## Chapter: 1

## Introduction

**1.1 Background Of Study**

Each and every office needs management software for manage all the work done in the office. Before start the work we have done some study such as which type of company it is. What they want in the software how to calculate their worker salary and attendance. How to store there stocks product.

* 1. **Objectives**

Around in attendance world, everything is technically sophisticated. The ultimate objective of the system is to provide facility to the user for management of a company. Security of this system is very high and the possibility of doing wrong in the calculation is low. Since, now-a-days every system become increasingly technically advanced, the proposed system will involve computerized Apply system, Recruitment system, database storage, retrieval (through several functions), evaluation, agreement, modifications and decision making supports which will make all processes involving the system much faster and easier for the users.

The main objective of this system is to record all information including stocks and profit of daily monthly , which is essential when we need any kind of information than we can easily find out whenever needed how many medicine are available or not. This software provides to find out which medicine are available which are not.

* 1. **Broad Objective**

The broad objective of this project is to use our institutional educational experience in the real life working.

**1.4 Specific Objective**

1. Easily view pharmacist info.

2. Easily view manager info.

3 Automatically show daily monthly report.

4 Easily view all prescription list & view all medicine list that are available.

5 Medicine price show automatically.

1. Easily view pharmacist info.
2. Easily view manager info.
3. Automatically show, daily monthly report.
4. Easily view all prescription list & view all medicine list that are available.
5. Medicine price show automatically.
6. Medicine can search drug name, category, type.

**1.5 Proposed System**

Compared to the current manual system, the implementation of Inventory Management System will reduced the time spent for paperwork. With the proposed system, the user will be able to monitor the movement of the products with ease. It will bring fast movement for purchase and selling Products. Also, it will give more exact data on monetary transaction.

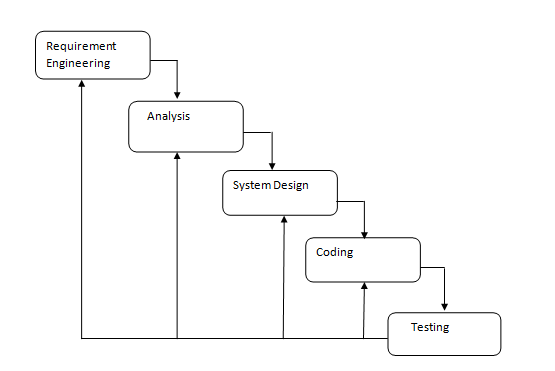
The system handles all aspects of the inventory control function. It allows the shopper to receive new batches of products, delete obsolete product and identification of product in the database. Furthermore, the system eases the process of stock replenishment.

**1.6 System Benefits:**

* Faster and efficient processing.
* Easily insert and manage result.
* History Backup.
* Easily accessible.
* Maintain privacy

**1.7 Software Process Model**

To do this project, I have chosen waterfall model as software process model.



**Fig 1.2 : Waterfall Process Model**

**Advantages of Waterfall Model:**

* This model is simple and easy to understand and use.
* It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
* In this model phases are processed and completed one at a time. Phases do not overlap.
* Waterfall model works well for smaller projects where requirements are very well understood

**Disadvantages of waterfall model:**

* Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.
* No working software is produced until late during the life cycle.
* High amounts of risk and uncertainty.
* Not a good model for complex and object-oriented projects.
* Poor model for long and ongoing projects.
* Not suitable for the projects where requirements are at a moderate to high risk of changing.

**When to use the waterfall model:**

* This model is used only when the requirements are very well known, clear and fixed.
* Product definition is stable.
* Technology is understood.
* There are no ambiguous requirements
* Ample resources with required expertise are available freely
* The project is short
* model phases are processed and completed one at a time

**1.8 Methodology**

The development process on “Pharmacy Management System” will complete following the structure described later on Software Analysis & Design. It is tentative in nature. The variables identified to manipulate through a handy inspection and from primary and secondary data.

**1.9 Data Sources:**

For this project in data collection phase we collected two types of data i.e.

1. Primary Data

2. Secondary Data

**Primary Data**

Primary data are generated within the organization. The organizations practical experience, observation, and face-to-face interview with our own web administrators helped us generate the primary data. Actually the primary data have collected through the alveron communication limited experience, observation, and face-to-face interview with both operators and tourist.

**Secondary Data**

Secondary data are collected by real life experience and studying different articles, newspapers, and research papers and of course information collected via Internet. Data, facts and statistics collected from different web sites and sources made us understand the project better.

**1.10 Feasibility Study**

Feasibility study determines whether that solution is feasible or achievable for the organization. There are three major areas of feasibility study. On studying the feasibility of the system, three major considerations are dealt with, to find whether the automation of the system is feasible.

* Technical feasibility
* Economic feasibility
* Operational feasibility

**Technical Feasibility**

For completed this project I used Notepad++ with Zencoding MySQL database system. I’ve also used HTML5 CSS and calculation are done in Ajax. And to run this project a simple configurable computer is needed. All this technology which I have mentioned above is ready to use. So I can consider this project is technically feasible.

To use the system users will require personal computer with windows operating system, Microsoft SQL Server database. All of these are available technologies now a day. So, the project is technically feasible.

**Economic Feasibility**

By using this system authority will be benefited. They save their time in medicine price calculation. They can sell many medicine within short time. We need not to pay extra money for any software (except windows operating system). Our hardware is enough to run our software. so we can say it is economically feasibility.

**Operational feasibility**

User will feel comfortable to use this system. There is no complexity that can confuse the user. The proposed system is designed from user who have minimal knowledge to operate computer. So the system is operationally feasible.

.

## Chapter: 2

**Requirement Engineering**

**2.1: User Requirement & System Requirement**

**User Requirement: Admin**

1. Admin will be able to login

2 Admin manage pharmacist.

3 Admin manage manager.

4 Admin manage medicine stock

5Admin generate report.

**System Requirement**: **Admin**

**1. Admin will be able to login.**

* 1. Admin will be to login with username & password.
  2. Admin must have authorized.

**2. Admin manage manager.**

2.1 : Admin add manager.

2.2 Admin view manager info.

**3. Admin manage Pharmacist**.

3.1 : Admin add pharmacist.

3.2 Admin view pharmacist info.

**4. Admin manage medicine stock**

4.1 : Admin see medicine list.

4.2 : Admin can search medicine.

4.3 Admin can edit delete medicine

**5. Admin generate report.**

5.1 Admin can see daily & monthly report.

**User requirement: Pharmacist**

1. Pharmacist will be able to login.

2. Pharmacist manage prescription.

3. Pharmacist manage stock.

**System requirement: Pharmacist**

**1. Pharmacist will be able to login.**

1.1 : Pharmacist will be able to login with username & pw.

1.2: Pharmacist must have authorized.

**2. Pharmacist manage prescription.**

2.1: Add patient information

2.2: Add patient drug.

2.3: See cart info.

**3. Pharmacist manage stock.**

3.1 : Admin see medicine list.

3.2 : Admin can search medicine.

3.3 Admin can edit delete medicine

**User Requirement : Manager**

1. Manager will be able to login with username & password.

2. Manager view prescription list.

3. Manager manage stock.

**System Requirement:**

**1. Manager will be able to login.**

1.1: Manager will be able to login with username & password.

1.2 : Manager must be authorized.

**2. Manager view prescription.**

2.1: Manager view all prescription list.

**3.Manager manage medicine stock.**

3.1: Manager add, delete , update medicine.

3.2: Manager search medicine.

3.3 Manager see all medicine stock.

**2.2 : Functional Requirement:**

1. Admin add manager.

2. Admin add pharmacist.

3. Admin view pharmacist info.

4. Admin view manager info.

5. Admin see report.

6. Admin see medicine stock.

6. Pharmacist add prescription , add cart drug.

7. Pharmacist see medicine stock & sell medicine.

8. Manager add medicine .

9. View medicine stock & edit delete medicine.

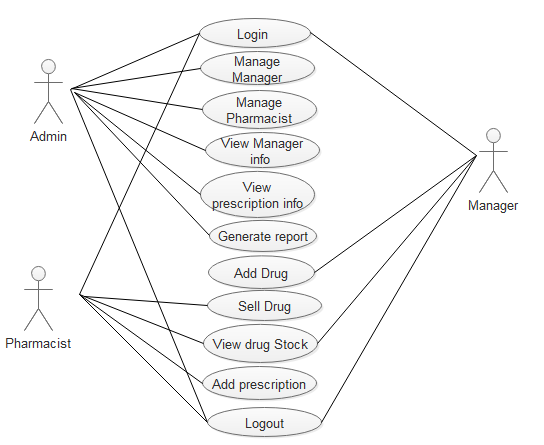
**2.3: Non Functional Requirement:**

1. System will need all the username and password stored in database.

2. System will need username and password information from the Admin

**2.4: Use Case Diagram:**

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

****

**Figure 2.1: Use Case Diagram**

**2.5 Use Case Text:**

Actor : Admin

Description: Admin can control all System. Admin can add manager, Pharmacist. I will add , delete, edit manager & pharmacist information. Can see daily report how many drug are sold today. Admin can also see monthly report by select month.

Actor: Pharmacist

Description: Pharmacist see prescription list & stock. Pharmacist can add prescription and sell medicine and can edit delete medicine stock

Actor: Manager.

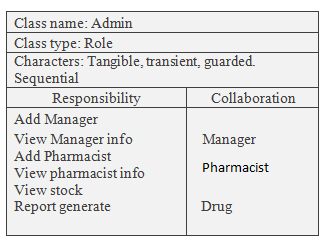
Description : Manager view manager info. Can see prescription list. Can add new medicine & view stock & can edit update this.

## Chapter: 3

**Analysis**

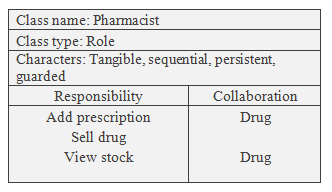
**3.1: Class Responsibilities Collaboration:**

**CRC : For Admin**

****

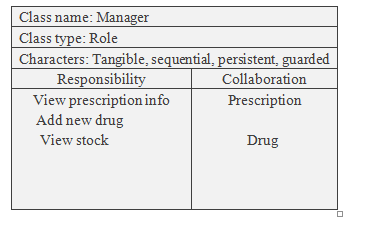
**Fig 3.1: CRC For Admin**

**CRC : For Pharmacist**

****

**Fig 3.2: CRC For Pharmacist**

**CRC For Manager:**

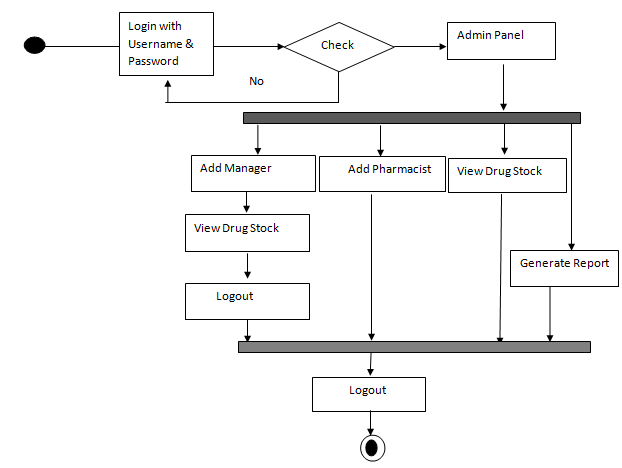
****

**Fig 3.4: CRC For Manager**

**3.2 Activity Diagram:**

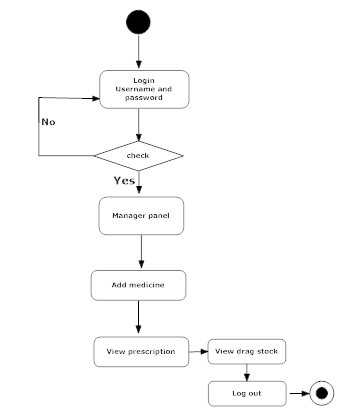
Activity diagrams, which are related to program flow plans (flowcharts), are used to illustrate activities. In the external view, I use activity diagrams for the description of those business processes that describe the functionality of the business system.

**Activity For Admin:**

****

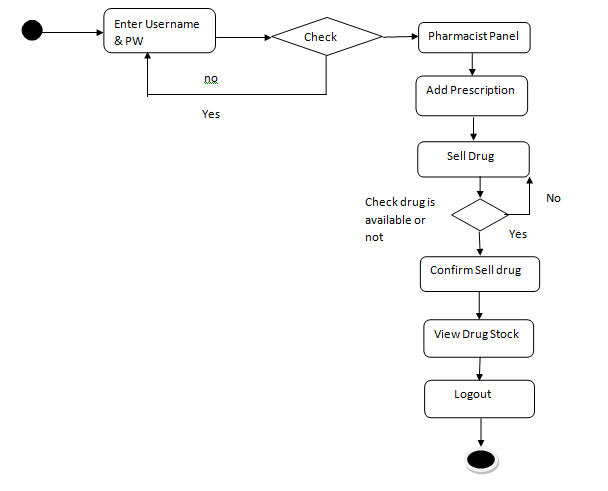
**Fig 3.4: Activity For Admin**

**Activity For Manager:**



**Fig 3.5: Activity For Manager**

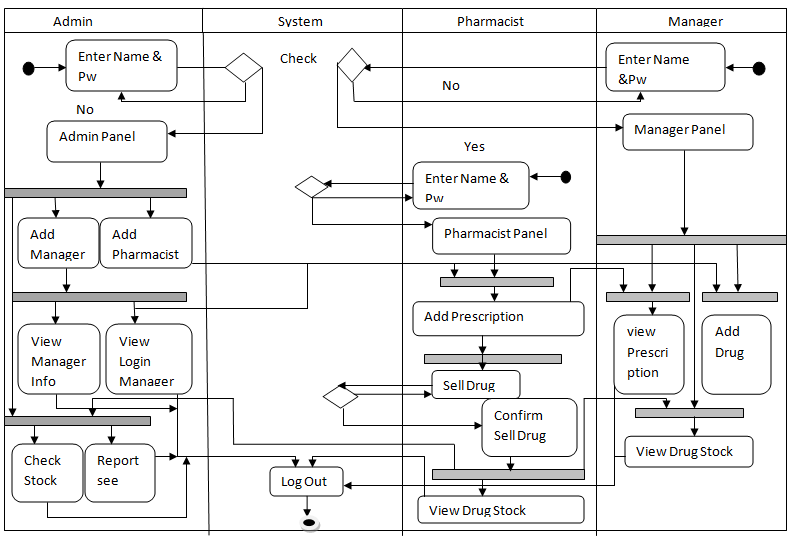
**Activity For Pharmacist:**



**Fig 3.6: Activity For Pharmacist**

**3.3 Swim lane Diagram:**

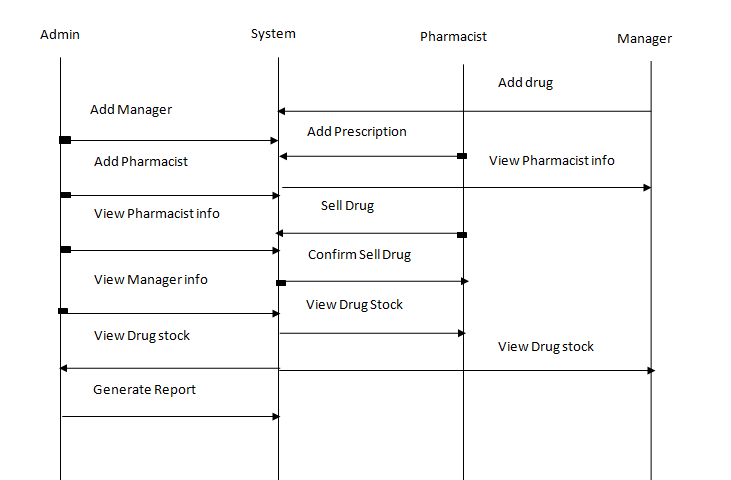
Swim lane diagrams are a unique form of flow chart that makes visualizing and analyzing business processes easier. The uniqueness around these diagrams is that they are setup and put in various lanes making the process and flow chart clearer.

****

**Fig 3.7: Swim Lane Diagram**

**3.4 Sequence Diagram:**

A sequence diagram is also known as a timing diagram, event diagram and event scenario. Object interactions usually begin at the top of a diagram and end at the bottom. In a sequence diagram, object interaction occurs through messages on the vertical and horizontal dimensions and are designated by horizontal arrows and message names



**Fig 3.7: Sequence Diagram**

## Chapter: 4

**Project planning and Scheduling**

**4.1: Function Proposed System:**

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
| Login into the System | F1 |
| Add Pharmacist | F2 |
| Add Manager | F3 |
| View Stock | F4 |
| View Pharmacist info | F5 |
| View Manager info | F6 |
| Add Medicine | F7 |
| Add patient | F8 |
| Calculate medicine Cost | F9 |
| Generate report | F10 |

**F1 Login functions:**

Input: User Name, Password, Type

Output: Login Successful, Login Failed

**F2 Add Pharmacist:**

Input: First Name , Last Name, Phn, Email, Salary,Username, Address, Password

Output: Success

Use table of the database:. First Name , Last Name, Phn, Email, Salary,Username, Address, Password

**F3 Add Manager:**

Input: First Name , Last Name, Phn, Email, Salary,Username, Address, Password

Output: Success/Error.

Use table of the database: :. First Name , Last Name, Phn, Email, Salary, Username, Address, Password

**F4 View Pharmacist info**

Output: First Name , Last Name, Phone, Email, Salary, Username, Address, Password

Use table of the database: Pharmacist list.

**F5 View Pharmacist info**

Output: First Name , Last Name, Phone, Email, Salary, Username, Address, Password

Use table of the database: Manager list.

**F6 Update Stock**

Input: Select medicine then can add amount of that.

Output: Success/Error.

Use table of the database: Sell, user.

**F7 Add Medicine :**

Input: Medicine name, category,type company then add.

Output: Success/Error.

Use table of the database: Medicine stock

**F8 Add patient information:**

Input: Patient Name, Age, Gender, Phone,month

Output: Success/Error.

Use table of the database: Notification.

**F9 Calculate medicine cost :**

Input: Search by then added them & put quantity

Output: Name , quality automatically price show

Use table of the database: Drug

**F10 Report See:**

**Input:** Selecting month

**Output:** Automatically all calculation show

**4.2: Function Point Analysis**

Function Oriented metrics are derived using an empirical relationship based on countable (direct) measures of software’s information domain and assessments of software complexity. Function points computed by comparing five information domain characteristics. The information domain values are as follows:

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

**Number of user outputs** – Each user output that provides application-oriented information to the user is counted.

**Number of user inquiries** – An inquiry defined as an on-line input that results in the generation of some immediate software response in the form of an on-line output. Each distinct inquiry counted.

**Number of files** – Each logical master file counted.

**Numbers of external interfaces** – All machine-readable interfaces that used to transmit information to another system counted.

**Unadjusted function Point Contribution**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Transaction Function | FTRs | DETs | Complexity | UFP |
| Search Medicine (EQ) | **1** | **4** | **Low** | **3** |
| See details (EQ) | **1** | **4** | **Low** | **3** |
| List medicine (EQ) | **1** | **4** | **Low** | **3** |
| Add to prescription (EI) | **2** | **8** | **average** | **6** |
| Checkout (EO) | **1** | **7** | **Low** | **4** |
| Show the report(EO) | **1** | **9** | **Low** | **3** |
| Total |  |  |  | **22** |

**Unadjusted function Point Contribution**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Function | RETs | DETs | Complexity | UFP |
| Medicine(ILF) | **1** | **2** | **low** | 7 |
| Medicine added (ILF) | **1** | **5** | **low** | 7 |
| Medicine quantity (EIF) | **2** | **8** | **low** | 5 |
| patient(ILF) | **1** | **2** | **low** | 7 |
| Total |  |  |  | 26 |

Rating is given out of five:

|  |  |
| --- | --- |
| System characteristics | Rating |
| Data communication | 2 |
| Distributed and data processing | 2 |
| Performance | 3 |
| Heavily | 3 |
| Transaction rate | 0 |
| Online data entry | 2 |
| End user | 0 |
| Data update | 4 |
| Complex processing | 3 |
| Reusability | 3 |
| Installation case | 4 |
| Operation | 3 |
| Facility change | 4 |
| Total | 33 |
|  |  |

Value Adjustment Factor (VAF) = (0.65 + (0.01 \* TDI))

= (0.65 + (0.01 \* 33))

= 0.99 ≈1

UFP= UFP(Transaction fn)+UFP(data fn)

=22+26

=48

Adjusted function point count = UFP\*VAF

=48\*1

Efforts for PHP = AFP \* Productivity

=48\*15.5

= 744 person hour/8 hours

=93 person days/3 pesron

= 31 days

= 1 month

|  |  |
| --- | --- |
| **4.3** | **Project Scheduling** |

Project scheduling is an activity of distributing the estimated efforts within the planned project duration.

**4.4 Project Schedule Chart:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Activity Name | W1 | W2 | W3 | W4 | W5 | W6 | W7 | W8 | W9 | W10 | | W11 | W12 |
| Analysis & Design |  |  |  |  |  |  |  |  |  |  | |  |  |
| Risk |  |  |  |  |  |  |  |  |  |  | |  |  |
| Coding |  |  |  |  |  |  |  |  |  |  | |  |  |
| Testing |  |  |  |  |  |  |  |  |  |  |  |  |  |

**4.4 Personnel Requirements’ Chart**

To complete the total project three persons will be required. They are

1. System Analyst ------------------------------ 1 Month

2. Coder----------------------------------------- 2 Month

3 Tester -----------------------------------------1 Montth

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Month1 | Month2 | Month3 | Month4 |
| System Analyst |  |  |  |  |
| Coder |  |  |  |  |
| Tester |  |  |  |  |

**4.5: Resource Requirement Chart:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **Month 1** | **Month 2** | **Month 3** | **Month 4** |
| **Business Modeling** | Pencil , Pad, Calculator |  |  |  |
| **Data Modeling** | Windows 7, MySQL, Php, Xampp |  |  |  |
| **Process Modeling** |  |  |  |  |
| **Application generation** |  |  | By different way of testing |  |
| **Testing** |  |  |  |  |

|  |  |
| --- | --- |
|  | **4.7 Process Based Estimation** |

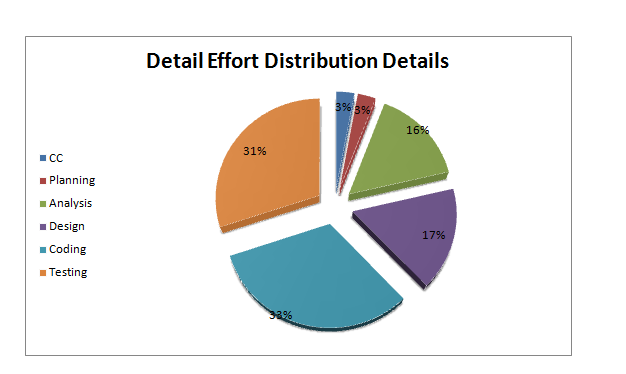
In process-based estimation, process is decomposed into a relatively small set of tasks and the effort required to accomplish each task is estimated. Process based estimation begins with a delineation of software functions obtained from the project scope. A series of software process activities must be performed for each function.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Activity | CC | Planning | Engineering | | Construction | | CE | Total |
| Function |  |  | Analysis | Design | Code | Test | n/a |  |
| F1 |  |  | 0.50 | 1 | 1 | 0.55 | n/a | 3.05 |
| F2 |  |  | 1.25 | 1 | 1 | 0.50 | n/a | 3.50 |
| F3 |  |  | 1 | 1 | 1.50 | 0.50 | n/a | 4.00 |
| F4 |  |  | 1.25 | 1 | 1.50 | 0.50 | n/a | 4.00 |
| F5 |  |  | 1 | 1 | 0.50 | 0.25 | n/a | 2.75 |
| F6 |  |  | 1 | 1 | 1.50 | 0.50 | n/a | 4.00 |
| F7 |  |  | 1 | 1 | 1 | 0.50 | n/a | 3.50 |
| F8 | 0.50 | 0.50 | 1 | 1 | 1 | 0.50 | n/a | 3.50 |
| F9 |  |  | 1 | 1 | 0.50 | 0.50 | n/a | 3.00 |
| F10 | 0.50 | 0.50 | 1 | 0.50 | 1 | 0.25 |  | 4.00 |
| Total | 1 | 1 | 10 | 9.50 | 10.5 | 4.55 |  | 32.3 |
| Effort | 3% | 3% | 16% | 17% | 33% | 31% |  |  |
|  |  |  |  |  |  |  |  |  |

**4.8: Effort Distribution**

The project estimation technique leads to estimates of work units required to complete the software development. In this project, 32.05% of full software development has been allocated to analysis and design, 40% has allocated to coding and the remaining 27% is allocated to software testing and support.

**4.9 Effort Distribution in Details:**



## Chapter: 5

**Project Management**

**5.1 : Personnel Cost Estimation:**

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 = 8 hours

Organization working hours per month=24.083\*8= 192.66 hours

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **No. of Members** | **Months** | **Salary** |
| System Analyst | 1 | 1 | 20,000.00 |
| Senior Developer | 1 | 1 | 18,000.00 |
| Risk Analyzer | 1 | 1 | 15,000.00 |
| Tester | 1 | 1 | 12,000.00 |
| **Total** | | | **60,000.00** |

**5.2 Hardware cost:**

The percentage of year is: 1/15 = 6.67% = 0.0667

The depreciation cost of computer is = (50,000 \* 0.0667) = 3,335.00

Two person computer cost= (3,335.00 \* 2) = 6670

The depreciation cost of scanner is = (2000 \* 0.0667) = 134.40

Two person scanner cost= (134.40\* 2) = 268.8

The depreciation cost of printer is = (4000 \* 0.0667) = 266.80

Two person printer cost= (134.40\* 2) = 266.80 \* 2 = 533.60

**5.3 Software Cost:**

|  |  |
| --- | --- |
| Microsoft Office 2010 | 160.00 |
| Microsoft Windows 7 | 200.00 |
| Xampp Sarver | 110.00 |

**5.4 Other Cost:**

|  |  |
| --- | --- |
| Furniture | 6,000.00 |
| Electricity bill | 600.00 |
| House Rent | 4,000.00 |
| Vehicle Rent | 500.00 |
| Extra | 1,000.00 |

**5.5 Account Table:**

|  |  |
| --- | --- |
| Particulars | TK |
| Salary-   * System Analyst * Senior Developer * Designer * Risk Analyzer * Tester | 20,000.00  18,000.00  15,000.00  2,000.00  12,000.00 |
|  | 60,000.00 /= |
| Hardware Cost –   * Computer * Scanner * Printer | 6070.00  268.8  533.60 |
|  | 7472.40 /= |
| Other Costs-   * Furniture * House Rent * Electricity bill * Vehicle Rent * Extra | 7,000.00  4,000.00  600.00  500.00  1,000.00 |
|  | 13,100.00 /= |
| **Total cost** | **87,572.40 /=** |

## Chapter: 6

**Risk Analysis**

**6.1 Risk Engineering:**

A risk is a serious problem that can occur in any time. So developer needs to awake always about it. It is necessary to analyze the risks in a project. Risk engineering is a most important part in software management, because if the risks are not well analyzed many problem can happen. Risk analysis and management have some steps that help the software developer to upcoming uncertain risk.

**6.2 Types of risks:**

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

1. Software Risk

2. Technological Risk

3. Business Risk

**Software Risks:**

These risks can hamper 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.

**Technological Risk:**

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.

**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 Building a system whose business needs have been changed.
* Management risks: Losing the support of senior management due to a change in focus or a change in people

**6.3 Risk Identification**

Risk Identification is a step by step process to specify risks to the project plan. By identifying known and predictable risks, it is possible to avoid or mitigate the identified risks.

The identified risks of different categories of this software project are given below:

* Size estimate may be significantly low
* Larger number of user than planned
* Unavailability of the development software
* Unavailability of the required hardware
* Customer will change the requirement
* Stuff inexperience
* Users resist the new system

After the projection of risks, a strategy for dealing with risk has to be developed. An effective strategy must consider three uses –

* risk monitoring
* risk management and contingency planning

1. **Risk mitigation:** Risk mitigation is a problem avoidance activity. As a proactive risk strategy has been chosen in this project, it is better to avoid risks.
2. **Risk management and contingency planning:** assumes that mitigation efforts have failed and that the risk has become a reality.

Mitigation, monitoring and management plan for each of the risks which lie above the cutoff line are described in the RMMM plan section of this report.

**6.4 Risk Monitoring**

* We can fix our technological risks by choose the perfect hardware and software combination that can be more helpful to avoid from risk to develop the system.
* People risks can be reduce by increasing consciousness among the group members.

Risks that derive from changes to the customer requirements are belongs to this requirements risks, so it can be monitoring by keep these requirements deeply in minds

## Chapter: 7

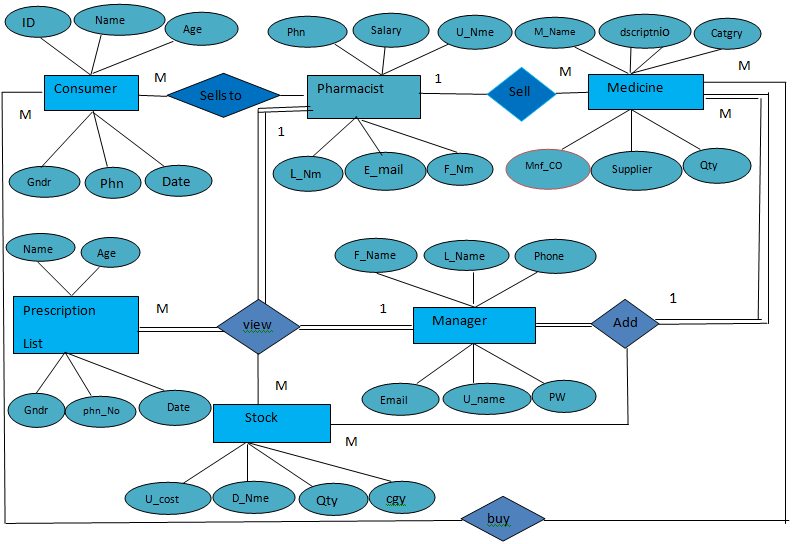
**Design**

**7.1 Entity Relationship Diagram:**

An entity-relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system’s entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure.

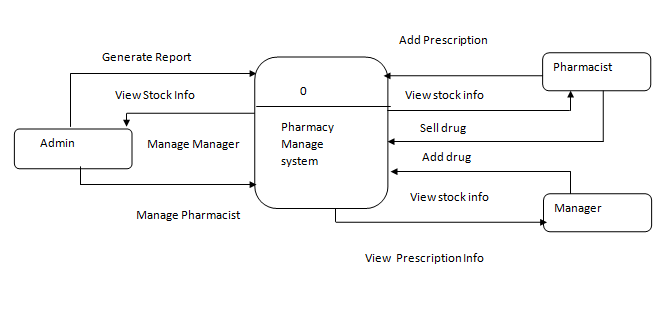
The elements of an ERD are:

* Entities
* Relationships
* Attributes



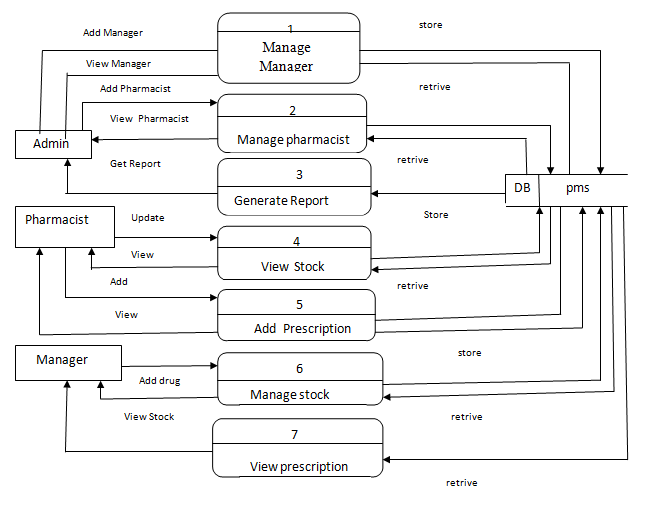
**Fig 7.1: Data Flow Diagram**

**DFD Context Level:**

****

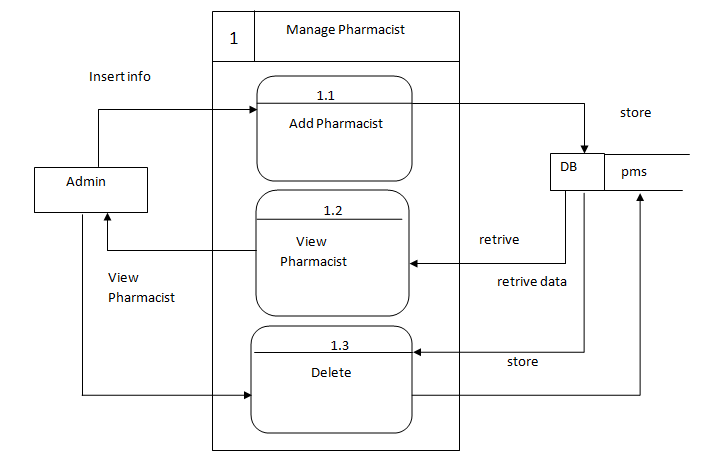
**Fig 7.2: DFD Context Level**

**DFD Levl 1 :**

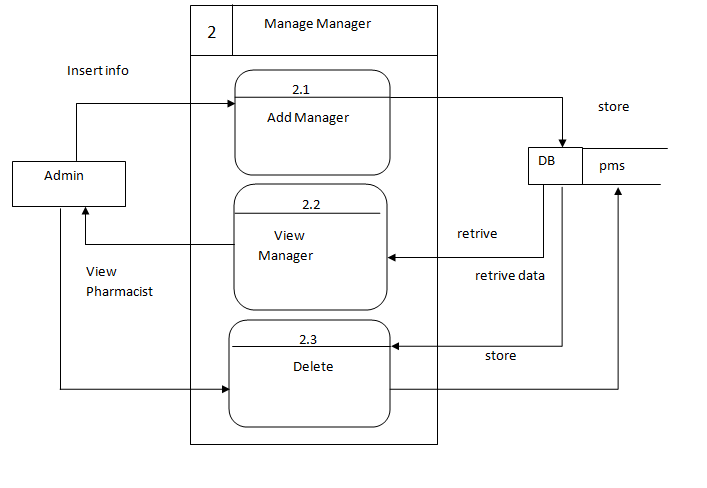
****

**Fig 7.3: DFD Level 2 Process 1**

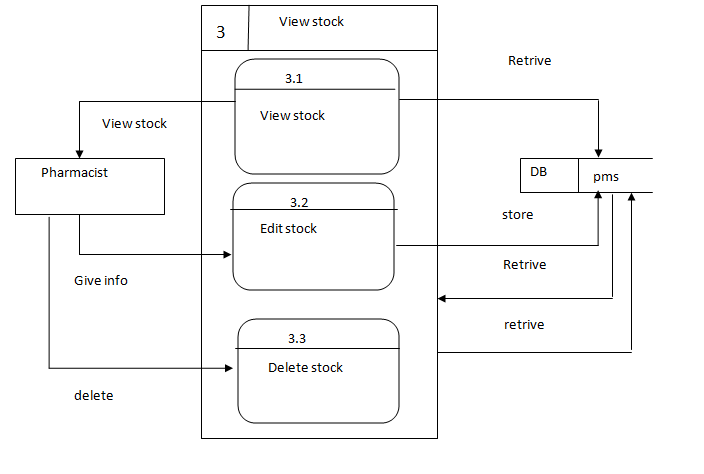
**Level 2 process 1:**

****

**Fig 7.4: DFD Level 2 Process 1**

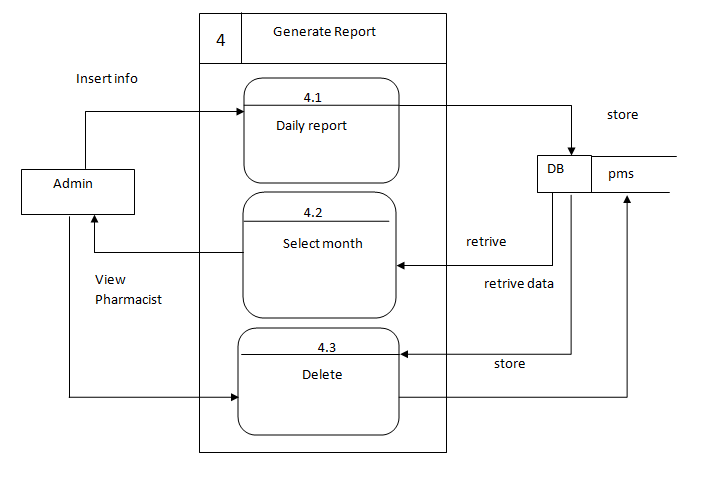
**Level 2 process 2: **

**Fig 7.5: DFD Level 2 Process 2**

**Level 2 Process 3 : **

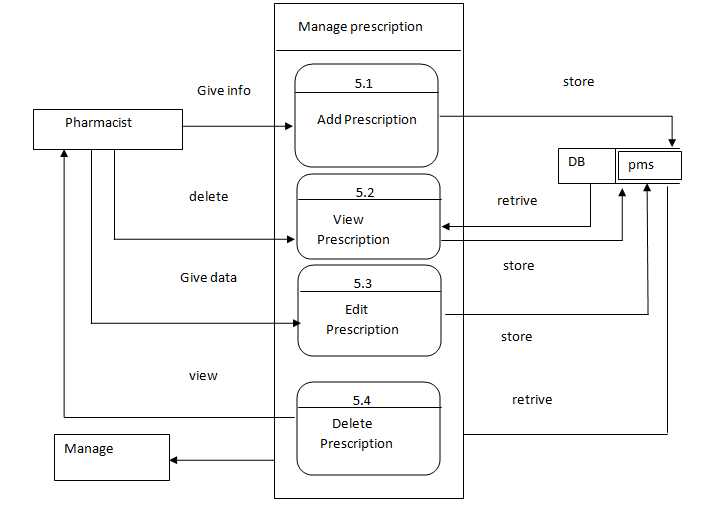
**Fig 7.6: DFD Level 2 Process 3**

**Level 2 process: 4**



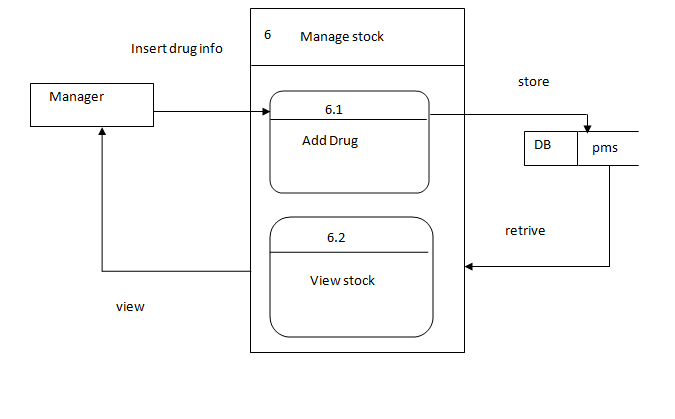
**Fig 7.7: Level 2 Process 4**

**Level 2 process 5:**



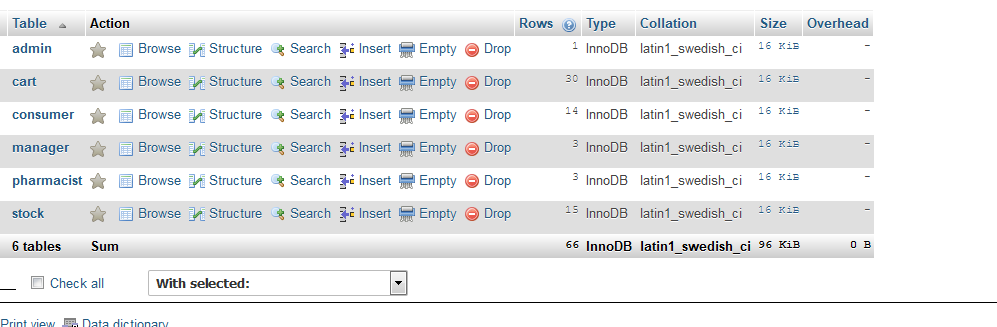
**Fig 7.8: DFD Level 2 Process 5**

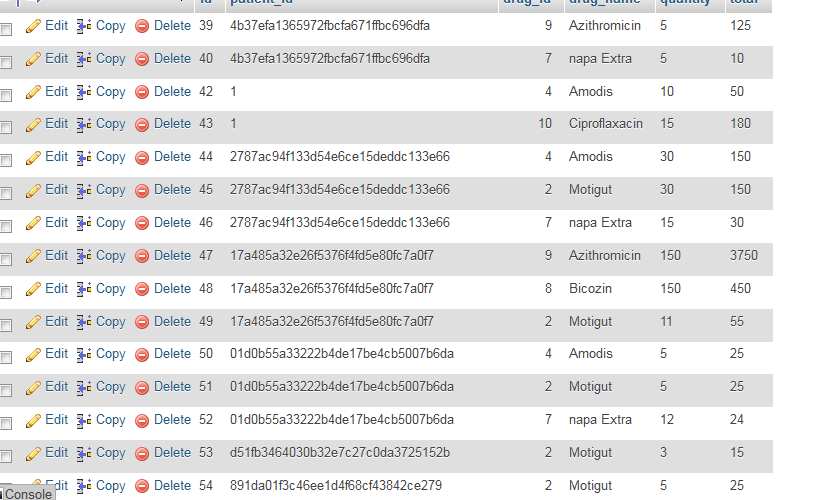
**Level 2 process 6:**

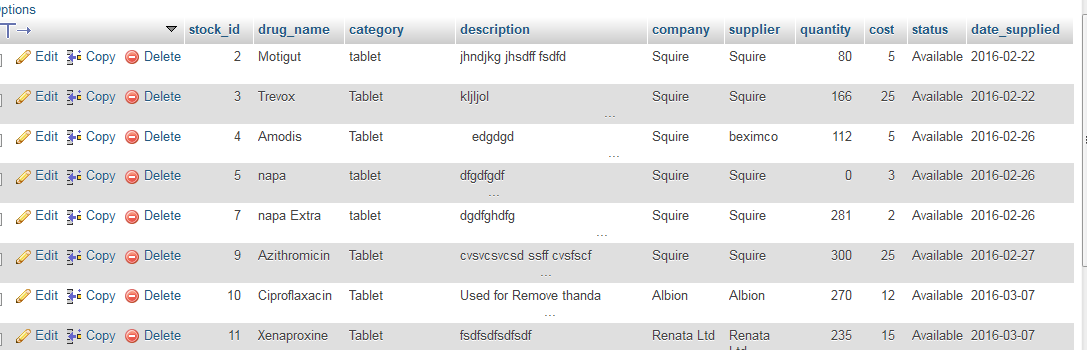
****

**Fig 7.9: DFD Level 2 Process 6**

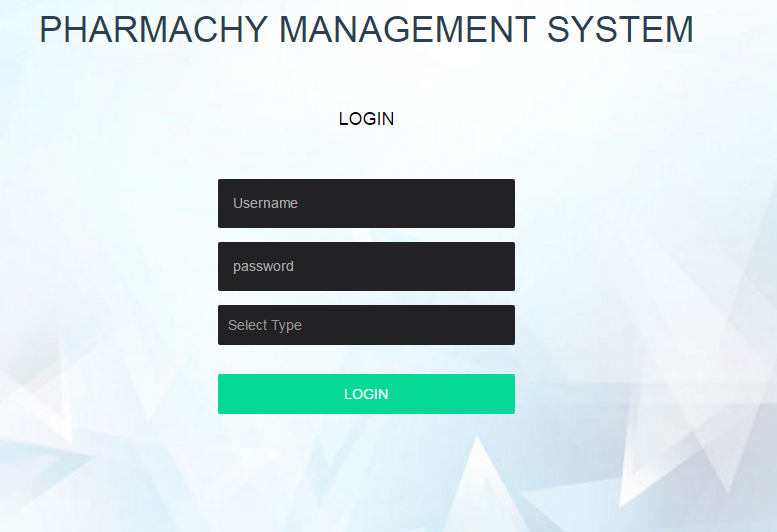
**7.3 Database Design Field :**

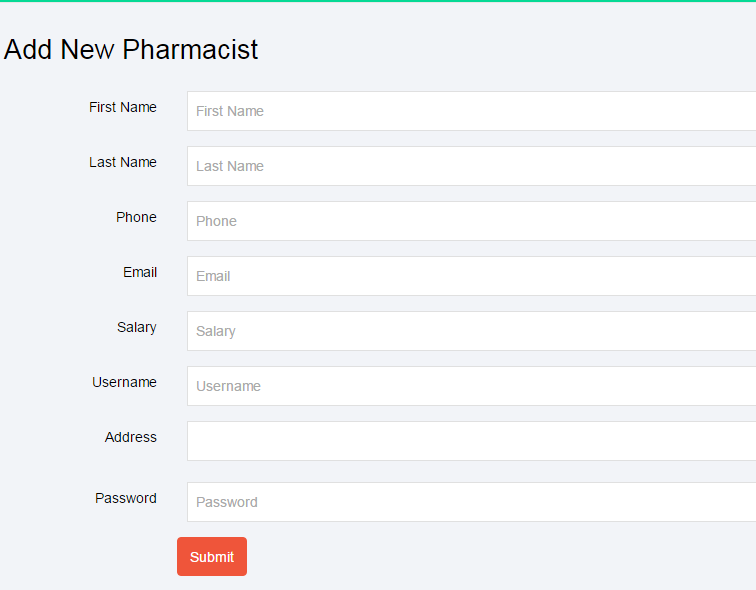


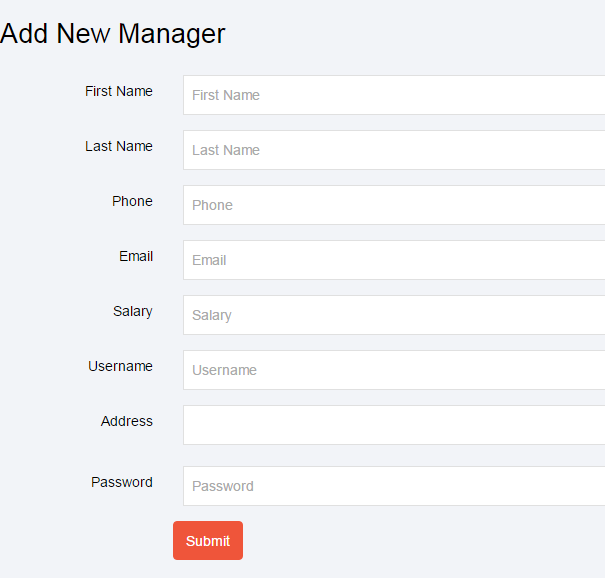


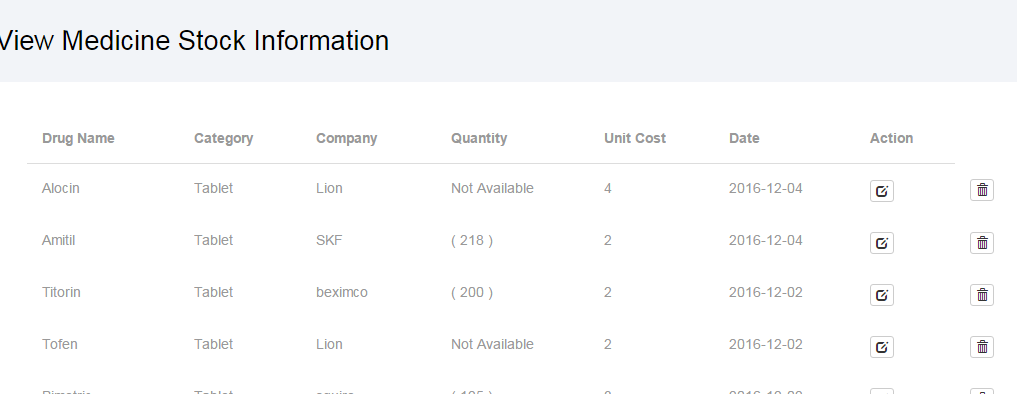


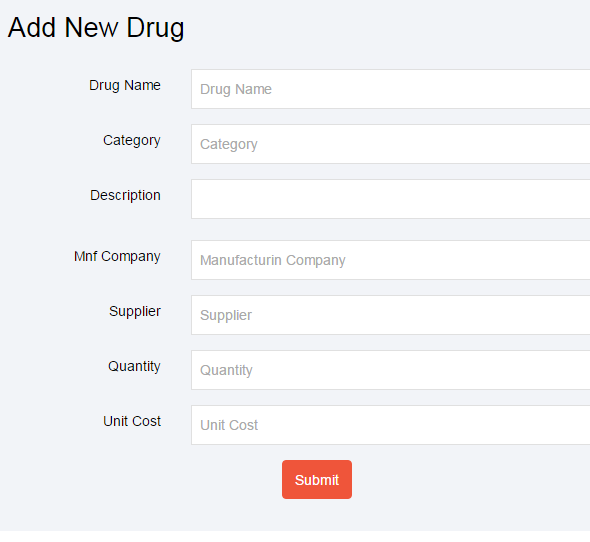
**7.4 Interface Design: Login Page**

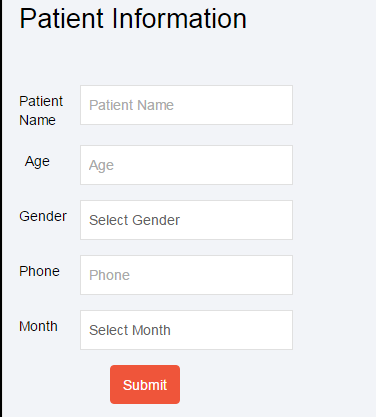


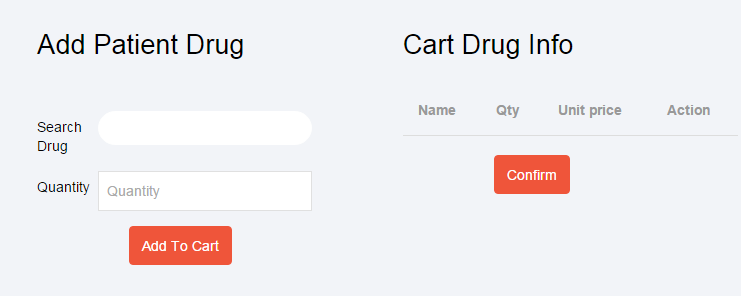










****

## Chapter: 8

## Testing

**8.1 System Quality Management**

Agreement on quality requirements, as well as clear communication to the software engineer on what constitutes quality, require that the many aspects of quality formally defined and discussed.

A software engineer should understand the underlying meaning of quality concepts and characteristics and their value of the software underdevelopment or maintenance.

We incorporate quality assurance and testing throughout our whole development process. Our experts utilize various quality assessment automation tools to ensure unmatched software quality.

**8.2 System Quality Management Process**

System quality management process involves,

1. Define the required product in term of its quality characteristics.

2. Planning the processes to achieve the required product.

**8.3 System Testing**

Testing means that one inspects behaviour of a program on a finite set of test cases like a set of inputs, execution preconditions, and expected outcomes developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement, for which valued inputs always exist. In practice, the whole set of test cases is considered as infinite, therefore theoretically there are too many test cases even for the simplest programs. So, this is considered that a program can be compatible with a limited set of test case.

There are many testing method. Among those we followed few simple testing methods. Those are:

1. Black-box testing

2. White-box testing

**Black-Box Testing:** In this testing the system is treated as a "black box", examining functionality without any knowledge of source code or any other knowledge how the system works. The testers are only aware of what the software is supposed to do. If the system gives the desired output for predefined input then the system is okay.

**White-Box Testing:** In this approach tester must have clear understanding on coding, software structure and all other aspect of the development factors. For this project the software developer played the role of black box tester. Here the test case for with the black-box testing failed; we took those cases as test case for white-box testing.

**Testing Design:**

|  |  |
| --- | --- |
| Testing scenario No:1 | |
| Scenario | Admin Login testing scenario of this system |
| Input’s | Username, password of admin for Login |
| Desired Output’s | When enter username, password then get successful level define. |
| Actual Output’s | For login our system work correctly |
| Verdict | Getting result from Desired Output’s and Actual Output’s decided this system is successful for login. |

|  |  |
| --- | --- |
| Testing scenario No:2 | |
| Scenario | Admin can add Pharmacist |
| Input’s | Pharmacist basic info for registration |
| Desired Output’s | When enter all basic info correctly, Pharmacist will be registered in the system. |
| Actual Output’s | For Pharmacist registration our system work correctly |
| Verdict | Getting result from Desired Output’s and Actual Output’s decided this system is successful for Pharmacist’s registration. |

|  |  |
| --- | --- |
| Testing scenario No:3 | |
| Scenario | Admin can edit Pharmacist’s details. |
| Input’s | Pharmacist basic info update by the admin |
| Desired Output’s | When changed basic information for the Pharmacist by the admin. |
| Actual Output’s | For update Pharmacist info our system work correctly |
| Verdict | Getting result from Desired Output’s and Actual Output’s decided this system is successful for update Pharmacist details. |

|  |  |
| --- | --- |
| Testing scenario No:4 | |
| Scenario | Admin can add Manager. |
| Input’s | Manager basic info for registration |
| Desired Output’s | When enter all basic info correctly, Manager will be registered in the system. |
| Actual Output’s | For Manager registration our system work correctly |
| Verdict | Getting result from Desired Output’s and Actual Output’s decided this system is successful for Manager’s registration. |

|  |  |
| --- | --- |
| Testing scenario No:5 | |
| Scenario | Admin can edit Manager’s details. |
| Input’s | Manager basic info update by the admin |
| Desired Output’s | When changed basic information for the Manager by the admin. |
| Actual Output’s | For update Manager info our system work correctly |
| Verdict | Getting result from Desired Output’s and Actual Output’s decided this system is successful for update Manager details. |

|  |  |
| --- | --- |
| Testing scenario No:6 | |
| Scenario | Admin can remove Manager’s |
| Input’s | Delete Manager from our database. |
| Desired Output’s | When admin want he/she can remove an Manager. |
| Actual Output’s | I check this process and get actual outputs |
| Verdict | The process is worked correctly and successfully. |

|  |  |
| --- | --- |
| Testing scenario No:7 | |
| Scenario | Take daily monthly reports and calculate profit. |
| Input’s | Admin give whole month all cost and incomes. |
| Desired Output’s | Systems take the information and calculate monthly profit and print the profit. |
| Actual Output’s | I check this process and get actual outputs |
| Verdict | The process is worked correctly and successfully. |

|  |  |
| --- | --- |
| Testing scenario No:8 | |
| Scenario | System adds medicine in stock product. |
| Input’s | Manager add medicine with all information. |
| Desired Output’s | System takes all information display the product in stocks. |
| Actual Output’s | I check this process and get actual outputs |
| Verdict | The process is worked correctly and successfully. |

## Chapter: 9

## Conclusion

**Conclusion**

The biggest experience working in this course is indeed being a part of designing and Implementing software. Here, most experience was round the designing issue. I have learnt a lot Of new things which was so much unknown to us. I have also learnt some technical issues which help to do better in future life. The following indicator will indicate some of our technical issue Which We have learnt and implemented from this project.we strongly believe that the proposed software **Pharmacy Management System** will must generate a large amount of traffic in global market place and we have a well organized team to build that software within schedule time frame. Implementation is the state in the software where the theoretical design is turned into a working system. By this, the users get the confidence that the system will work effectively. The system can be implemented only after through testing. Implementation walkthroughs ensure that the completed system actually solves the original problem. This walkthrough occurs just before the system goes into use, and it should include careful review of all manuals, training materials and system documentation. Again, users, the analyst and the members of the computer services staff may attend this meeting

## Chapter: 10

## Bibliography

**10.1 Books**

[1] Pressman, Roger S. *Software Engineering: A Practitioner’s Approach*. 5th ed. Boston: McGraw Hill, 2004.

[2] Kendall, E. Kendall. *System Analysis and Design.*7th New Delhi: Prentice Hall.