Kabul University

Computer Science Faculty

Department of Software Engineering

Pharmacy Management System

Bachelor’s Thesis

In Computer Science

Authors:

- Dawood Adib
- Rohullah Asadi

Supervisors:

- Asst. Prof. Maryam Farda
- Asst. Prof. Mohammad Rafi Bahez

2016

Acknowledgement

From the final project, I would like to be appreciating Prof. Mohammad Rafi Bahez and Prof. Maryam Farda for contributing their ideas and in-depth knowledge in the field. A warm thanks is extended to them for sharing their resources, opinion, knowledge, and experience and skills in programming and development methodology, so generously.

I would also like to personally thanks to my partner, Rohullah Asadi. When I am doing the final year project, he offers a lot of information to me in order to finished assignment. He always said any difficult subject is possible to fulfill. I would like to personally appreciate to her.

After that, I want to thank to our parents for their evermore support and encouragement. Then, I would to thanks to all my friends and classmates who have provided some opinion. I can successfully complete the project. It is very high achievement.

Abstract

A system in Pharmacy Management System is the important component for public Pharmacy of Kabul population because a system can provide access to many of the information that exists in Pharmacy, which is necessary component of almost any research experience today. The purpose of this system is to computerize and replace the current paper-based system and Administrator can register medicine add, edit, delete and view the information about that. Pharmacy uses the manual system to manage the medicine stock. It needs the pharmacist assistant check the medicine twice a week to check expire date of the medicine in the storage and the medicine that out of stock. The pharmacist assistant take out the medicine that rise the expired date and keep it at the safety place to avoid mistake the expired medicine to be sell. If there is the out of stock of the medicine, the purchase order form is filling by the pharmacist to order the medicine from the medicine manufacturer. The manual system that use is not very reliable to the pharmacy management today. Therefore, the new approach of system application is needs to make the pharmacy management more efficient and effective. Within PMS application, the medicine stock that is rise at the expired date will been detected and the pharmacist assistants no need to check medicine twice a week anymore. They just login to the system and check it at the list of the medicine that display by the system. For purchasing the order, pharmacist can generate it automatically from the system when the medicine stock is out of stock.

Table of Contents

1. CHAPTER I

INTRODUCTION

1.1 Project Background

Sistan Pharma Company is a major company in trade of medicine and healthcare equipment provider in Afghanistan. Sistan Pharma co. buys medicine from abroad manufacturers, import medicine, store them in the stock, and distribute/sell medicine to sub-branches and local drugstores. Pharmacy Management System (PMS) is the process of managing all processes and tasks related to Sistan Pharma co. including: buying medicine, import to stock, stock management, and selling medicine to local branches and customer drugstores.

The domain of this project is Information Communication Technology in trade of medicine industry and Healthcare. Within the growth of the Information Communication Technology and Medical Technology, the system developer takes this opportunity to help the pharmacy business for managing processes and jobs in the field of medicine trading.

The organization that relates to Pharmacy Management system (PMS) is the Sistan Pharma Company and all same businesses in the field of medicine trade. PMS is developed to register a list of medicine Manufacturers Company that Sistan Pharma buy medicine from them and record details of inventory imported to the stock and manage medicines in the stock. The system notify the manager near expiration medicines and inventory out-of-stock. This system also does the selling process, list of customers and sub-branches are registered, when medicine sold to the customer, system reduce it form stock inventory. Additionally, PMS provide the report of the medicine that is sold. The modules involved in PMS are medicine buy and import module, medicine stock module, selling process module, medicine notification module, medicine bill module, user authentication module, user registration module, statistic of medicine sales module, monthly report module, employee registration module, employee salary payment module, daily payment module, and report of medicine module.

1.2 Problem Statements

First, process of importing medicine, storing in the stock, and selling takes long time. Therefore, the time is wasted is wasted for doing all of this process separately. Second, for the medicines stock management, the pharmacy manager must check it manually and no warning message is given for the medicine expiration and decrease of medicine to finish. So, PMS provide the advantage to setup the process in managing and notification alert program to warn about the medicine stock condition.

Finally yet important, no analyses are done for the frequency type of medicine usually bought by the customers or sub-branches at that area. This is also important to determine the medicines that are demanded more from the customers so that pharmacist can be prepared to order more for that type of medicine.

1.3 Objective

The objectives that to be achieved from Pharmacy Management System application is:

I. To record in details imported medicine attributes and money.

II. To provide the reliable medicine management system.

III. To provide a secure system application in terms of information retrieval.

IV. To help in decision making.

V. To provide a systematic medicine inventory.

VI. To help in managing staffs and salary payment.

1.4 Scope

The project scope will define the boundaries of PMS, which include PMS functionality, users and operating location/availability.

1.4.1 System Functionality Modules to develop are:

A. medicine buy and import module

List and classifying the medicine relate to the symptoms that occur. The other Function in this module is adding, delete and update medicine from the list and class of the medicine.

B. Medicine stock module

This module will process medicine inventory adding to list when imported it to the stock and reduce from inventory when sold to customers.

C. Selling process module

Record the medicines that is sold to the customer. Create bill that give the quantity and type of medicine and amount of money. Also generates list of medicines that are more frequently bought by the customer.

D. medicine notification module

The notification about medicine near expiration date and inventory out-of-stack alert message.

E. User authentication module

The authentications are dividing to two (2) types, top manager administrator authentication and pharmacy manager administrator authentication. Administrator user can view all the process includes, selling transaction, reports, medicine stock and manipulate list of medicine. Pharmacy manager administrator can only do the process like selling and buying process, view medicine stock, and list the medicine.

F. Employee salary payment module

Function to record the payment of employees' salary and calculation of payments.

G. User Registration module

Handle for the new user registration to be the authorized user in PMS application.

H- Statistic of medicine sale module

Display the statistic for medicine selling in bar graph visualization.

I. Monthly report module

Display the medicine buying statistic to show the total sold for each medicine.

J. Report of medicine module

Provide the report to display the medicines that want to be ordered.

1.4.2 System Users

A. Pharmacy top manager

- Responsible for buying and selling of medicine.

- Responsible to monitor the medicine stock alert warning.

- Pay employee salary.

- Check the analyst of the frequency of sold medicine.

B. Pharmacy stock manager

- Responsible in adding imported medicine.

- Responsible for providing ordered medicine for customer.

- Keep track of medicine expiration date and inventory out-of-stack.

C. Customer

- Order medicine to buy from Pharmacy Company.

- Pay ograyee and remaining debts.

1.4.3 Operating location availability

A. Available to deploy at all type of pharmacy major seller in Afghanistan.

B. A Server computer at main branch.

C. Client computer at pharmacy manager place.

D. Client computer at stock manager place.

1.4.4. Hardware and software requirement

I. Operating system platform: Windows (XP, 7, 8) Professional

II. Browser: Internet Explorer 6.0, Mozilla Firefox, Google Chrome

III. Hardware as required for running windows 7 professional

   a. A minimum hard disk space of 20 Gigabytes (GB)

   b. RAM size of 2GB

   c. Pentium 4 dual processor CPU

1.5 Project Significance

The pharmacy manager can work in the systematic environment with the warning process for medicine expiration and medicine out-of-stock. Process of importing and selling is synchronized with database.

PMS is also a more effective system because it can produce the analysis of the total medicine sold, so that the company can know the customer buying pattern. Because of that, pharmacy manager can be more prepared to increase the order of the medicine that more frequency bought by the customer or sub-braches.

This system can help management of pharmacy to produce the bill information of medicine to the customer. Therefore, customer can be more aware to the details of the medicine that they want to buy.

1.6 Expected Output

Pharmacy Management System is concerned to the delivery of system functionality as first priority to be handled. The functionalities that involve in this system are user authentication, retrieve medicine data, notification of medicine condition, calculation of medicine inventory, and insert medicine data, update data, and delete data should be done successfully.

PMS process the order given by customer. The calculation process quantity of available medicine and notify reduction of medicine in stock which is, the pop up message display when the inventory out-of-stock.

There are the four (4) reports consist of the update report for every process involve. Statistic medicine sales report is the selling report for the certain period of time. Then, monthly report for the medicine that frequently bought by customer. This report is to provide report as reference to set the medicine that is has the priority to order first and many. Finally, report for medicine, which is, to display the level of the quantity for the medicine at the certain time. Employee salary payment report, which shows paid money at a period of time and total amount of remaining salary.

1. 7 Conclusion

Pharmacy Management System application is a reliable and integrated pharmacy management system that help the business company in better management. It provides easy management of pharmacy stock and a notification system that alerts pharmacy manger about expiration date of medicines and inventory out-of-stock. There are six (6) objectives that will be achieved to make the system implementation successful. The objectives are to record in details imported medicine attributes and money. To provide the reliable medicine management system. To provide a secure system application in terms of information retrieval. To help in decision making. To provide a systematic medicine inventory. To help in managing staffs and salary payment.

Scope of this system application is the boundary to guide the system development to reduce the unnecessary process before it occurs. The scopes are divided to four (4) categories. These categories are the limit of the development can achieve. There are, system functionality, users, operating location availability, and hardware and software requirement. Project significance is describing for the advantage of the PMS- system application when it is to be implemented. The expected result of this system is the successful of the all modules can be implemented and all objectives can be achieved.

For the all of the result and solution that achieve in CHAPTER I, Introduction will be delivered to the next chapter, CHAPTER II, Literature Review and Project Methodology.

2. CHAPTER II

LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 Introduction

In this chapter, Literature Review and Project Methodology are the reviewing from the before research that are made by the other people. The reviewing is including the process that relate to the PMS system application. As the result, this chapter can determine the fact and findings of PMS, which are, the domain that this system involve, project methodology that use to develop, project requirement as the tools to develop and the project schedule and milestones.

2.2 Facts and Findings on PMS

2.2.1 Domain

Domain for PMS is Information and Communication Technology, ICT in Commerce and Healthcare. According to Healthcare providers are human and people are not infallible and unfortunately, errors do happen occasionally. Today's technology gives us good solutions to support pharmacy business and prevent a good deal of those unwanted errors.

An outstanding example of an area where ICT can play an important part is the exchange of data in pharmacy trading business and managing business system.

Because of that, PMS provide the selection of medicine to the pharmacist to indicate the quality of the pharmacy management. Then, the alert process is the systematic system that gives many advantages to pharmacy, that is, to prepare the stock medicine and the priority to treat the disease that more frequently occur.

Based on explanations above, the important point that they want to highlight are the technology in ICT can provide the solution to the problem and help to give the systematic process in the healthcare program. This is, use the system that can identify the patient and caregivers unique identification to as the security of them to communicate.

The importance of ICT that can help to manage the current operational to be more effective. Furthermore, there are methods that can be implemented by using ICT, strategic system. Strategic system means that modem ICT infrastructure can handle the producing and achieving of healthcare knowledge to the person that required, which are, patients, clinical professionals, healthcare managers, administrators and policy makers.

2.3 Project Methodology

PMS is use two (2) types of methodology approach to develop the system. The methodologies are Software Development Life Cycle, SDLC to the system development and Database Life Cycle to develop the database.

2.3.1 Software Development Life Cycle, SDLC

System development is organized into phases, with deliverables and milestones to measure progress [1] .

SDLC is the history of an information system. It is the reference by the database design and the application developer. Feasibility study can help in

[1] Shelly, Gary B and Harry J Rosenblatt. *Systems Analysis And Design*. 8th ed. Boston, Mass.: Thomson Course Technology, 2010, p 18.

the initial state in the SDLC. This methodology is divided to five (5) phases, which are, planning, analysis, detailed system design, implementation, and maintenance.

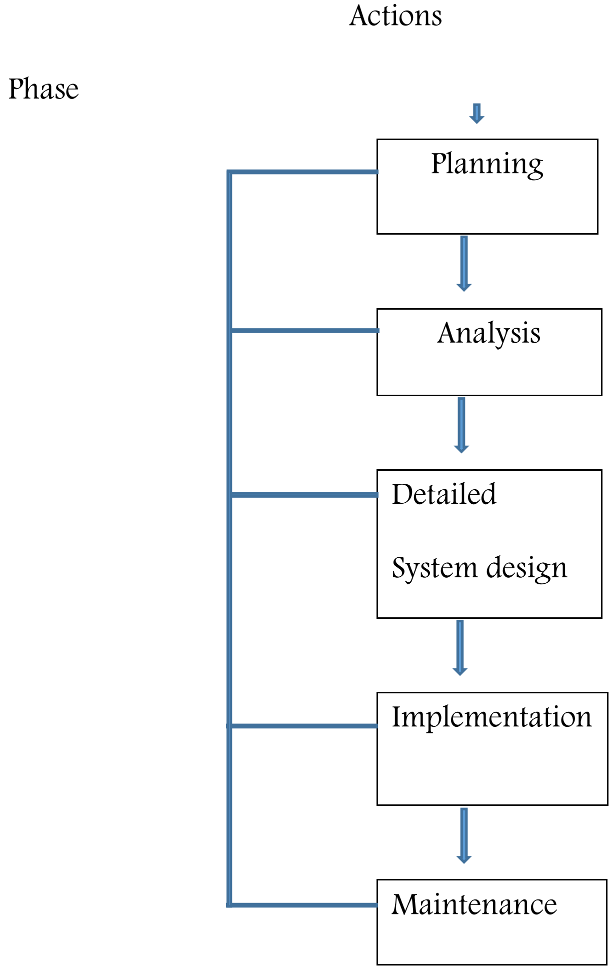


Figure 2.1: The Software Development Life Cycle, SDLC

Figure 2.1 show the software development life cycle that will be implemented to develop the PMS. It is will be describing according to the phases that exists in the software development life cycle.

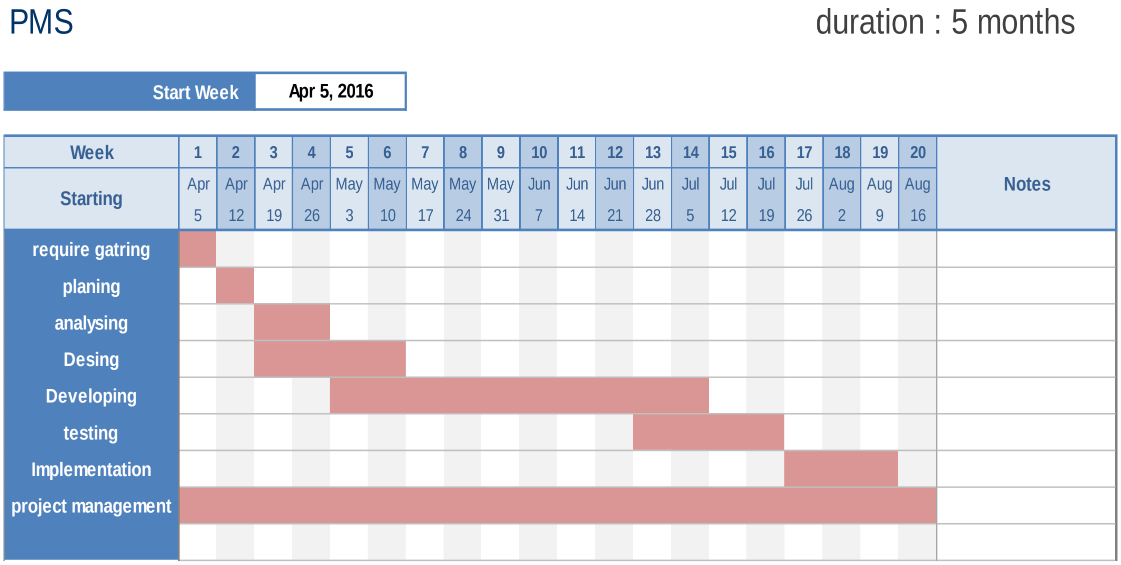
2.3.1.1 Planning

Planning phase is the general overview of the pharmacy and the objectives. There are two (2) indicates to be focused, the initial assessment and feasibility study. Initial assessment needs to the manual system changed to the new system (PMS), as the systematic management of the pharmacy.

For the feasibility study, it shows the hardware and software needed in general. Hardware is include the operating system is used, Windows 7 Professional platform, software development used Laravel framework of php language, and database development used MySQL.

In planning we used Gantt chart and PERT/CPM chart for project scheduling. Project scheduling is the process of deciding how the work in a project will be organized as separate tasks, and when and how these tasks will be executed [1] . You estimate the calendar time needed to complete each task, the effort required, and who will work on the tasks that have been identified.

[1] Sommerville, Ian. Software Engineering, 9th ed. Boston, MA: Pearson Education Inc., 2011, pp. 624-638.



PMS duration : 5 months

Start Week	Apr 5, 2016

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Notes
Starting	Apr 5	Apr 12	Apr 19	Apr 26	May 3	May 10	May 17	May 24	May 31	Jun 7	Jun 14	Jun 21	Jun 28	Jul 5	Jul 12	Jul 19	Jul 26	Aug 2	Aug 9	Aug 16	
require gatring	🟥																				
planing		🟥																			
analysing			🟥	🟥																	
Desing			🟥	🟥	🟥	🟥															
Developing					🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥							
testing													🟥	🟥	🟥	🟥					
Implementation																	🟥	🟥	🟥		
project management	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	

Figure 2.2 Gantt chart

Figure 2.2 Gantt chart of PMS project, project duration is 5 months. Gantt chart shows each activity starting date, duration and end date, and it show the total duration of PMS project.

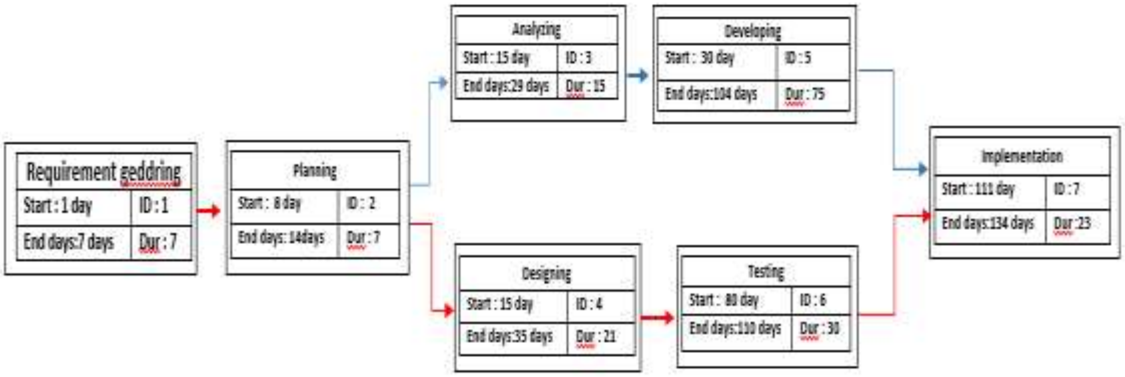


Figure 2.3 PERT/CPM chart
Critical path:1-2-4-6-7

Figure 2.3 show PERT/CPM chart, in this chart tasks are shown as boxes which each task has name, id, dependent task, duration. Also this chart shows the critical path in our project.

2.3.1.2 Analysis

Analysis phase is the definition of the problem that defines at planning phase. Pharmacist and system developer identify the process of the PMS system. It also needs for study the user requirement and the manual system. The result is logical data design. It is the specifying the conceptual data model, inputs, processes, and expected output requirements. PMS has nine (9) modules or processes that to be function.

The logical design is described using of data flow diagram, DFD and entity relationship, ER diagram, ERD. The results of the logical design are entities, attribute and the relationship of the entities within the database. It also describes the function of modulus for each process within the database environment. The users that involve in the PMS are pharmacy top manager, pharmacy stock manager and customer.

2.3.1.3 Detailed system design

Detailed system design phase is the completed design of the system processes. It is the specification of the PMS interface system, menus of the system, and the reports. The other approach that used is the web application development methodology.

2.3.1.4 Implementation

The implementation phase is starting at process installing hardware and software requirements. Installation hardware is setting up the PC desktop hardware requirements specification. Process of operating system installation is based on Windows XP Professional, Windows 7 and 8 Professional platform installation. Laravel php framework and MySQL are installing following the steps that required.

The language is used are Pre-Hypertext Processor, PHP script and Structured Query Language, SQL script. These scripts are type in PMS system application development. This will be testing and debugging, until it is ready to be delivered. The actual database is created and customizes the tables and view, and user authentication.

The testing process is to be doing to test the system application operation. After testing is concluded, the final documentation is reviewed and printed, and end users are trained. The system is in full operation at the end of this phase but will be continuously evaluated and fine-tuned.

2.3.1.5 Maintenance

Maintenance phase is involve when PMS is used in several times, there are the changes of the PMS that is requested by the pharmacy manager. The changes generate the system maintenance activities, which can be grouped into three types, corrective maintenance in response to system errors, adaptive maintenance due to changes in the business environment, perfective maintenance to enhance the system. For the PMS it maybe include these three (3) maintenance activities which is, pharmacist want to add the module or delete the module of the PMS application.

3. CHAPTER III

SYSTEM ANALYSIS AND DESIGN

3.0 Introduction

System is a collection of interrelated components that works together to achieve a purpose. System analysis is referred to the systematic examination or detailed study of a system in order to identify problems of the system, and using the information gathered in the analysis stage to recommend improvements or solution to the system.

System design is an abstract representation of a system components and their relationship and which describes the aggregated functionality and performance of the system. System design is also the overall plan or blueprint for how to obtain answer to the question being asked. The design specifies which of the various type of approach.

3.1 System Analysis

System analysis is the study of sets of interacting entities, including computer systems analysis. This field is closely related to requirements analysis or operations research. It is also "an explicit formal inquiry carried out to help someone identify a better course of action and make a better decision than he might otherwise have made. System Analysis is a methodology that involves the application of systematic approaches to collects facts about an existing system with the aim of improving it or replacing it with more efficient system within the context of the available resources. In other words, System analysis can also be viewed as the process of investigating a system, identifying problems and using the information to recommend improvements to the system.

3.1.1 Analysis of Existing System

Before we analyze the design of the proposed system, we need to carefully highlight the problems of the existing system so as to avoid recurrence. This analysis serves as a pointer on how to embark on building the proposed system that will help the Pharmacist provide optimal drug inventory management by monitoring the drug movement and state in the pharmacy. The problems of the current system should be outlined. Below are some of the problems associated with the existing system;

- Significant amount of time is allocated for writing the order as the pharmacist needs to go through the stock balance and make rough estimate for the amount to order based on Figures.
- The state of drugs in stock is manually checked.
- Mistake of selling expired drugs to customers.
- Too much workload on employees
- Filing cabinet in the pharmacy with paper record.

3.1.2 Analysis of Proposed System

From the problems listed in the existing system, the implementation of the proposed system shall focus on:

- Pharmacists having access to the proposed system at any time.
- Ensuring effective policing by providing statistics of the drugs in stock.
- Improving the efficiency of the system by ensuring effective monitoring of services and activities.
- Generating report within a specified period of time.
- Reducing the employees' workload.

3.2 Requirement Definition

Preliminary investigation plays an important role in developing a satisfactory requirement. Its' as a result of thorough investigation of how the current or the existing system works using the facts gathered at the preliminary investigation that leads to focusing on the possibility of replacing the existing system or improving upon the existing system. This task involves information gathering.

3.2.1 Method of Information Gathering

Collection of fact is the act of getting and gathering information from various sources in order to be able to compose the project. Data used for designing of the system were gathered through several means. Therefore the method used in the design and collections of information from various sources are as follows:

- Collecting and analyzing existing materials on the project topic, written by different expert.
- Studying the present system in detail and the organizational style.

- Knowing and understanding the input and output processes of the existing system.
- Interviews: A qualitative form of interview was conducted in the pharmacy to know the equipment needed, and the mode of operation of the old system.
- Primary data: This source has to do with the text book contacted for the development of this project.

3.3 System Design

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements through system modeling. One could see it as the application of systems theory to produce development. The design of this system will be user friendly. It shall be designed in such a way that employees will be able to navigate easily through the information supplied on the system.

In other words, system design consists of design activities that produce system specifications satisfying the functional requirements that were developed in the system analysis process. System design specifies how the system will accomplish. System design is the structural implementation of the system analysis.

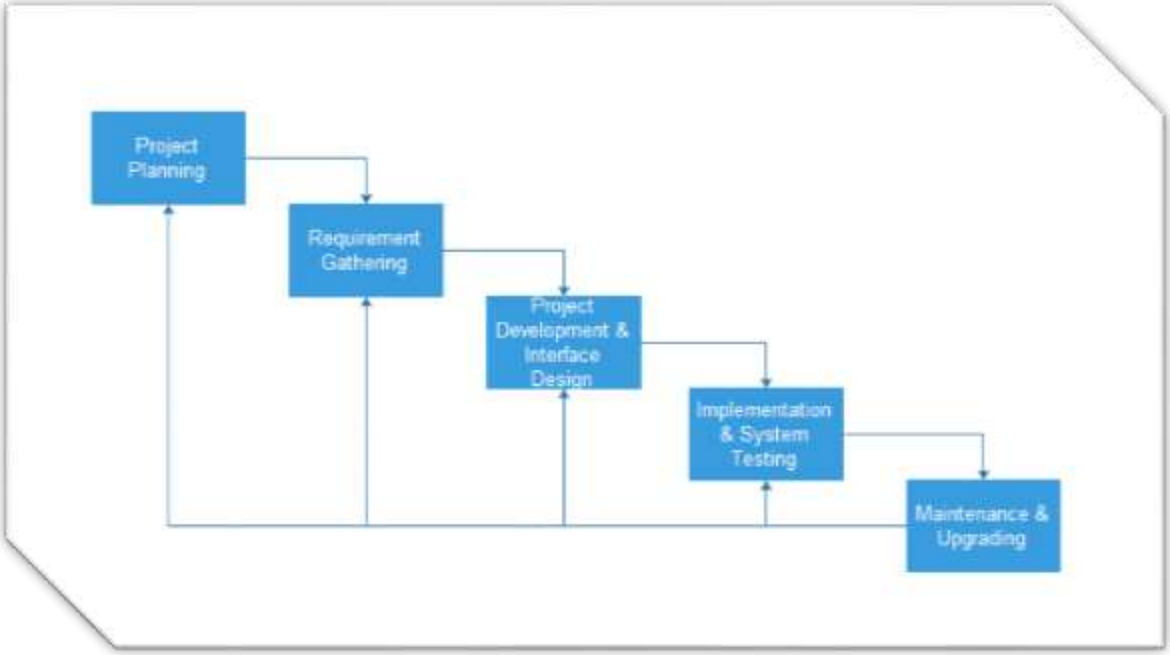

Project Planning
Requirement Gathering
Project Development & Interface Design
Implementation & System Testing
Maintenance & Upgrading

Figure 3.1: Waterfall Model of System Development Life Cycle

The diagram above is a system development life cycle that illustrates how the design of the project is broken down into five different phases, which are Project Planning, Requirement Gathering, Project Implementation and Interface Design, Implementation and System Testing, Maintenance and System Upgrading.

The proposed Pharmacy Management System for Sistan Pharma Company will start with project planning by determining the users of the system, aims and objectives of the project. After these, extensive research will be done to determine how to design an effective system, as well as to review the current system. Then, the design was with an initial prototype of the system, and then refined it based on their suggestions. Phases of analysis, design and implementation were performed iteratively until users and designers agreed on a final system specification. At this point, the project could move to the final implementation phase.

3.4 System Modelling

During the system requirements and design activity, systems may be modelled as a set of components and relationships between these components. These are normally illustrated graphically in a system architecture model that gives the reader an overview of the system organization. System modelling helps to give more detailed system specifications which are in form of graphical representations that can describe problem to be solved or the system that is to be developed. Because of the graphical representations used, models are often more understandable than detailed natural language description of the system requirements. Examples of such modelling tool is a System Flowchart.

3.5 System Flowchart

System flowchart is a type of diagram that represents an algorithm or process, showing the steps as boxes of various kinds, and their order by connecting these with arrows. This diagrammatic representation can give a step-by-step solution to a given problem. Process operations are represented in these boxes, and arrows connecting them represent flow of control. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields. Different symbols are used in the flowchart to represent input, output, decision, connectors and process.

3.5.1 Data Flow Diagram

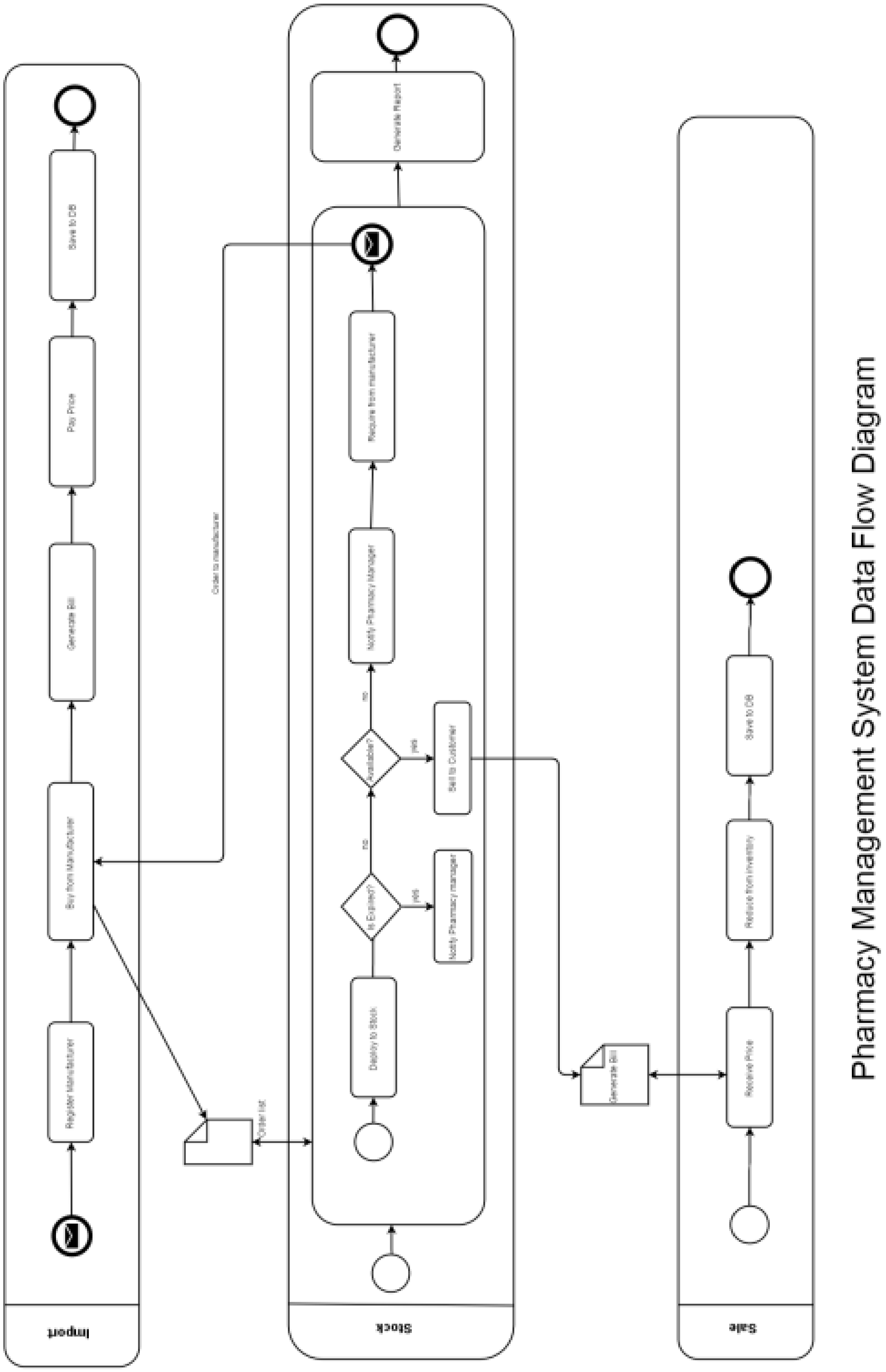


Figure 3.2 PMS Data Flow Diagram

Figure 3.2 shows the PMS Data Flow Diagram. This diagram shows the overall data flow in the different parts of the system.

3.5.2 Login Activity Flowchart

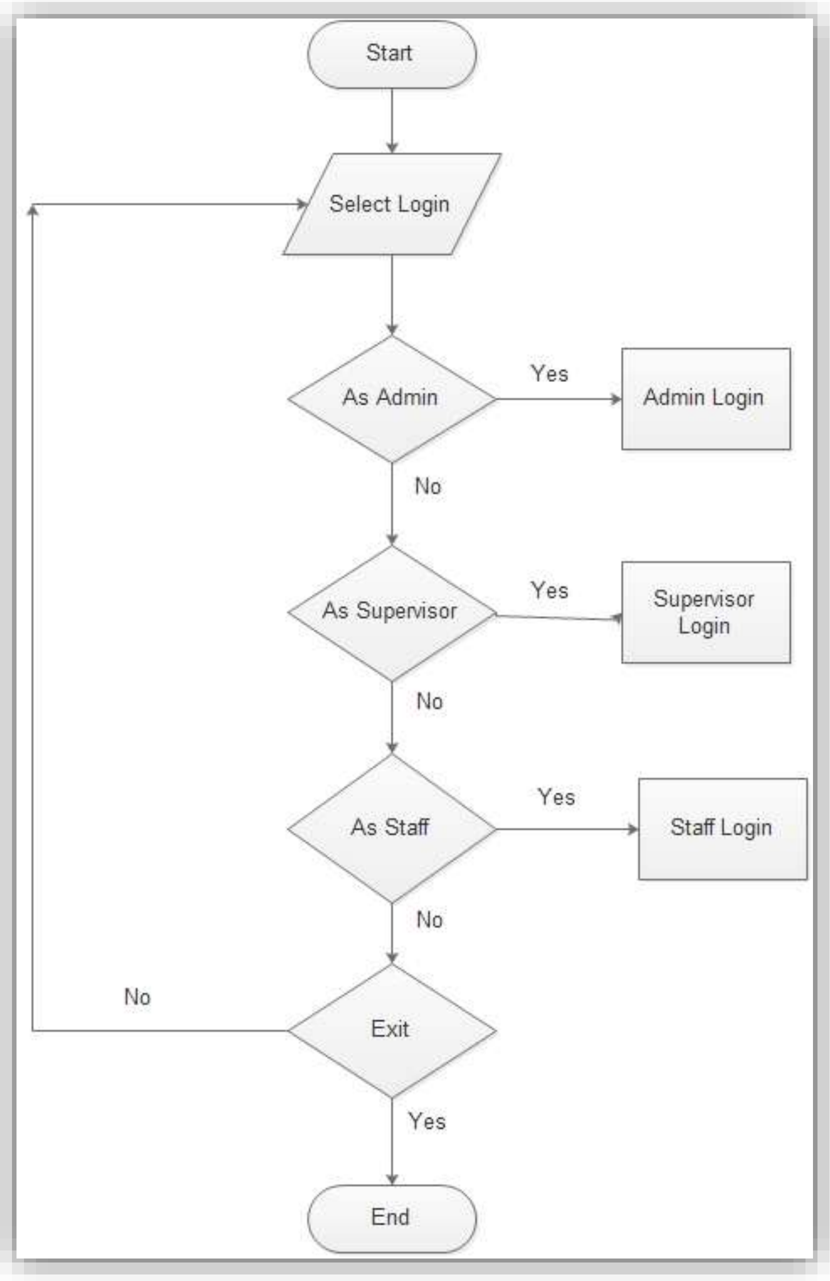


Figure 3.3 Login activity flow chart

Figure 3.3 shows the login flow chart which authorized registered users can login and unauthorized user login is not allowed.

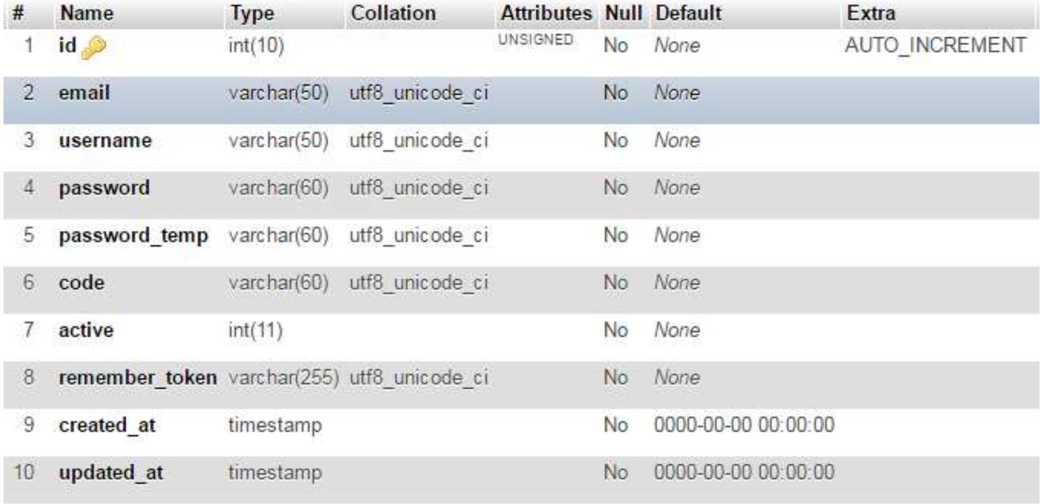
3.6 Database Design

This is a shared collection of data that are related or files that are to meet the immediate need of PMS users. These data are used in PMS for storing data of entities in database system.

3.6.1 Physical design or data dictionary

In physical design of PMS database every entity in the system is presented as database tables with related attributes/fields. Each table have its own attributes and it show have a unique field as primary key, and a key as foreign key.

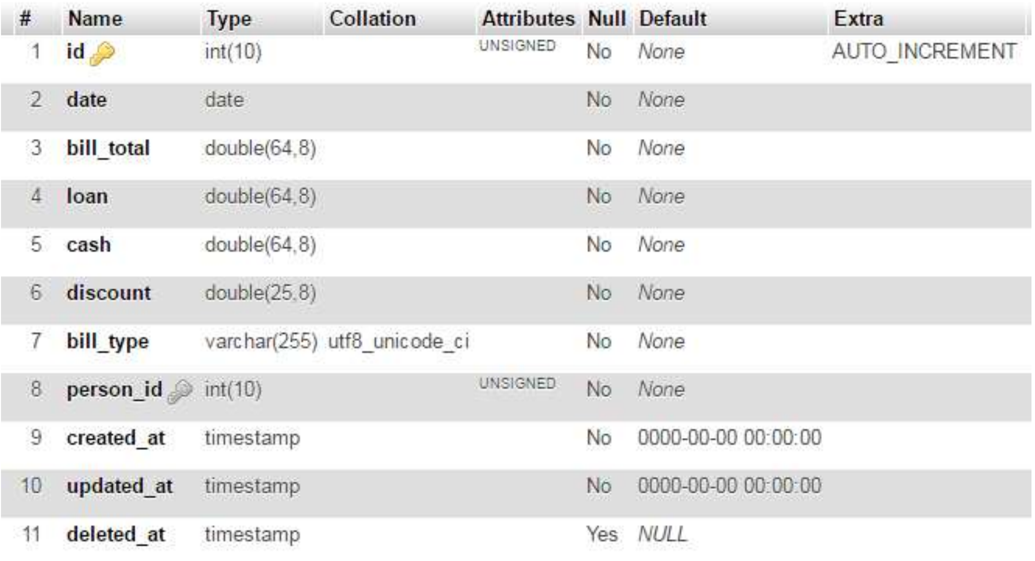
3.6.1.1 User Table



#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id 🔑	int(10)		UNSIGNED	No	*None*	AUTO_INCREMENT
2	email	varchar(50)	utf8_unicode_ci		No	*None*	
3	username	varchar(50)	utf8_unicode_ci		No	*None*	
4	password	varchar(60)	utf8_unicode_ci		No	*None*	
5	password_temp	varchar(60)	utf8_unicode_ci		No	*None*	
6	code	varchar(60)	utf8_unicode_ci		No	*None*	
7	active	int(11)			No	*None*	
8	remember_token	varchar(255)	utf8_unicode_ci		No	*None*	
9	created_at	timestamp			No	0000-00-00 00:00:00	
10	updated_at	timestamp			No	0000-00-00 00:00:00	

Figure 3.6.1.1 shows the User table, it has 10 fields and id is set as primary key. This table stores information about registered users in system.

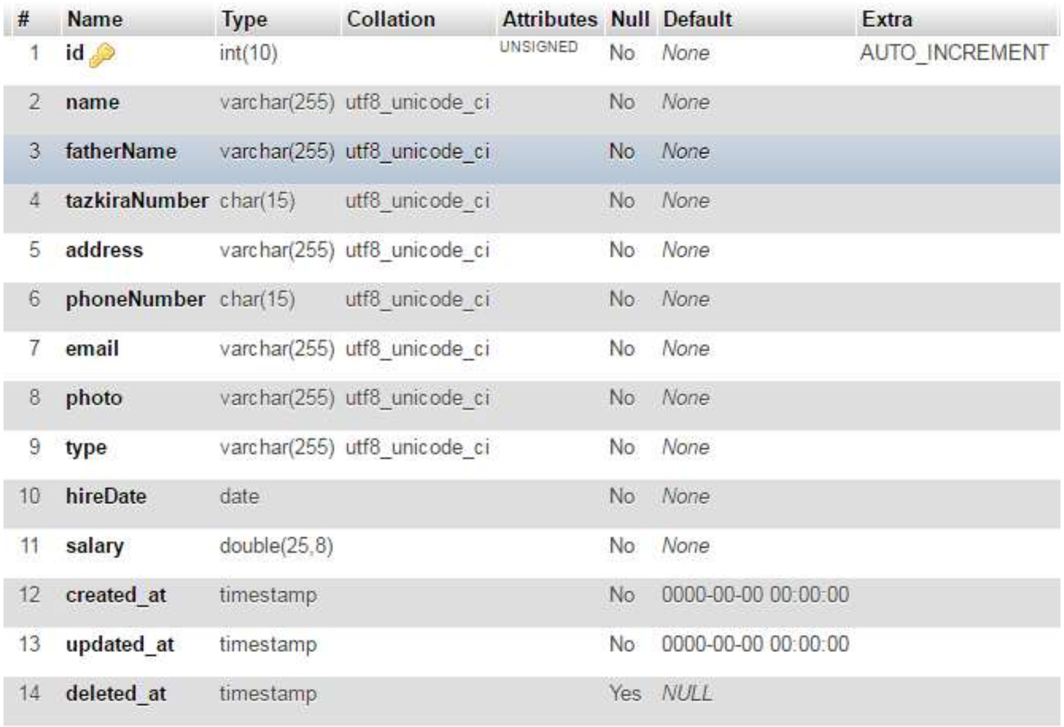
3.6.1.2 Bill Table



#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id 🔑	int(10)		UNSIGNED	No	*None*	AUTO_INCREMENT
2	date	date			No	*None*	
3	bill_total	double(64,8)			No	*None*	
4	loan	double(64,8)			No	*None*	
5	cash	double(64,8)			No	*None*	
6	discount	double(25,8)			No	*None*	
7	bill_type	varchar(255)	utf8_unicode_ci		No	*None*	
8	person_id 🔑	int(10)		UNSIGNED	No	*None*	
9	created_at	timestamp			No	0000-00-00 00:00:00	
10	updated_at	timestamp			No	0000-00-00 00:00:00	
11	deleted_at	timestamp			Yes	*NULL*	

Figure 3.6.1.2 Bill table. Information about sold medicines are stored in this table. The id is primary key and person-id is foreign key.

3.6.1.3 Employee Table



#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id 🔑	int(10)		UNSIGNED	No	*None*	AUTO_INCREMENT
2	name	varchar(255)	utf8_unicode_ci		No	*None*	
3	fatherName	varchar(255)	utf8_unicode_ci		No	*None*	
4	tazkiraNumber	char(15)	utf8_unicode_ci		No	*None*	
5	address	varchar(255)	utf8_unicode_ci		No	*None*	
6	phoneNumber	char(15)	utf8_unicode_ci		No	*None*	
7	email	varchar(255)	utf8_unicode_ci		No	*None*	
8	photo	varchar(255)	utf8_unicode_ci		No	*None*	
9	type	varchar(255)	utf8_unicode_ci		No	*None*	
10	hireDate	date			No	*None*	
11	salary	double(25,8)			No	*None*	
12	created_at	timestamp			No	0000-00-00 00:00:00	
13	updated_at	timestamp			No	0000-00-00 00:00:00	
14	deleted_at	timestamp			Yes	*NULL*	

3.6.1.3 Employee Table. In this table information about employees and their salary are stored. Employee id is set as primary key.

Other tables used in PMS database is attached in appendix C.

3.7 Logical design

In logical design of database the logical relationship between tables/entities specified. We showed the logical relationship as Entity Relationship (ER) Diagram. In ER Diagram relations between entities and kind of their relation (one-to-one, one-to-many) is graphically showed.

3.7.1 Entity Relation Diagram

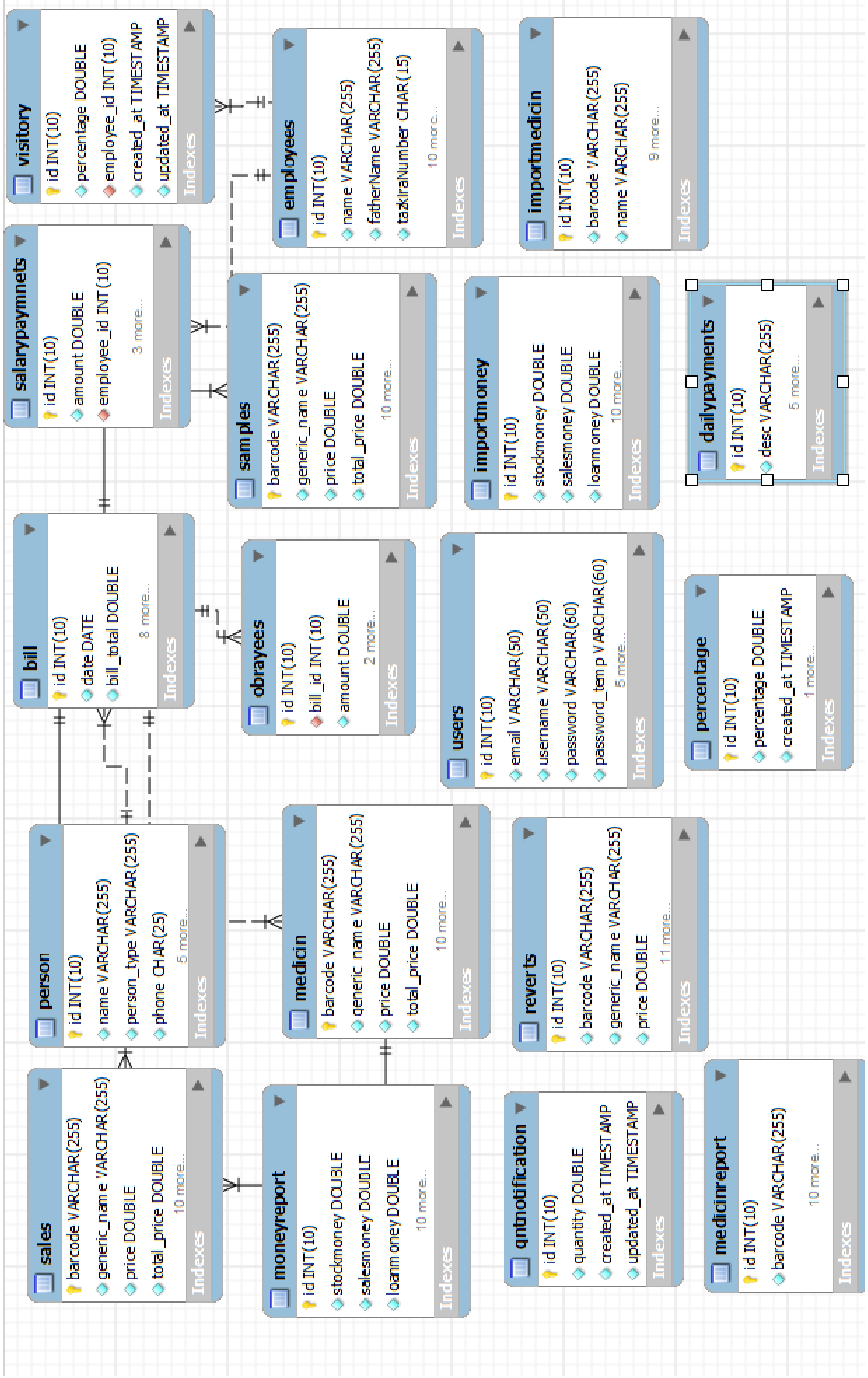


Figure 3.7.1 shows the Entity Relationship Diagram. This diagram shows relationship between tables in PMS database.

4. CHAPTER IV

SYSTEM IMPLEMENTATION

4.0 Introduction

System implementation is a stage in system life cycle whereby a new system is developed, installed and made ready for use. It is this stage that all details and key point in the requirement specification are implemented. System implementation therefore, is a very essential stage in which its success determines to a great extent the success of the new system. At this instance, after all is said and done the system is duly ready to be implemented (Pharmacy Management System).

System design is concerned mainly with the coordination of activities, job procedures and equipment utilization in order to achieve organizational objectives. It addresses data input and output data, processing and interface.

This stage involves the design of the new Pharmacy Management System a case study of Sistan Pharma Company.

4.1 Choice of Programming Language

Choosing a programming language depends on your language experience and the scope of the application you are building. While small applications are often created using only one language, it is not uncommon to develop large applications using multiple languages.

The proposed application to be built is a web based application that needs internet facilities to function.

The choice of environment to develop this application is Laravel framework of php. Laravel includes technologies like HTML, CSS with Bootstrap framework, JavaScript, and it supports connection to many database servers including MySQL.

Laravel is a free, open-source PHP web framework, created by Taylor Otwell and intended for the development of web applications following the model–view–controller (MVC) architectural pattern. Some of the features of Laravel are a modular packaging system with a dedicated dependency manager, different ways for accessing relational databases, utilities that aid in deployment and maintenance, and its orientation toward syntactic sugar.

Based on Laravel documentation "Laravel aims to make the development process a pleasing one for the developer without sacrificing application functionality. Happy developers make the best code. To this end, we've attempted to combine the very best of what we have seen in other web frameworks, including frameworks implemented in other languages, such as Ruby on Rails, ASP.NET MVC, and Sinatra"[1].

Laravel is accessible, yet powerful, providing powerful tools needed for large, robust applications. A superb inversion of control container,

[1] https://laravel.com/docs/4.2/introduction

expressive migration system, and tightly integrated unit testing support give you the tools you need to build any application with which you are tasked.

4.2 System Testing and Debugging

Testing is an integral part of software development processes. This is to ensure that the quality requirement of the application is not compromised by testing and debugging program modules before they are integrated, testing the system to ensure an effective inter-operability after integration. Debugging has to do with fixing of errors encountered during program execution. System testing deals with the real life testing of the system, to ascertain how far it has gone in carrying out the expected task. This was carried out in two phases.

Number one is the source code testing (white box testing) which examine the logic of the program. Secondly, the specification testing which involves the examination of the system as regard to what it should do and how it should be done given specific conditions. This includes inputting data, collecting its output and comparing it with the output of the old system and assessing it to see if it can replace the old system.

4.3 System Documentation

Documentation. Descriptive information (e.g., hardcopy manuals, on-line help files, Web sites) that portrays the use and/or operation of the system [1] .System documentation is a crucial aspect of implementation process. It describes the working of components and serves as a method of communication between application developers and users. It also helps

[1] Pressman, Roger S. *Software Engineering*. 5th ed. Boston, Mass.: McGraw Hill, 2001, pp.274.

future analysis of application either by the same or different system analysts and developers. We wrote a user manual for system users and a final documentation for all over application development, also as much as possible we used good programming techniques which help future developers in reading and understanding of our work.

4.4 Hardware Requirement

- A minimum hard disk space of 20 Gigabytes (GB)
- RAM size of 2GB
- Pentium 4 dual processor CPU
- A VGA colour monitor
- Mouse
- Keyboard

4.5 Software Requirement

- Windows operating system such as Windows 7 or Windows 8.
- Internet Explorer 9, Mozilla Firefox, Google Chrome.

4.6 Database Specification

A database is a single file which consists of structured data and records which are stored in minimum or no duplication of data. It is therefore a constructed, consistent and controlled pool of data. A good database must be common to all users and independent of the programs which use it to generate output.

MySQL was used as the database application tool for designing the database management system. The database management system is limited only to database administrator (Management). Whilst the system designer/developer/programmer is responsible for maintaining and upgrading of the database and the whole software.

4.7 Module Description

4.7.1 Login Form

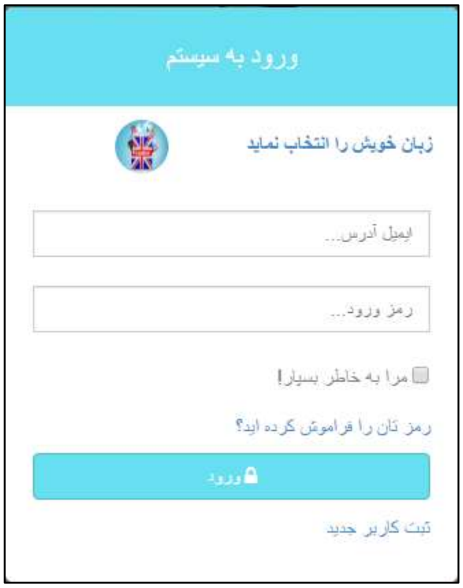
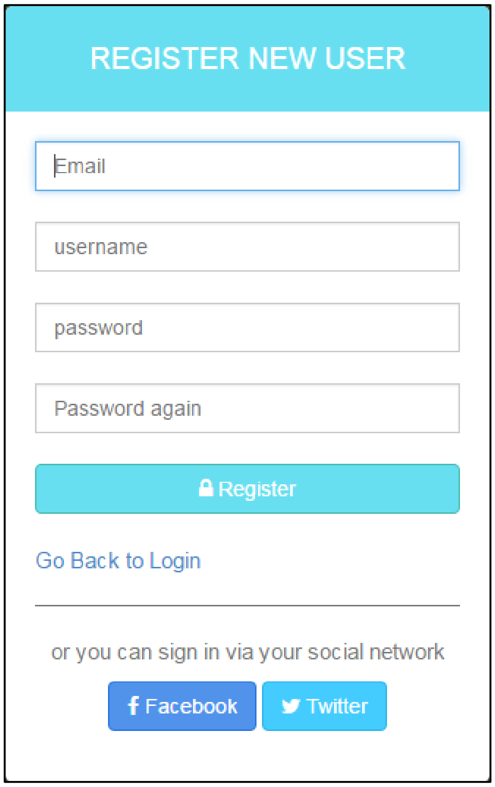


Figure 4.7.1 shows the login form which authorized users enter email and password to login. If user had forgot the password they can require password reset from admin.

4.7.2 User Registration Form



REGISTER NEW USER

Email

username

password

Password again

Register

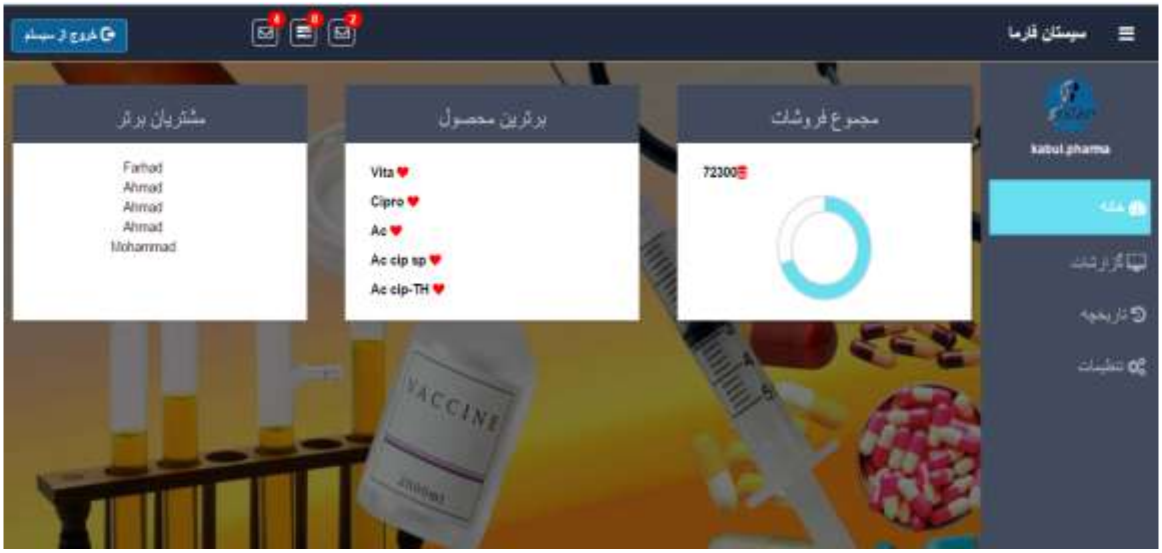
Go Back to Login

or you can sign in via your social network

Facebook Twitter

Figure 4.7.2 user registration Form for signing up new users to system.

4.7.3 Main Page



مشتریان برتر

Farhad
Ahmad
Ahmad
Ahmad
Mohammad

برترین محصول

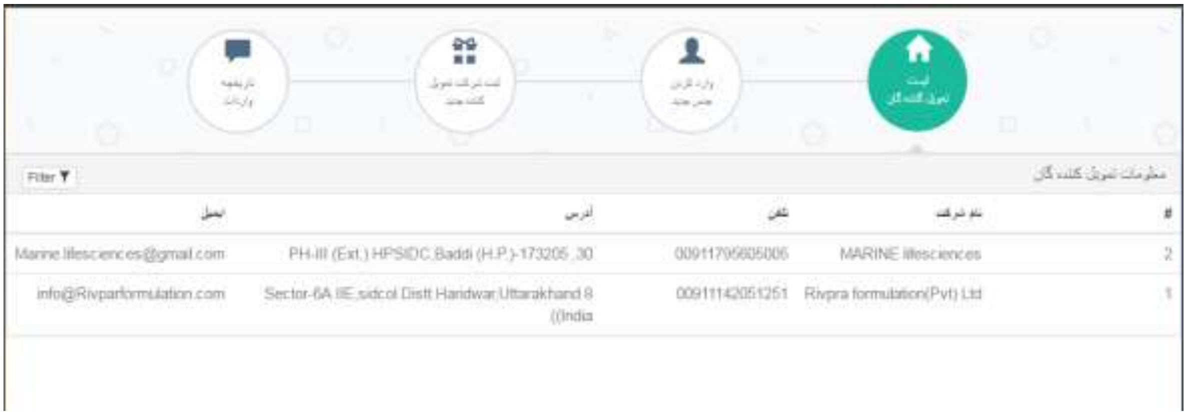
Vita ♥
Cipro ♥
Ac ♥
Ac cip sp ♥
Ac cip-TH ♥

مجموع فروشات

72300

Figure 4.7.3 Home page of application which shows an overall report of selling, top products, and top customers. Also at left top of page admin can see notification of medicine expiration, inventory out-of-stack, and pending task.

4.7.4 Import page

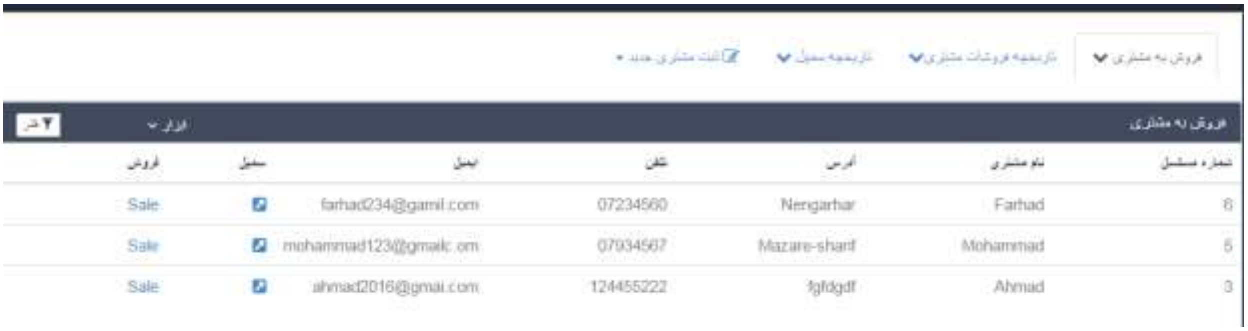


Filter ▼ معلومات تمویل کننده گان

ایمیل	آدرس	تلفن	نام شرکت	#
Marine.lifesciences@gmail.com	PH-III (Ext.) HPSIDC Baddi (H.P.)-173205 ,30	00911795605005	MARINE lifesciences	2
info@Rivparformulation.com	Sector-6A IIE,sidcol Distt Haridwar,Uttarakhand 8 ((India	00911142051251	Rivpra formulation(Pvt) Ltd	1

Figure 4.7.4 shows the import page where admin can see a list of provider companies, import new medicines from suppliers, register new supplier company, and see history of import.

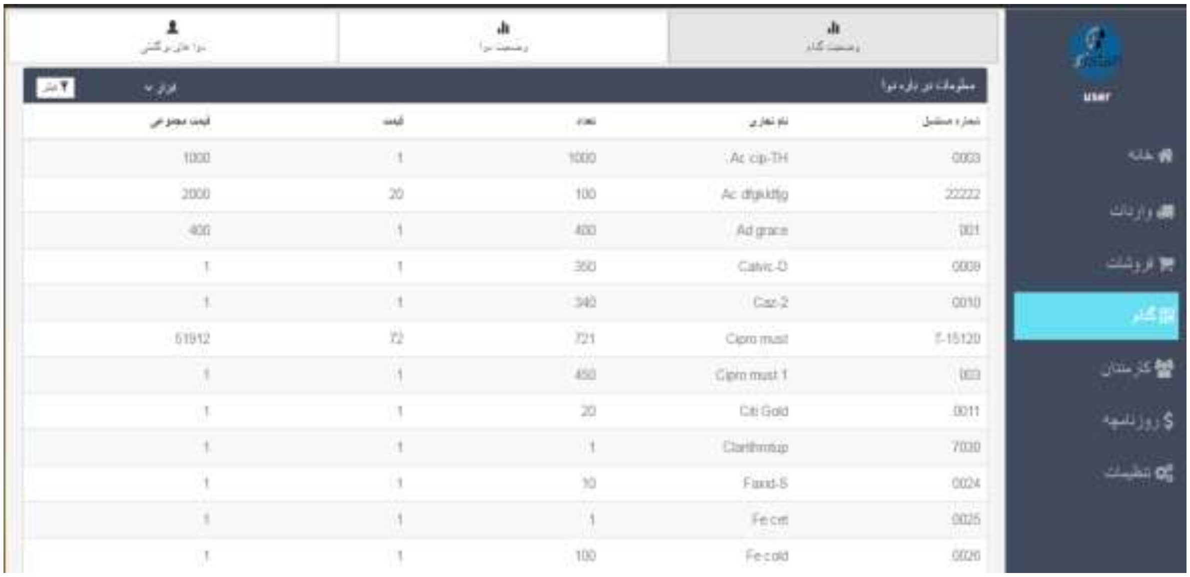
4.7.5 Sales page



فروش	سمپل	ایمیل	تلفن	آدرس	نام مشتری	شماره مسلسل
Sale	✎	farhad234@gamil.com	07234560	Nengarhar	Farhad	6
Sale	✎	mohammad123@gmailc.om	07934567	Mazare-sharif	Mohammad	5
Sale	✎	ahmad2016@gmai.com	124455222	fgfdgdf	Ahmad	3

Figure 4.7.5 Sales page provides options of selling or sell ass sample to customer, viewing history of sales and samples, and registering new customer.

4.7.6 Stock page



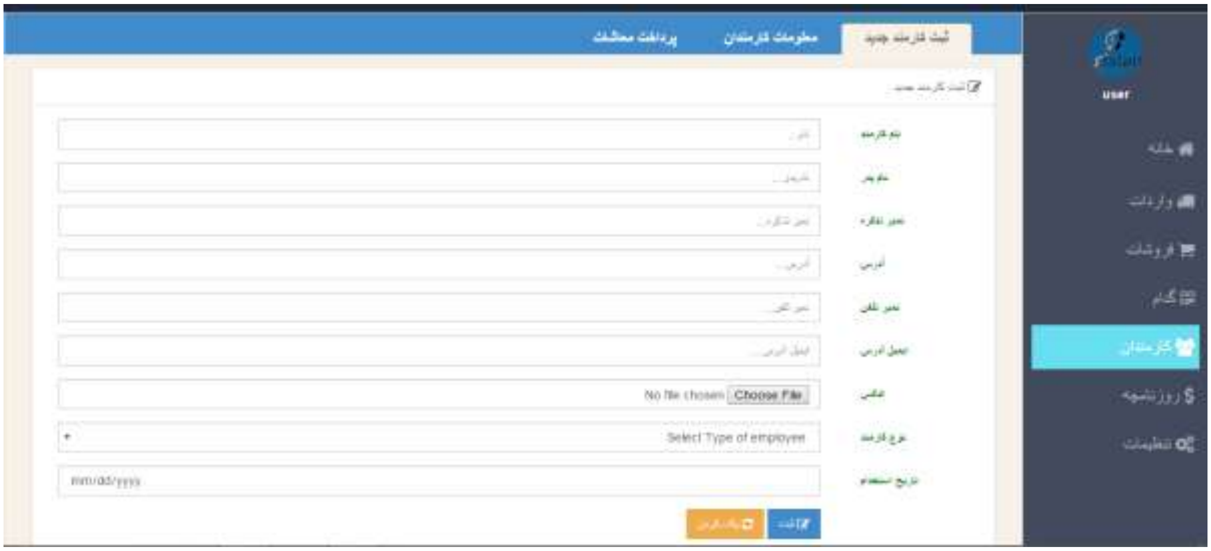
دوا های برگشتی | وضعیت دوا | وضعیت گدام

معلومات در باره دوا

قیمت مجموعی	قیمت	تعداد	نام تجاری	شماره مسلسل
1000	1	1000	Ac cip-TH	0003
2000	20	100	Ac dfgkldfg	22222
400	1	400	Ad grace	001
1	1	350	Calvic-D	0009
1	1	340	Caz-2	0010
51912	72	721	Cipro must	T-15120
1	1	450	Cipro must 1	003
1	1	20	Cti Gold	0011
1	1	1	Clarithrotop	7030
1	1	10	Faxid-S	0024
1	1	1	Fe cet	0025
1	1	100	Fe cold	0026

Figure 4.7.6 shows the stock page. In this page admin can see stock condition, medicine condition, and list of revert medicine.

4.7.7 Employee page



ثبت کارمند جدید | معلومات کارمندان | پرداخت معاشات

ثبت کارمند جدید

نام کارمند	
نام پدر	
نمبر تذکره	
آدرس	
نمبر تلفن	
ایمیل آدرس	
عکس	Choose File No file chosen
نوع کارمند	Select Type of employee
تاریخ استخدام	mm/dd/yyyy

Figure 4.7.7 Employee page. In employee page admin can add new employee, see information of employees, and pay employee’s salary.

4.7.8 Daily payment page

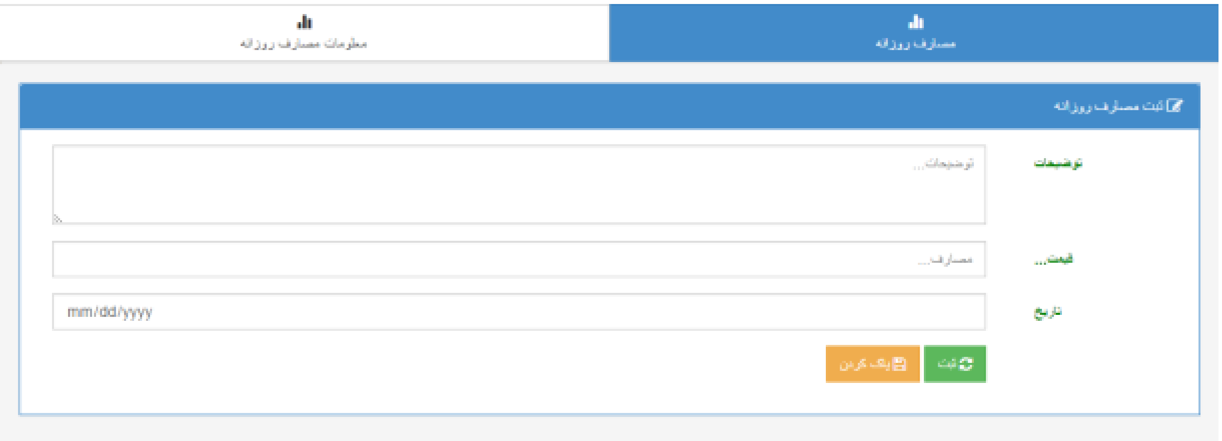


Figure 4.7.8 Daily payment page. In this page user can add and see report of daily payments for trivial goods.

4.8 System Maintenance

Maintenance is a continuous process of making modifications and upgrading the application. This usually commences after the application has gone into use. Maintenance phase is involve when PMS is used in several times, there are the changes of the PMS that is requested by the pharmacy manager. The changes generate the system maintenance activities, which can be grouped into three types, corrective maintenance in response to system errors, adaptive maintenance due to changes in the business environment, perfective maintenance to enhance the system. For the PMS it maybe include these three (3) maintenance activities which is, pharmacist want to add the module or delete the module of the PMS application.

5 CHAPTER V

CONCLUSION AND RECOMMENDATION

5.0 Summary

Pharmacy management system is designed to improve the accuracy, enhance safety and efficiency in the pharmaceutical store. It is a computer based system which helps the Pharmacist to improve inventory management, cost, medical safety etc.

Pharmacy management system was developed to ensure the security of information and reliability of Pharmacy records when accessing and providing services to the customers. The information gathered during the data collection was properly analyzed and the results provided the basis for the new system. The system was tested and found to be functional and the outputs produced by this system were encouraging. The application will hence reduce the loss of information unlike the existing system and also information will be processed fast.

5.1 Conclusion

Effective implementation of this software will take care of the basic requirements of the pharmacy management system because it is capable of providing easy and effective storage of information related to activities happening in the stipulated area. With these, the objectives of the system design will be achieved.

In order to allow for future expansion, the system has been designed in such a way that will allow possible modification as it may deem necessary by the pharmacy management, whenever the idea arises.

5.2 Recommendation

Designing this application (Pharmacy management system) is not an easy task. It all started from the requirement gathering and passes through so many other stages before completion.

Based on the benefits of this system and tremendous value it will add to customer-user satisfaction, the below recommendation will be considered;

It is recommended that the new system should be used with the necessary specifications of the system requirements and provision for an uninterrupted power supply should be made available throughout the hours of operation of the pharmacy to avoid power outage. There should also be basic computer knowledge for the users of the software.

REFERENCES

1- Pressman, Roger S. “*Software Engineering”*. 5th ed. Boston, Mass.: McGraw Hill, 2001.

2- Shelly, Gary B and Harry J Rosenblatt. “*Systems Analysis And Design”*. 8th ed. Boston, Mass.: Thomson Course Technology, 2010.

3- Sommerville, Ian. “Software Engineering”. 9th ed. Boston, Mass.: Pearson Education Inc., 2011.

Appendix A

Sales Module Model Code

PMS Application uses Model View Controller (MVC) design pattern. It divides a given software application into three interconnected parts, so as to separate internal representations of information from the ways that information is presented to or accepted from the user. Here we put the Model code of Sales Module.

```
<?php

        /**
        *
        */
        class sale extends Eloquent
        {

                protected $table        = 'sales';
                protected $primarykey  = 'barcode';

                public function bill() {
                        return $this->belongsT0('bill');
                }
        }

?>
```

Appendix B

Sales Module Controller Code

Here we just put part of Sales Module Controller.

```php
<?php
    /**
    *
    */
    class salesController extends BaseController
    {
        public function salesGet() {
            $customer = DB::table('person')->orderBy('id','desc')
                                      -
>where('person_type','customer')->get();
            // username
            $username = Auth::user()->username;
            $type = 'loan';
            return View::make('saleas',array(
                'customers' => $customer,
                'expire'        => '',
                'type' => $type,
                'username' => $username

                ));
        }
        // register customer
        public function registerCustomerPost() {
            $firstName = Input::get('firstName');
            $address = Input::get('address');
            $phone = Input::get('phone');
            $email       = Input::get('email');
            $person_type = 'customer';

            $person = new person;
            $person->name = $firstName;
            $person->person_type = $person_type;
            $person->phone = $phone;
            $person->address = $address;
            $person->email = $email;

            if($person->save()) {
                return Redirect::route('sales-get')
                        ->with('success','مشتری جدید موفقانه ثبت
گردید!');
            }else {
                return Redirect::route('sales-get')
                     ->with('error','ثبت مشتری جدید ناموفق بود لطفا دوباره کوشش
نمایید!');
            }
```

```
		}
		/*
		| SALES TO CUSTOMER
		*/
		public function salesToCustomer(){
			// $material		=
array('Siftrakson','Afghanistan','12','paid','2010-10-10','2011-10-
10','1000','8 grm','30',)
			$material = Input::get('material');
			$loan = Input::get('remain');
			$paid = Input::get('pay');
			$total = Input::get('all');
			$discount = Input::get('discount');
			$discount = $discount * $total / 100;
			$suplier_id = Input::get('suplier_id');
			$date = date("Y-m-d");
			$bill_type = '';
			$person_type = Input::get('person_type');
			$visitor	= Input::get('visitor');

			if($person_type == 'sample') {
				$bill_type = 'sample';
				if (count($material) < 1) {
				return "لطفاً اول معلومات را وارد نمایید!";
			}else{
```

```

                $bill = new bill;
                $bill->date = $date;
                $bill->bill_total      = $total;
                $bill->loan = $loan;
                $bill->cash = $paid;
                $bill->discount        = $discount;
                $bill->bill_type       = $bill_type;
                $bill->person_id = $suplier_id;
                $bill->save();

                $bill_id = Bill::latest('id')->first();
                $bill_id = $bill_id->id;

                $num_elements = 0;
                while ($num_elements < count($material)) {
                    $medicin = new sample;
                    $medicin->generic_name =
    $material[$num_elements];
                    $num_elements++;
                    $medicin->madein =
    $material[$num_elements];
                    $num_elements++;
                    $medicin->quantity =
    $material[$num_elements];
```

```
                    // take the quantity
                    $quantity = $material[$num_elements];
                    $num_elements++;
                    $medicin->status = $material[$num_elements];
                    $num_elements++;
                    $medicin->product_date =
$material[$num_elements];
                    $num_elements++;
                    $medicin->expire_date  =
$material[$num_elements];
                    $num_elements++;
                    $medicin->price = $material[$num_elements];
                    $num_elements++;
                    $medicin->weight =
$material[$num_elements];
                    $num_elements++;
                    $medicin->barcode  =
$material[$num_elements];
                    // Take Batch Number
                    $batchNumber      =
$material[$num_elements];
                    $num_elements++;
                    $medicin->total_price     =
$material[$num_elements];
                    $medicin->bill_id = $bill_id;
                    $medicin->save();
```

```
                        $stock          =
medicin::where('barcode',$batchNumber)->first();
                        $medicin = $stock->quantity;
                        $price          = $stock->price;
                        $stock          = $medicin - $quantity ;
                        $totalPrice = $stock * $price;
                        $affectedRows = medicin::where('barcode', '=',
$batchNumber)->update(array('quantity' => $stock,'total_price' =>
$totalPrice));
                        $num_elements++;
                  }
                  return "سمپل موفقانه ثبت گردید!";
                  }

                  }else {

                  if($person_type == 'Customer') {
                        $bill_type  = 'customerSell';
                  }else {
                        $bill_type = 'agencySell';

                  }
                  if (count($material) < 1) {
                        return "لطفاً اول معلومات خود را وارد نمایید!";
                  }else {
```

```

            $bill     = new bill;
            $bill->date = $date;
            $bill->bill_total       = $total;
            $bill->loan = $loan;
            $bill->cash = $paid;
            $bill->discount       = $discount;
            $bill->bill_type       = $bill_type;
            $bill->person_id = $suplier_id;
            $bill->save();

            $bill_id = Bill::latest('id')->first();
            $bill_id = $bill_id->id;

            $num_elements = 0;
            while ($num_elements < count($material)) {
                $medicin = new sale;
                $medicin->generic_name =
$material[$num_elements];
                $num_elements++;
                $medicin->madein =
$material[$num_elements];
                $num_elements++;
                $medicin->quantity =
$material[$num_elements];
```

```
                              // take the quantity
                              $quantity = $material[$num_elements];
                              $num_elements++;
                              $medicin->status = $material[$num_elements];
                              $num_elements++;
                              $medicin->product_date =
$material[$num_elements];
                              $num_elements++;
                              $medicin->expire_date  =
$material[$num_elements];
                              $num_elements++;
                              $medicin->price = $material[$num_elements];
                              $num_elements++;
                              $medicin->weight =
$material[$num_elements];
                              $num_elements++;
                              $medicin->barcode  =
$material[$num_elements];
                              // Take Batch Number
                              $batchNumber                        =
$material[$num_elements];
                              $num_elements++;
                              $medicin->total_price      =
$material[$num_elements];
                              $medicin->bill_id = $bill_id;
                              $medicin->save();
```

```
                $stock          =
medicin::where('barcode',$batchNumber)->first();
                $medicin = $stock->quantity;
                $price          = $stock->price;
                $stock          = $medicin - $quantity ;
                $totalPrice = $stock * $price;
                $affectedRows = medicin::where('barcode', '=',
$batchNumber)->update(array('quantity' => $stock,'total_price' =>
$totalPrice));
                $num_elements++;
            }
            if($visitor != 'انتخاب کارمند') {
                $visitorPercentage = percentage::first();
                $visitory = new visitor;
                $percentage =  $visitorPercentage->percentage
* $total / 100;
                $visitory->percentage      = $percentage;
                $visitory->employee_id   = $visitor;
                $visitory->save();
             }
            return "بیل موفقانه اجرا شد!";
            }
            }
        }
        /*
        | RETRIEVE DATE BASE ON BATCH NUMBER
```

```
    */
    public function retrieveDate() {
        $batchNumber = Input::get('batchNumber');
        $person_type = Input::get('person_type');
        $stock  = medicin::where('barcode',$batchNumber)->first();
        if($person_type == 'Customer') {

        $Visitor = employee::where('type','Employee')->get();
        $x = '<div class="form-group">
   <div class="col-sm-9">
                <select class="form-control" id="visitor-select">
                <option>انتخاب کارمند</option>

                ';
        foreach ($Visitor as $value) {
            $x .= '<option value='.$value->id.'>
                        '.$value->name.'
                    </option>';
        }

        $x .='
        </select>
        </div>
```

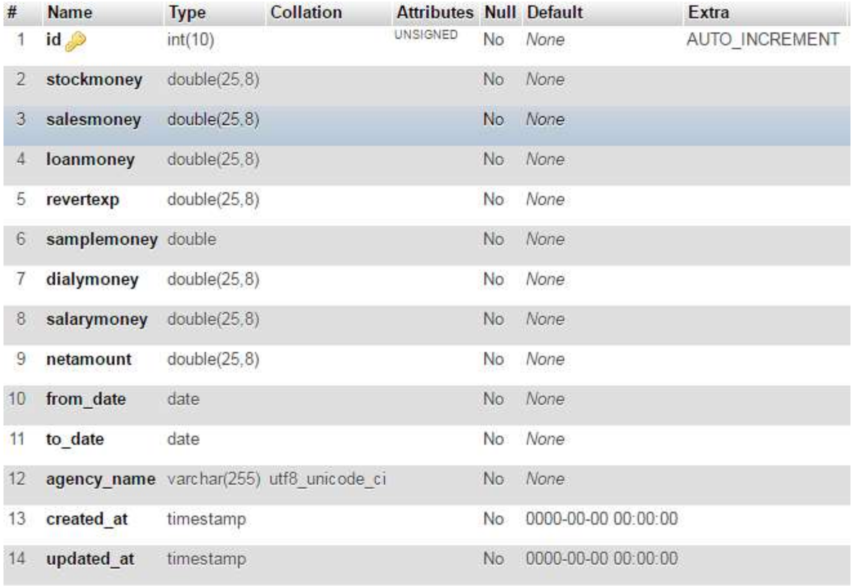
```
                  <label for="date" class="col-sm-3 control-label"> انتخاب
کاارمند</label>
                  </div>';

                  }else {
                        $x = '';
                  }
         return '<div id="sales-content-bill">
          <div class="form-group">
           <div class="col-sm-9">
            <input class="form-control" id="name" name="name"
type="text" value='.$stock->generic_name.'>
           </div>
             <label for="name" class="col-sm-3 control-label"> نام
دوا</label>
          </div>
          <div class="form-group">
           <div class="col-sm-9">
            <input class="form-control" id="weight" name="weight"
type="text" value='.$stock->weight.'>
           </div>
           <label for="weight" class="col-sm-3 control-label"> نام
تجاری</label>
          </div>
          <div class="form-group">
           <div class="col-sm-9">
           <input class="form-control" id="madIn" name="madeIn"
type="text" value='.$stock->madein.'>
          </div>
            <label for="madIn" class="col-sm-3 control-
label">ساخت</label>
        </div>
        <div class="form-group">
          <div class="col-sm-9">
           <input class="form-control" id="amount" name="amount"
type="number">
          </div>
            <label for="amount" class="col-sm-3 control-
label">تعداد</label>
        </div>
        <div class="form-group">
          <div class="col-sm-9">
           <input onblur="total();" class="form-control" id="price"
name="price" type="number">
          </div>
          <label for="price" class="col-sm-3 control-label">قيمت</label>
        </div>
        <div class="form-group" style="display: none;">
          <div class="col-sm-9">
           <input class="form-control" id="totalPrice"
name="TotalPrice" type="number">
          </div>
          <label for="Tatal Price" class="col-sm-3 control-label"> قیمت
مجموعی</label>
         </div>
         <div class="form-group">
          <div class="col-sm-9">
           <input class="form-control" id="productDate"
name="product_date" type="date" value='.$stock->product_date.'>
          </div>
          <label for="date" class="col-sm-3 control-label"> تاریخ
تولید</label>
         </div>
         <div class="form-group">
          <div class="col-sm-9">
           <input class="form-control" id="expireDate"
name="expire_date" type="date"  value='.$stock->expire_date.'>
          </div>
            <label for="date" class="col-sm-3 control-label"> تاریخ
انقضا</label>
         </div>
        </div>
            }
```

Appendix C

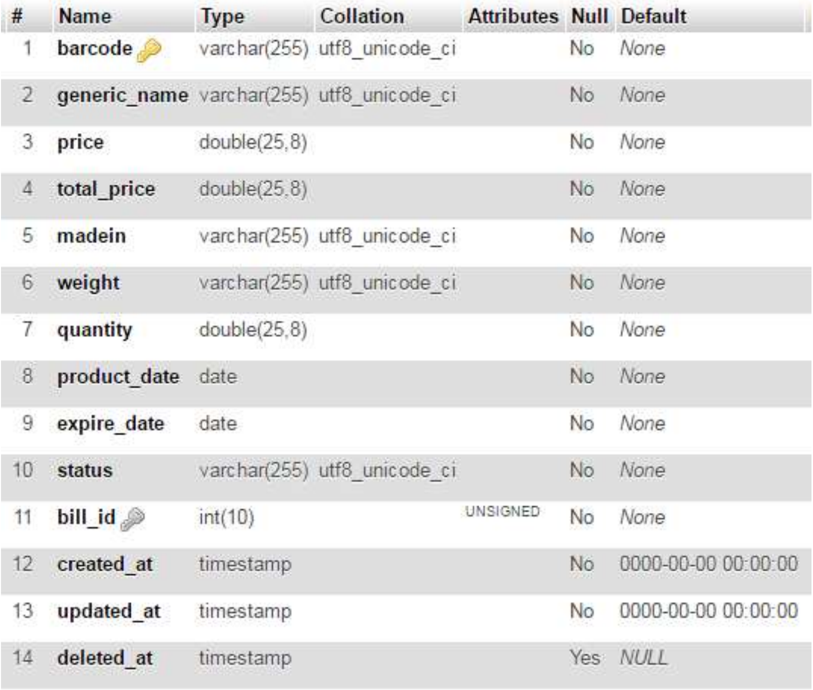
Tables in PMS Database

C.1 Import_money Table



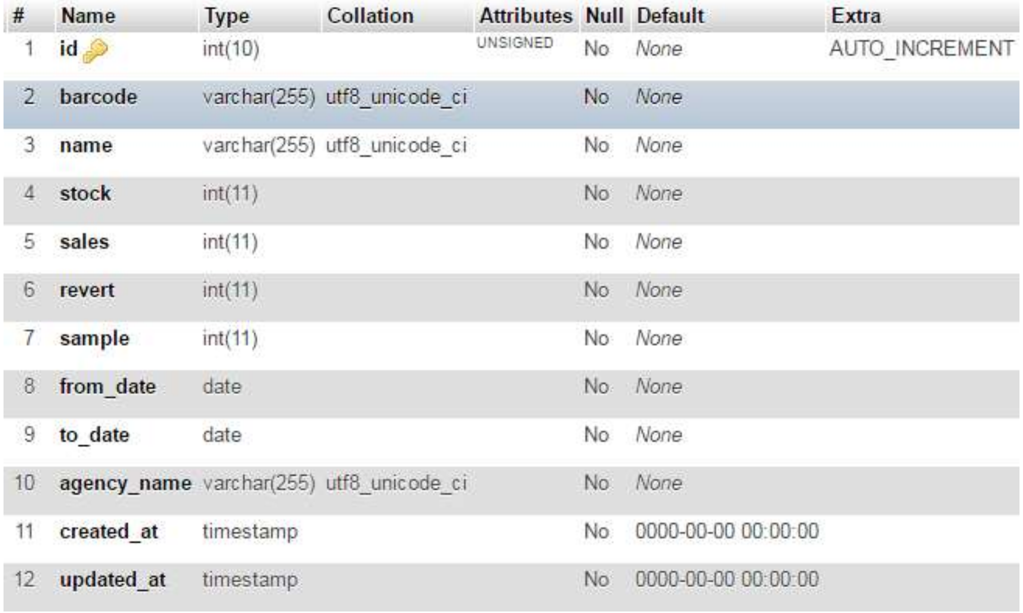
#	Name	Type	Collation	Attributes	Null	Default	Extra
1	**id** 🔑	int(10)		UNSIGNED	No	*None*	AUTO_INCREMENT
2	**stockmoney**	double(25,8)			No	*None*	
3	**salesmoney**	double(25,8)			No	*None*	
4	**loanmoney**	double(25,8)			No	*None*	
5	**revertexp**	double(25,8)			No	*None*	
6	**samplemoney**	double			No	*None*	
7	**dialymoney**	double(25,8)			No	*None*	
8	**salarymoney**	double(25,8)			No	*None*	
9	**netamount**	double(25,8)			No	*None*	
10	**from_date**	date			No	*None*	
11	**to_date**	date			No	*None*	
12	**agency_name**	varchar(255)	utf8_unicode_ci		No	*None*	
13	**created_at**	timestamp			No	0000-00-00 00:00:00	
14	**updated_at**	timestamp			No	0000-00-00 00:00:00	

C.2 Medicine Table



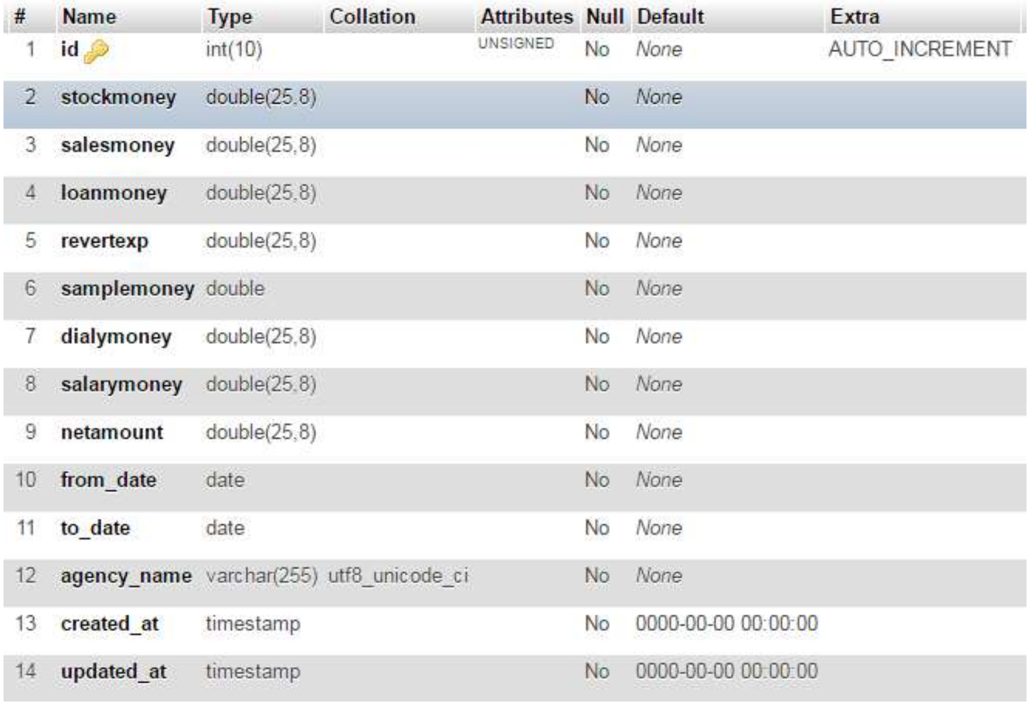
#	Name	Type	Collation	Attributes	Null	Default
1	**barcode** 🔑	varchar(255)	utf8_unicode_ci		No	*None*
2	**generic_name**	varchar(255)	utf8_unicode_ci		No	*None*
3	**price**	double(25,8)			No	*None*
4	**total_price**	double(25,8)			No	*None*
5	**madein**	varchar(255)	utf8_unicode_ci		No	*None*
6	**weight**	varchar(255)	utf8_unicode_ci		No	*None*
7	**quantity**	double(25,8)			No	*None*
8	**product_date**	date			No	*None*
9	**expire_date**	date			No	*None*
10	**status**	varchar(255)	utf8_unicode_ci		No	*None*
11	**bill_id** 🔑	int(10)		UNSIGNED	No	*None*
12	**created_at**	timestamp			No	0000-00-00 00:00:00
13	**updated_at**	timestamp			No	0000-00-00 00:00:00
14	**deleted_at**	timestamp			Yes	*NULL*

C.3 Medicinereport Table



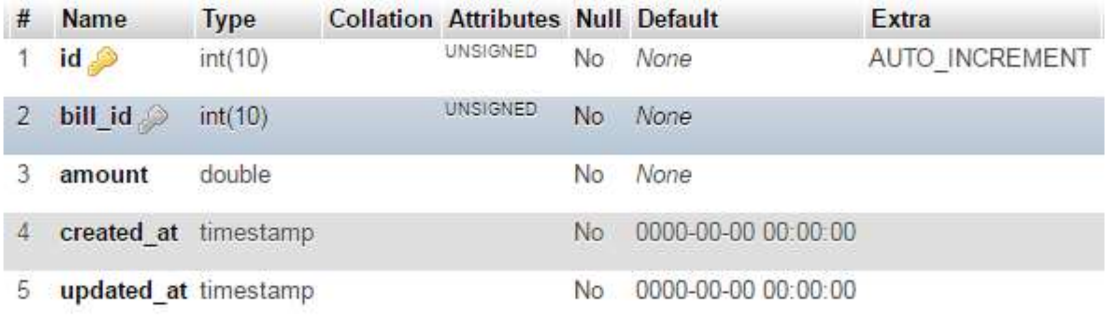
#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id 🔑	int(10)		UNSIGNED	No	*None*	AUTO_INCREMENT
2	barcode	varchar(255)	utf8_unicode_ci		No	*None*	
3	name	varchar(255)	utf8_unicode_ci		No	*None*	
4	stock	int(11)			No	*None*	
5	sales	int(11)			No	*None*	
6	revert	int(11)			No	*None*	
7	sample	int(11)			No	*None*	
8	from_date	date			No	*None*	
9	to_date	date			No	*None*	
10	agency_name	varchar(255)	utf8_unicode_ci		No	*None*	
11	created_at	timestamp			No	0000-00-00 00:00:00	
12	updated_at	timestamp			No	0000-00-00 00:00:00	

C.4 Moneyreport Table



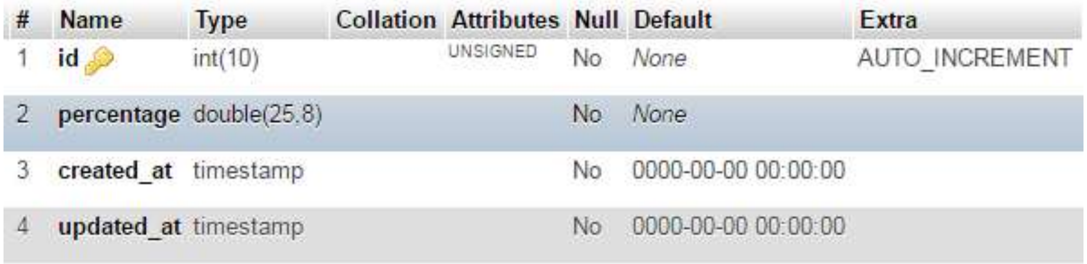
#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id 🔑	int(10)		UNSIGNED	No	*None*	AUTO_INCREMENT
2	stockmoney	double(25,8)			No	*None*	
3	salesmoney	double(25,8)			No	*None*	
4	loanmoney	double(25,8)			No	*None*	
5	revertexp	double(25,8)			No	*None*	
6	samplemoney	double			No	*None*	
7	dialymoney	double(25,8)			No	*None*	
8	salarymoney	double(25,8)			No	*None*	
9	netamount	double(25,8)			No	*None*	
10	from_date	date			No	*None*	
11	to_date	date			No	*None*	
12	agency_name	varchar(255)	utf8_unicode_ci		No	*None*	
13	created_at	timestamp			No	0000-00-00 00:00:00	
14	updated_at	timestamp			No	0000-00-00 00:00:00	

C.5 Obrayees Table



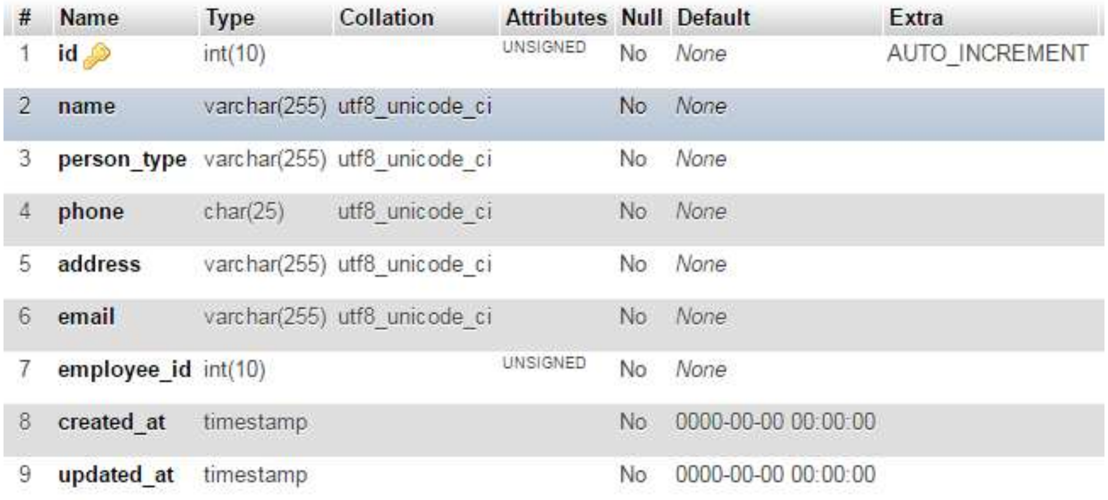
#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id 🔑	int(10)		UNSIGNED	No	*None*	AUTO_INCREMENT
2	bill_id 🔑	int(10)		UNSIGNED	No	*None*	
3	amount	double			No	*None*	
4	created_at	timestamp			No	0000-00-00 00:00:00	
5	updated_at	timestamp			No	0000-00-00 00:00:00	

C.6 Percentage Table



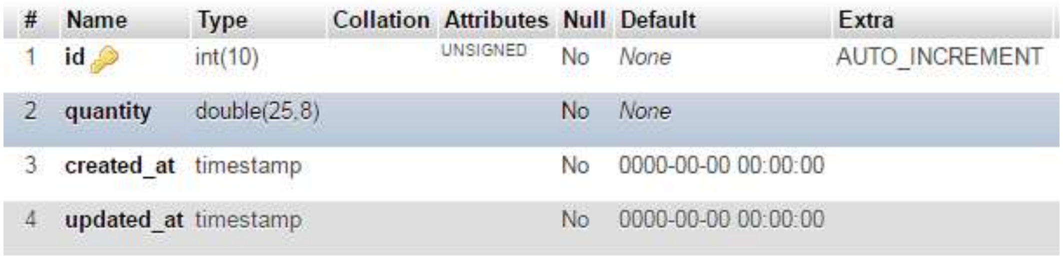
#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id 🔑	int(10)		UNSIGNED	No	*None*	AUTO_INCREMENT
2	percentage	double(25,8)			No	*None*	
3	created_at	timestamp			No	0000-00-00 00:00:00	
4	updated_at	timestamp			No	0000-00-00 00:00:00	

C.7 Person Table



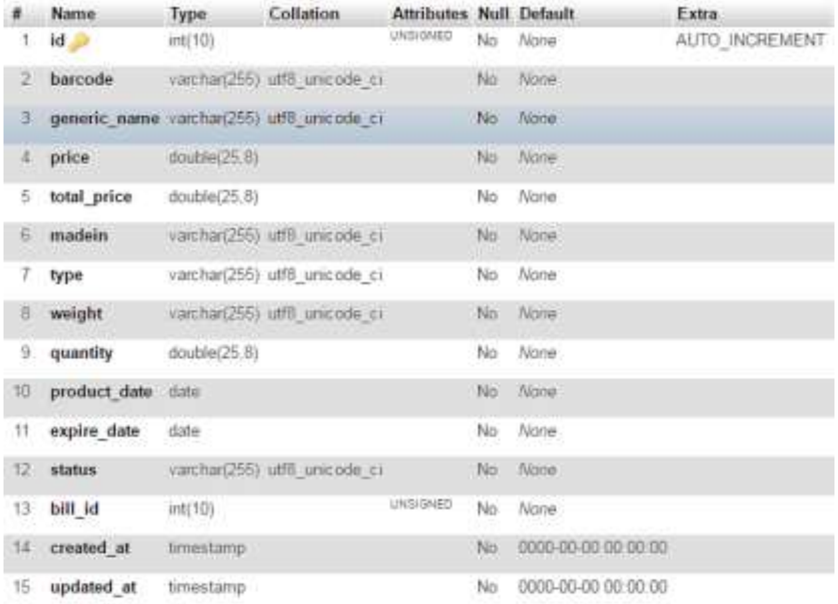
#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id 🔑	int(10)		UNSIGNED	No	*None*	AUTO_INCREMENT
2	name	varchar(255)	utf8_unicode_ci		No	*None*	
3	person_type	varchar(255)	utf8_unicode_ci		No	*None*	
4	phone	char(25)	utf8_unicode_ci		No	*None*	
5	address	varchar(255)	utf8_unicode_ci		No	*None*	
6	email	varchar(255)	utf8_unicode_ci		No	*None*	
7	employee_id	int(10)		UNSIGNED	No	*None*	
8	created_at	timestamp			No	0000-00-00 00:00:00	
9	updated_at	timestamp			No	0000-00-00 00:00:00	

C.8 Qntnotification Table



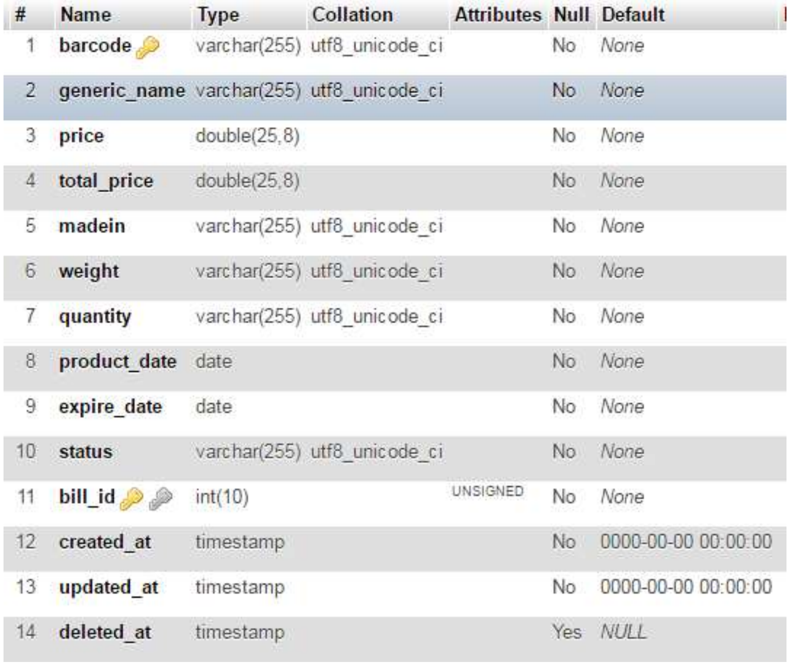
#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	int(10)		UNSIGNED	No	*None*	AUTO_INCREMENT
2	quantity	double(25,8)			No	*None*	
3	created_at	timestamp			No	0000-00-00 00:00:00	
4	updated_at	timestamp			No	0000-00-00 00:00:00	

C.9 Reverts Table



#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	int(10)		UNSIGNED	No	*None*	AUTO_INCREMENT
2	barcode	varchar(255)	utf8_unicode_ci		No	*None*	
3	generic_name	varchar(255)	utf8_unicode_ci		No	*None*	
4	price	double(25,8)			No	*None*	
5	total_price	double(25,8)			No	*None*	
6	madein	varchar(255)	utf8_unicode_ci		No	*None*	
7	type	varchar(255)	utf8_unicode_ci		No	*None*	
8	weight	varchar(255)	utf8_unicode_ci		No	*None*	
9	quantity	double(25,8)			No	*None*	
10	product_date	date			No	*None*	
11	expire_date	date			No	*None*	
12	status	varchar(255)	utf8_unicode_ci		No	*None*	
13	bill_id	int(10)		UNSIGNED	No	*None*	
14	created_at	timestamp			No	0000-00-00 00:00:00	
15	updated_at	timestamp			No	0000-00-00 00:00:00	

C.10 Sales Table



#	Name	Type	Collation	Attributes	Null	Default
1	**barcode** 🔑	varchar(255)	utf8_unicode_ci		No	*None*
2	**generic_name**	varchar(255)	utf8_unicode_ci		No	*None*
3	**price**	double(25,8)			No	*None*
4	**total_price**	double(25,8)			No	*None*
5	**madein**	varchar(255)	utf8_unicode_ci		No	*None*
6	**weight**	varchar(255)	utf8_unicode_ci		No	*None*
7	**quantity**	varchar(255)	utf8_unicode_ci		No	*None*
8	**product_date**	date			No	*None*
9	**expire_date**	date			No	*None*
10	**status**	varchar(255)	utf8_unicode_ci		No	*None*
11	**bill_id** 🔑 🔑	int(10)		UNSIGNED	No	*None*
12	**created_at**	timestamp			No	0000-00-00 00:00:00
13	**updated_at**	timestamp			No	0000-00-00 00:00:00
14	**deleted_at**	timestamp			Yes	*NULL*