should be tested. Validation testing provides final assurance that software meets all functional, behavioral and performance requirements. The black-box testing method is exclusively used in validation.

8.2 System Testing Methodology

- **Black-box Testing**

Black-box testing which is also known as behavioral testing focuses on the functional requirements of the software. It enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. Black-box testing method will be applied to test the modules of LMS.

Figure 8.1 *White-box Testing*

- **White-box Testing**

White-box testing, which also known as glass-box testing, is a test case design method that uses the control structure of the procedural design to derived test cases. Using white-box testing methods, software engineer can derive test cases that,

1. Guarantee that all independent paths within a module have been exercised at least once
2. Exercise all logical decisions on their true and false sides
3. Execute all loops at their boundaries and within their operational bounds
4. Exercise internal data structures to ensure their validity.

The modules that contain some complex calculations or decision making code such as check the availability of the library item will be tested using white-box method.

8.3 Testing Design

Testing scenario No: 1	
Scenario	Login testing scenario of system
Input’s	Username, password of admin for Login
Desired Outputs	When enter Username, password then get access level define.
Actual Outputs	For login my system works correctly.
Verdict	Getting result from Desired Outputs and Actual Outputs decided this system is successful for login.

Testing scenario No: 2	
Scenario	Login testing scenario of system
Input’s	Username, password of Staff admin for Login
Desired Outputs	When enter Username, password then get access level define.
Actual Output	For login my system works correctly.
Verdict	Getting result from Desired Outputs and Actual Outputs decided this system is successful for login.

Testing scenario No: 3	
Scenario	Creating Medicine Presentation testing scenario of system
Input’s	Admin insert medicine presentation.
Desired Outputs	Insert into database and show in a table
Actual Outputs	Successfully inserted and showed in table.
Verdict	Getting result from Desired Outputs and Actual Output’s decided this system is successful for creating medicine presentation.

Testing scenario No: 4	
Scenario	Creating Generic testing scenario of system
Input’s	Admin insert medicine generic.
Desired Outputs	Insert into database and show in a table
Actual Outputs	Successfully inserted and showed in table.
Verdict	Getting result from Desired Outputs and Actual Outputs decided this system is successful for creating medicine generic.

Testing scenario No: 5	
Scenario	Creating medicine testing scenario of system
Input’s	Admin insert medicine generic and medicine name.
Desired Outputs	Insert into database and show in a table.
Actual Outputs	Successfully inserted and showed in table.
Verdict	Getting result from Desired Outputs and Actual Outputs decided this system is successful for creating medicine generic and medicine name.

Testing scenario No: 6	
Scenario	Creating Supplier Information testing scenario of system
Input’s	Admin insert medicine generic and medicine name.
Desired Outputs	Insert into database and show in a table, also export CSV.
Actual Outputs	Successfully inserted, showed in table and CSV export Done.
Verdict	Getting result from Desired Outputs and Actual Output’s decided this system is successful for creating medicine generic and medicine name.

Testing scenario No: 7	
Scenario	Insert Medicine Purchase Info. testing scenario of system
Input’s	Admin insert medicine purchase info. with date, price and medicine info.
Desired Outputs	Insert into database with valid data and show in table.
Actual Outputs	Successfully inserted with valid data and showed in table.
Verdict	Getting result from Desired Outputs and Actual Outputs decided this system is successful for Insert medicine purchase.

Testing scenario No: 8	
Scenario	Update Medicine Purchase Info. testing scenario of system
Input’s	Admin update medicine purchase info. with date, price and medicine info.
Desired Outputs	update into database with valid data and show in table.
Actual Outputs	Successfully updated with valid data and showed in table & row.
Verdict	Getting result from Desired Outputs and Actual Outputs decided this system is successful for Update medicine purchase.

Testing scenario No: 9	
Scenario	Sales Medicine and print testing scenario of system
Input’s	Admin sales multiple medicine and print Invoice.
Desired Outputs	Add multiple medicine and insert into database with valid data and print.
Actual Outputs	Successfully added medicine, insert into database and showed in a invoice, also print.
Verdict	Getting result from Desired Outputs and Actual Outputs decided this system is successful for sales and print invoice of medicine.

Testing scenario No: 10	
Scenario	Sales Medicine and print testing scenario of system
Input’s	Staff sales multiple medicine and print Invoice.
Desired Outputs	Add multiple medicine and insert into database with valid data and print.
Actual Outputs	Successfully added medicine, insert into database and showed in invoice, also can print.

Verdict	Getting result from Desired Outputs and Actual Outputs decided this system is successful for sales and print invoice of medicine.

Chapter 9 | Conclusion

9.1 Preface

Today is the age of modern science and information and online communication, which is critical to development of more effective operational and management process. To provide better and uninterrupted services to the employee of Recursion Technologies Ltd a group of Software specialist working together to keep the service all time. I was fortunate and blessed to get this lucky break to work some of these efficient hard working friendly engineers. My earnest thanks, gratitude and salutations to these wonderful people from the deep down inside my heart.

9.1.1 Practicum and Its Value

In your career development as with most life issues there is direct relationship between effort and reward. To me, practicum can be as a transition from engineering college study life to a real world workplace through hands on experience of engineering practices.

The four years of undergraduate engineering studies gives a student theoretical and practical knowledge. Using that knowledge and observing live operational system, the practicum program clarifies those subjects‘ matters to another level blessed with practical working skills. Considering this fact, it gives us an immense pleasure to say that my practicum was a successful event.

Practical work experience doesn‘t have any other alternatives. Before getting into the job student should have a real world work experiences in a major field of study. Now a day‘s recruiter no longer considers just high grades, good communication skill, part time work experiences. They highly consider the work experiences of an applicant. Students with better work experiences are getting the better job opportunities.

Recursion Technologies Ltd gives us the opportunity of working in a professional working environment. During the internship period I have tried my level best to make my system efficient. I followed the lessons, methods, tools and techniques that I have learned during my study period at IUBAT. Successful software development is a blend of standard development practices, proper theoretical knowledge and the developer‘s creativity.

Student of College of Engineering and Technology (CEAT) at IUBAT go for this practicum program carrying 9 credit hours weight, which goes for a semester long and usually after the completion of the course work. A report submitted after the completion of the practicum followed by a presentation and a comprehensive examination on the overall four years education.

9.2 Future Plan

- Single page Interface
- Page Load free process
- Sales statement with graph presentation
- Normalize database
- Develop the whole coding process with SOLID principal.
- Single login system for both admin and staff.

9.3 Limitation

- Separate login system for admin and staff
- Email system
- Bar code scan to get data from product

9.4 Conclusion:

The biggest experience working at Green Software and Technology is indeed being a part of designing and implementing software. my most experience was JavaScript error finding . I have learnt a lot of new things which was so much unknown to me. I have also learnt some technical issues which help to do better in future life. The following indicator will indicate some of my technical issue which i have learnt and implemented from this project.

- The designing strategy of a web based project.
- The analyzing strategy of a web based project.
- New environment of programming languages.
- Developing a new project using the existing project.
- How to coding MY SQL and designing by PHP with framework, HTML and CSS.

All this things will help me to develop my future carrier.

REFERENCES:

1. Requirement Engineering Assurance Available, Retrieved February 02, 2019,From https://www.tutorialspoint.com/software_engineering/software_requirements.htm

2. Longstreet, David. (2005). Fundamentals of Function Point Analysis. Retrieved

April 2, 2019 ,From http://www.softwfaremetrics.com/fpafund.html.

3. How to calculate function points .Retrieved 2 April, 2019, From http://stackoverflow.com/questions/34473698/how-to-calculate-functionpoints

4.System quality assurance. Retrieved April 15,2019 , From https://en.wikipedia.org/wiki/Software_quality_assurance

5. Testing method. Retrieved 03 April, 2019, From http://www.test-institute.org/Software_Testing_Methods.php