**TABLE OF CONTENTS**

**CHAPTER 1 - Software Project Management Plan (SPMP)**

|  |
| --- |
| **Table of Contents                                                                     Page No.** |
| **1.      INTRODUCTION**  **1.1                Project Overview**  **1.2                Project Deliverables**  **2.      PROJECT ORGANIZATION**  **2.1            Software Process Model**  **2.2                Roles and Responsibilities**  **2.3                Tools and Techniques**  **3.      PROJECT MANAGEMENT PLAN**  **3.1                Tasks**  **3.2                Information Gathering**  **3.3                Resources Needed**  **3.4                Dependencies and Constraints**  **3.5                Risks and Contingencies**  **4.      PROJECT SCHEDULING**  **4.1                 Gantt Chart** |

**CHAPTER 2 - Software Requirements Specifications (SRS)**

|  |
| --- |
| **Table of Contents                                                  Page No.** |
| **1.      INTRODUCTION**  **1.1                Product Overview**  **2.      SPECIFIC REQUIREMENTS**  **2.1                External Interface Requirements**  **2.1.1                         User Interfaces**  **2.1.2                         Software Interfaces**  **2.1.3                         Communications   Protocols**  **2.2                Software Functional Requirements**  **2.3                Software Non-Functional Requirements**  **2.3.1                         Reliability**  **2.3.2                         Availability**  **2.3.3                         Security**  **2.3.4                         Maintainability**  **2.3.5                         Portability**  **2.3.6                         Performance** |
|  |

**CHAPTER 3 - Software Design Description (SDD)**

|  |
| --- |
| **Table of Contents                                                                       Page No.** |
| **1        INTRODUCTION**  **1.1                Design Overview**  **1.2.1                         Data Flow Diagram**  **1.2.2                         Use Case Diagram**  **1.2.3                            Class Diagram**  **1.2.4                           Activity diagram**  **1.2.5                       Sequence Diagram**  **1.3                Database Design**  **1.3.1                         Entity Relationship Diagram**  **1.3.2                         Tables Structure**  **1.3.3                             Normalization Details**  **1.3.4                            Data Dictionary**  **2       SYSTEM ARCHITECTURAL DESIGN**  **2.1                Chosen System Architecture**  **3       DETAILED DESCRIPTION OF COMPONENTS**  **3.1       Component 1**  **-**  **-**  **-**  **3.nComponent n**  **4       USER INTERFACE DESIGN**  **4.1                Description of the User Interface**  **4.1.1             Screen Images** |
|  |

**\**

**CHAPTER4 - Software Test Documentation (STD)**

|  |
| --- |
| **Table of Content                                                                               PageNo.** |
| **1       INTRODUCTION**  **1.1                System Overview**  **1.2                Test Approach**  **1.3                Testing Objectives**  **2.      TEST PLAN**  **2.1                Features to be tested**  **2.2                Features not to be tested**  **3       TEST CASES**  **3.1 Unit Testing**  **3.2       Functional Testing**  **3.3                System Testing**  **3.4                Integration Testing**  **3.5                Validation Testing**  **3.6                Top down integration**  **3.7 Bottom up integration**  **4          Conclusion &Future Scope**  **APPENDIX**  **Bibliography**  **References** |
|  |

1. INTRODUCTION

Medical emergencies can happen anywhere at any time. Not always do you find a chemist store in your nearby locality. Also there's a need for a medical consultancy which allows us to have a doctor in every which situation we want.

Meaning we get to consult a doctor before making an actual appointment with a doctor.

E-Commerce website for pharmaceutical drugs and health care products would help us to have medicines and health care at our door at anywhere. It is a helpful solution to the problem faced by senior citizens living on their own.

Along with that we intend to save the money wasted to make an appointment with a doctor when you don't actually need it.

* 1. **METHODOLGY**

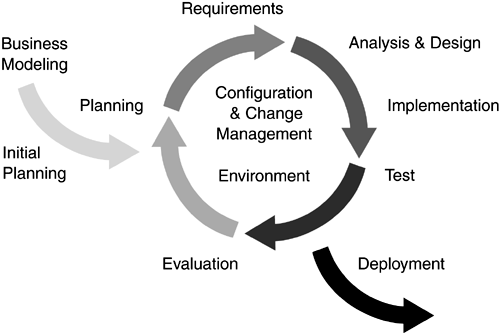


Fig 1.1 Rational Unified Process

RUP is a software development process from Rational, a division of IBM.

It divides the development process into four distinct phases that each involve business   modeling, analysis and design, implementation, testing, and deployment

**1.2 PURPOSE**

The purpose of this software requirements specification is to verify that all the specifications are correct and are verified. This document also serves to ensure that the software is traceable throughout its software development life cycle.

Developing E-Commerce website for medicines will allow people to order medicines from their place and have it anywhere they want . It integrates the benefits of an online medical shopping site with the feeling of a virtual dispensary aura.  It will usher in the existing offline way of buying drugs with the perfect blend of synchronous and asynchronous interaction. It encapsulates a medical help and a medical store.

**1.3 SCOPE**

The scope of this project is spread across the city and can be used by any individual.  Be it a boy aged 11 or a old aged woman In the 80s; consultation can be taken by individual irrespective of his qualifications. This can be used from any place in the city. Further development into the current system can lead to a widespread, even nationwide use of web consultation. Some features are-

**1.3.1** Facility to shop drugs online**.**

**1.3.2** Facility to have drugs at customized rates.

**1.3.3** Facility to know what type of specialist you need.

**1.3.4** Facility to know which is the best doctor for a particular specialization.

**1.3.5** Facility to know your doctor.

1.3.6 Facility to know whether you actually need a doctor.

**1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS**

**1.4.1 Administrator:** Administrator is the website owner who can add and remove medicines and keep track of the activities happening on the website. Administrator also keep updated records of the doctors.

**1.4.2 User:** User is the customer who shops for drugs on the website. User is also the individual looking for medical help.

**1.4.3 Actor:** An actor is someone or something outside the system that interacts with the system (e.g. an end-user or system administrator might be actors or another system with which the system interacts might also be an actor).

**1.4.4 Attribute:** In data modeling, specific items of data that can be collected for a class. Any software performing either the final stage in a process, or a task not apparent to the user. A common usage is a verification of permissions based on user-class and security. The end-user accesses the application and pages are retrieved based on user-class permissions.

**1.4.5 Client-server:** A common form of distributed system in which software is split between server tasks and client tasks. A client sends requests to a server asking for information or action and the server responds.

**1.4.6 End User:** The ultimate consumer of a product, especially the one for whom the    product has been designed. End-users for a patient management application include physicians, nurses, epidemiologists, outreach workers and health care providers. The one who is in need of medicines and can not walk to the chemist, the one who needs consultation from a doctor.

**1.4.7  Functional Requirement:** A description of what a system should be able to do–a function it should perform.

**1.4.8 Non-functional Requirement:** Software design requirements related to the efficiency, reliability, portability, and usability of the system. Also known as supplementary requirement.

**1.4.9 Requirement:** A requirement describes a condition or capability to which a system must conform; either derived directly from user needs, or stated in a contract, standard, specification, or other formally imposed document. A requirement is a desired feature, property, or function to be met by the application.

**1.4.10 Software Requirement:** A software requirement is a specification of an externally observable behavior of the system; for example, inputs to the system, outputs from the system, functions of the system, attributes of the system, or attributes of the system environment.

**1.4.11 Software Requirement Specification:** A project artifact that defines the complete system requirements through use cases and supplementary specifications.

**1.4.12 Use-case Specification:** The text explanation or description of a use case.

**1.4.13 User Interface:** The programming that controls a display for the user (usually on a computer monitor) and that allows the end user to interact with the system with commands and mechanisms to control system operation and input data.

**1.4.14 JAR file:** Deployed version of the project.

**1.4.15 System DSN:** Data source stores information about how to connect to the indicated data provider. A user data source can be used on the current machine.

**1.5 Tools Used**

Application architecture- Java ,J2EE

* Java

Java is an object oriented programming language developed by Sun Microsystems, a company best known for its unix end UNIX workstations. Java language was designed to be small, simple and portable across platforms, both at the source and the binary level.

* J2EE

Java platform, enterprise edition or java EE is a widely used platform for server programming in the programming language. The Java platform differs from the java Standard edition platform in that it adds libraries which provide functionality to deploy distributed multi-tier Java software.

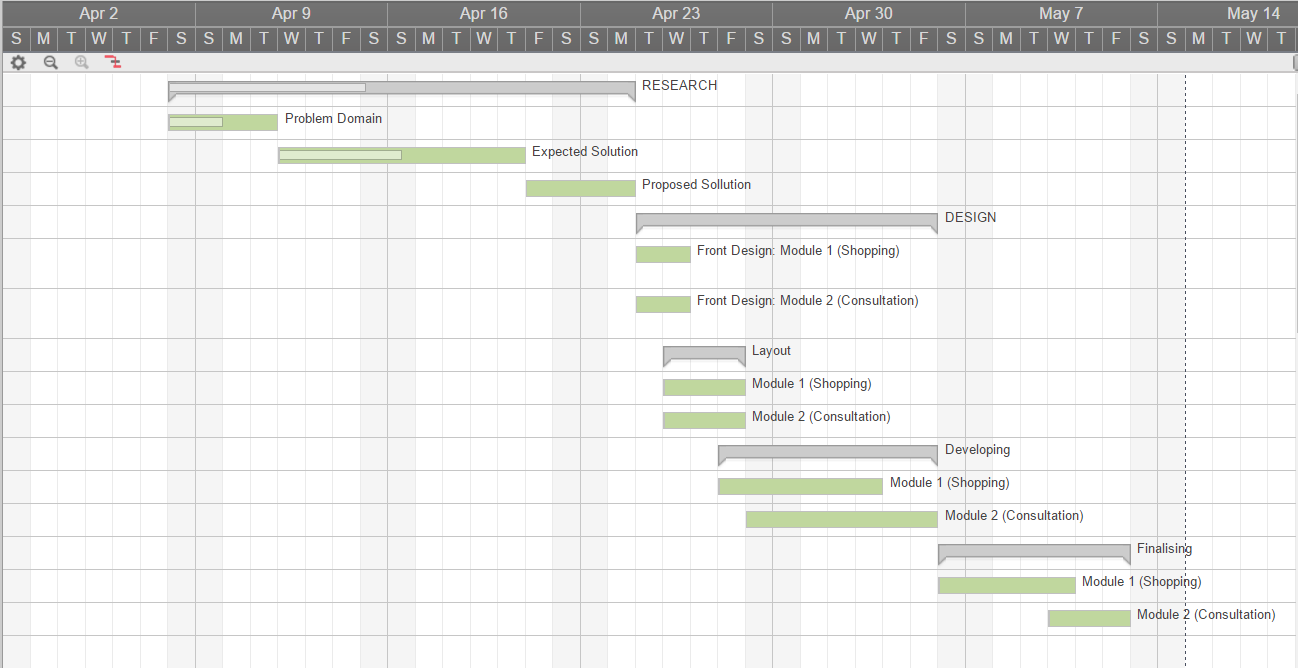
* web server

web server is a free , certified java EE 5 server for building and managing Java applications. It is IBM supported distribution GERONIMO that uses Apache tomcat for servlets container and Axis2 for web services.

* database platform db2

it is the database management system that delivers a flexible and cost effective database platform to build robust on demand business applications.

**1.6 Gantt chart**

****

**Fig 1.2 Gantt Chart**

**1.6 OVERVIEW**

The remaining SRS will include 5 sections: -

**1.6.1** Overall Description will describe major components of the system, interconnection and external interfaces.

**1.6.2** Specific Requirements will describe the functions of actors, their role in the system and constraints.

**1.6.3** Change Management Process will identify the change management process to be used to identify, log, evaluate, and update the SRS to reflect changes in project scope and requirements.

**1.6.4** Document Approvals will identify the approvers of the SRS document. Approver name, signature, and date should be used.

**1.6.5** Supporting Information -The supporting information makes the SRS easier to use.

It includes:

**1.6.6.1**Table of Contents

**1.6.6.2** Index

**1.6.6.3** Appendices

**2.  SPECIFIC REQUIREMENTS**

**2.1 PRODUCT PERSPECTIVE**

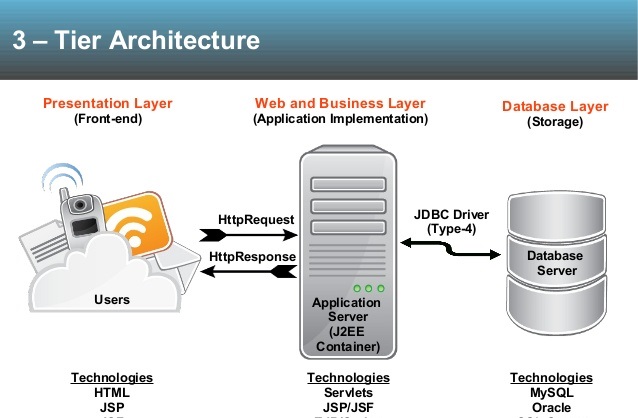
****

Fig 2.1 System architecture

**2.2 EXTERNAL INTERFACES**

The specific Requirements for the project is to provide users the platform for easy and secure facility. Some of these Requirements are as follows:-

The software first of all authenticates the user (Administrator, normal customer)

and then provides list of task that can be performed and after proper selection corresponding information is displayed. If the selection is that of manipulation then appropriate fields will be displayed and after filling the updated information, changes are reflected.

To take input from the administrator about the access rules. The access rules would be defined in form of source and destination IP addresses, source and destination ports, type of the protocol.

To capture the information and data.

To provide User Authentication, User Profile, Data Filter, Command filter, Session Time, Session id.

To filter the information and data on the basis of rule set by administrator.

To add and delete the already existing rules.

To build simple to understand and attractive interface.

**2.1.1 User Interfaces**

Since this is a web based application this the interface should be user friendly. Navigation, data entry, fields selection must be easy. A decent comprehensible articulate appearance and easy navigation is what a user needs. User of the system is provided graphical user interface.

Various interfaces for the product could be

* Homepage

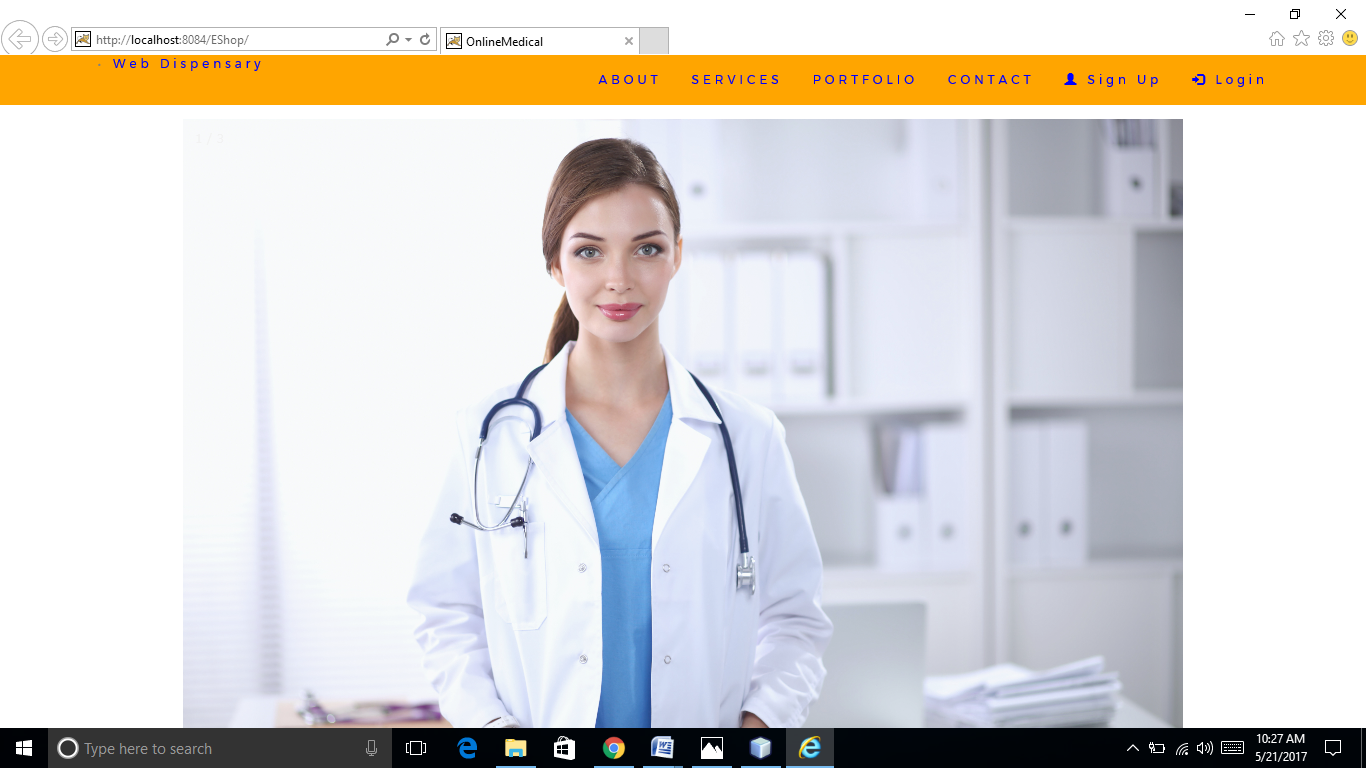


Fig 2.2 Company page

* Registration Page

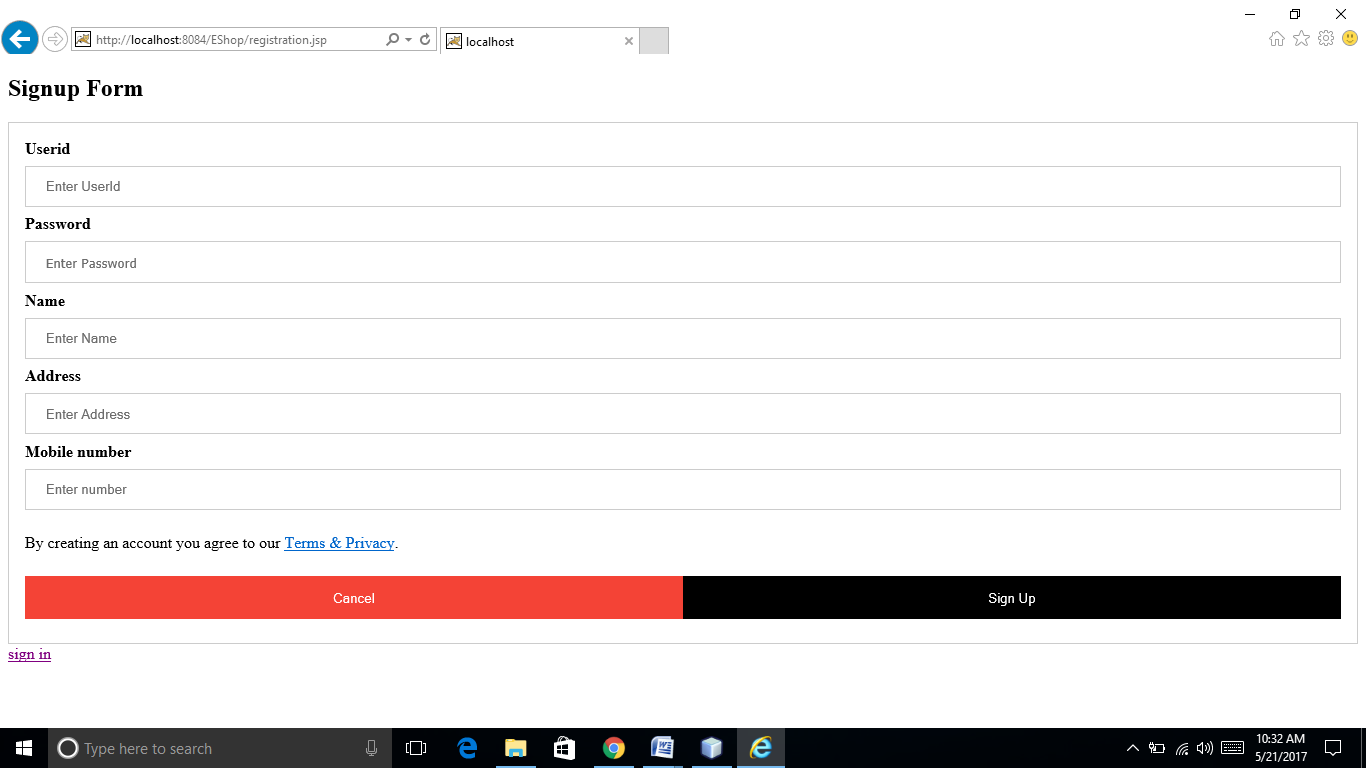


Fig 2.3 Sign up page

* Login page

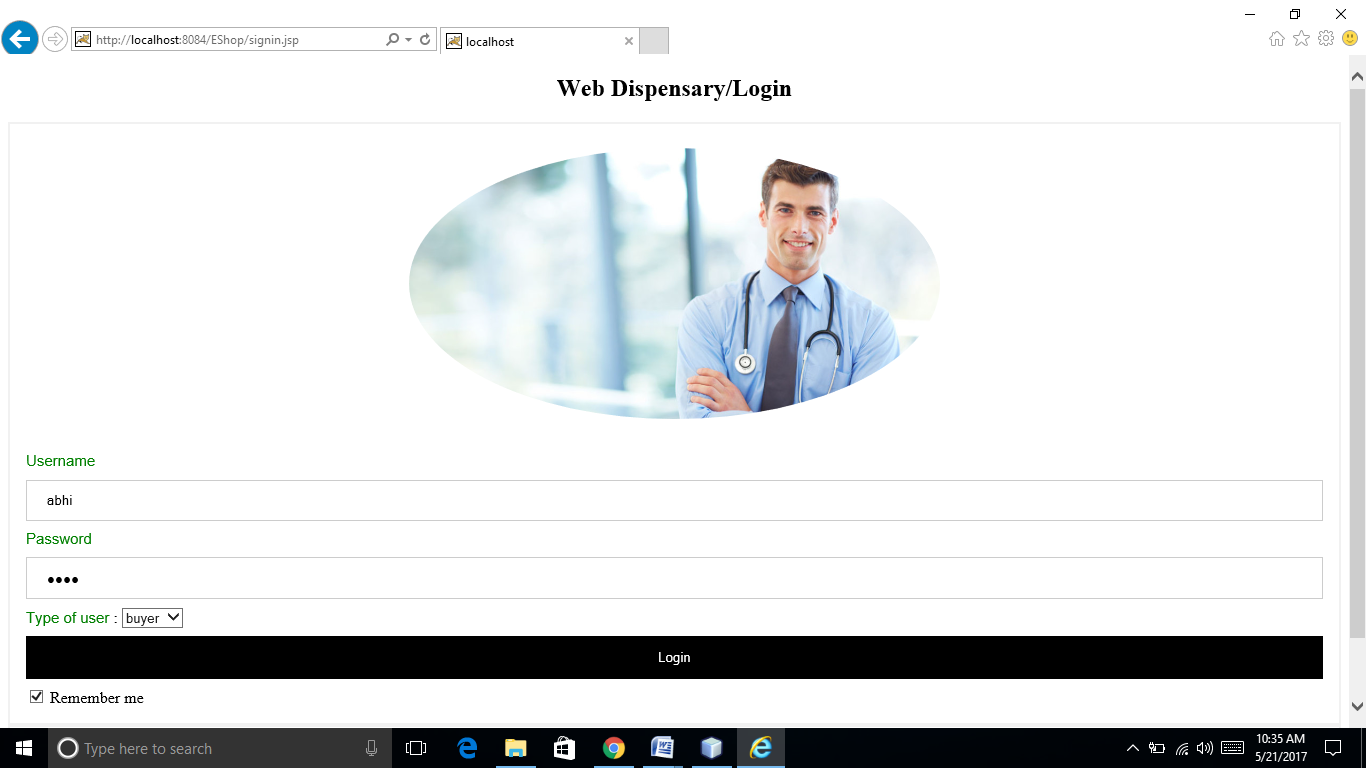


Fig 2.4 Login page

* Search Page

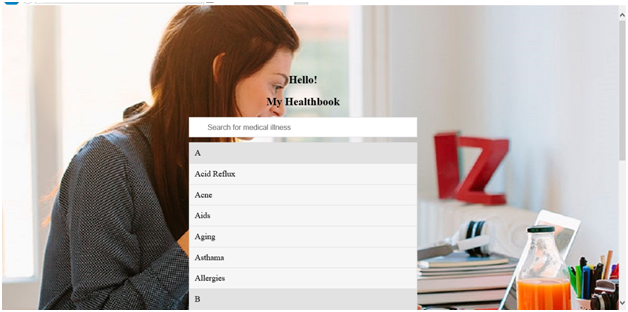


Fig 2.5 Explore illness

* Buy cart Page

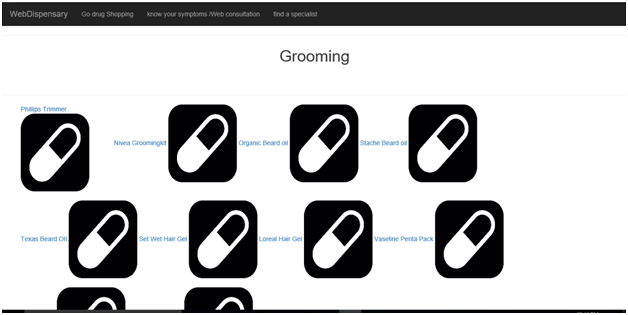


Fig 2.6 Shop drugs

* **Search for doctors Page**



Fig 2.7 Explore doctors

2.1.2 Hardware Interfaces

A system will run over the internet all the hardware should be connected to the internet and this is the hardware interface for the system. Example – WAN,LAN etc.

Hardware requirement end the user is strtaight forward that is any web browser. Although hardware should be good enough to support the web servers.

client side - internet explore 6,intel pentium 4, ram 256 MB

server side-RAD , DB2, intel pentium 4, RAM 512GB

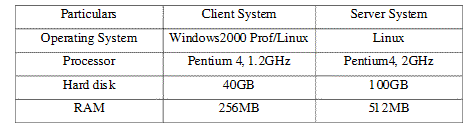


Fig 2.8 Hardware Interfaces

2.1.3 Software Interfaces

The system shall be capable on running on any version of OS. The application should support all major web browsers that will make it convenient for the user to access our system with ease, the backend that is the database services will be used to a great extent and should be designed efficiently. The system is on server so it requires any scripting language like JSP. The system requires database for the storage for any transaction of the system like MySQL.

**Software Configuration:**

**Operating System:** Microsoft Windows 8.1

**Application Software:** MySQL, NetBeans IDE 8.1

**Server side:** An Apache web server will accept all requests from the client and forward it accordingly. A database will be hosted centrally using MySQL.

**Client side:** An OS which is capable of running a modern web browser which supports JavaScript and  HTML5.

2.1.4 Communication Interfaces

The Web Dispensary systems shall use

Client (customer) on Internet will be using HTTP/HTTPS protocol.

Client (system user) on Internet will be using HTTP/HTTPS protocol.

**2.2 Software Functional Requirements:**

**1. User Registration:**

Users would need to register first to start blogging or accessing the whole website. The process for registration would be as is outlined in the process below:

Web Dispensary -> Not a Member yet? -> Enter an unique username, email-id, contact no, password -> Click on Register

**2. Login Functionality:**

Users can sign in by using the username and password he/she entered while registration. After this, user should be able to access the whole website easily.

**2.3 Functions:**

With the online dispensary , we expect to facilitate patients with an sit at home order technique for their drugs . Also we allow them an online consultation; articles and suggestions from the doctors in order to save them the amount of doctor appointment.

This system is aimed at the smooth functioning of the online medication procedure. To

  achieve this, our application has been divided into the following modules:

E commerce platform for shopping medicines

E commerce platforms for shopping of healthcare products

online search for the best doctor based on your symptoms

online consultation; from a doctor's textual database based on your symptoms

**Usability:** It is useful in following situations:

patients can order their medicines online.

People can order their healthcare products  online.

Anybody looking for medical consultation can  have it online without actually having to go to the doctor.

people can know about the kind of specialist they need.

People can know about various drugs and medicines they have been using.

People can learn about natural ways to fight illness.

**2.4 PERFORMANCE REQUIREMENTS**

The system can perform in heavy traffic also multiple faculties and students can simultaneously interact. The students will not have to wait even if the teacher is interacting to any other students. The system is designed so as to utilize the CPU resources in an optimum manner.

**2.5 DESIGN CONSTRAINTS**

* Java will be used for developing the project as all team members are trained to  use it.
* Net beans IDE will be used as the development tool due to its familiarity with all the team members.
* Since all the team members are trained to use MySqlServer 5.0 as the database server, so it will be implemented in the back-end development.
* GUI is not designed in hindi.

**2.5.1  STANDARDS COMPLIANCE**

There shall be consistency in variable names within the system. The graphical user interface shall have a consistent look and feel. Since we will access the centralized database  so any change in that database will also affect the data transaction in our project also.

**2.6 SOFTWARE SYSTEM ATTRIBUTES**

**2.6.1 Reliability**

Mean Time between Failures (MTBF) — The MTBF depends upon the connectivity of network at the location of the node. At a location having good network reach ability MTBF should be high. The software is reliable. It will work efficiently until LAN connectivity is there.

**2.6.2 Availability**

The system shall be available 24 hours a day, 7 days a week. In case  of failure the system shall allow users to restart the application after failure with the loss of at most 12 characters of input.

**2.6.3 Security**

Since the plan is to deploy the server component of project on Linux so any

kind of virus or spyware threats are out of consideration and on client side if client have a

LAN connection with the server, then they can use the project. Along with that, different

levels of security are provided for Administrator and User.

**2.6.4 Maintainability**

The system shall utilize interchangeable plugins. The system shall be easily updatable for fixes and patches. The system shall create logs of all changes, updates, or fixes that are done to the site.the system shall be easy to upgrade.

**2.6.5 Portability**

The system shall be extremely portable via the usb drive.The system shall be easy to migrate or backed up via another usb drive.

**2..6.6 Supportability**

This software is supportable on any platform with java compatibility.

3. **Diagrams**

3.1. E-R diagram

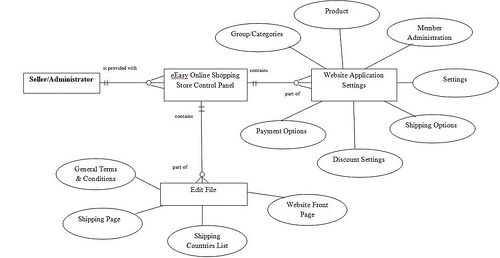


Fig 3.1 E-R diagram

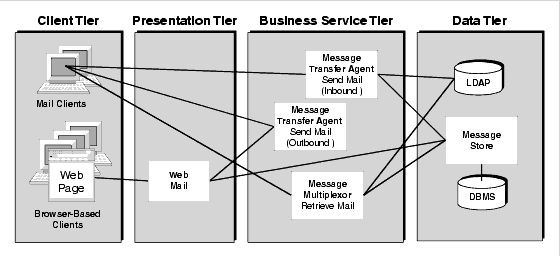
**3.2 Architecture Diagram** 

Fig 3.2 Architecture Diagram

3.3 USE CASE DIAGRAM

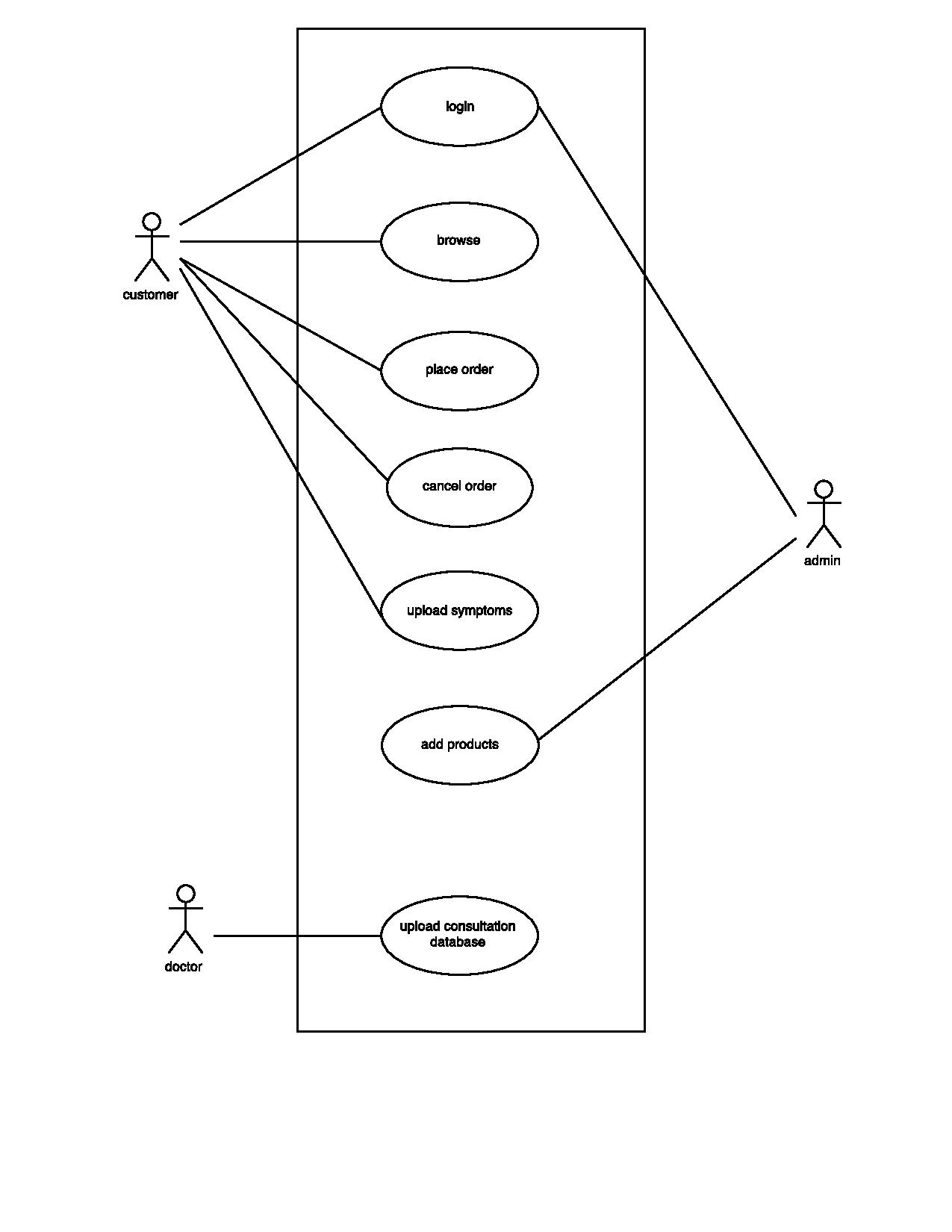


Fig 3.3 Use Case Diagram

3.4 CLASS DIAGRAM

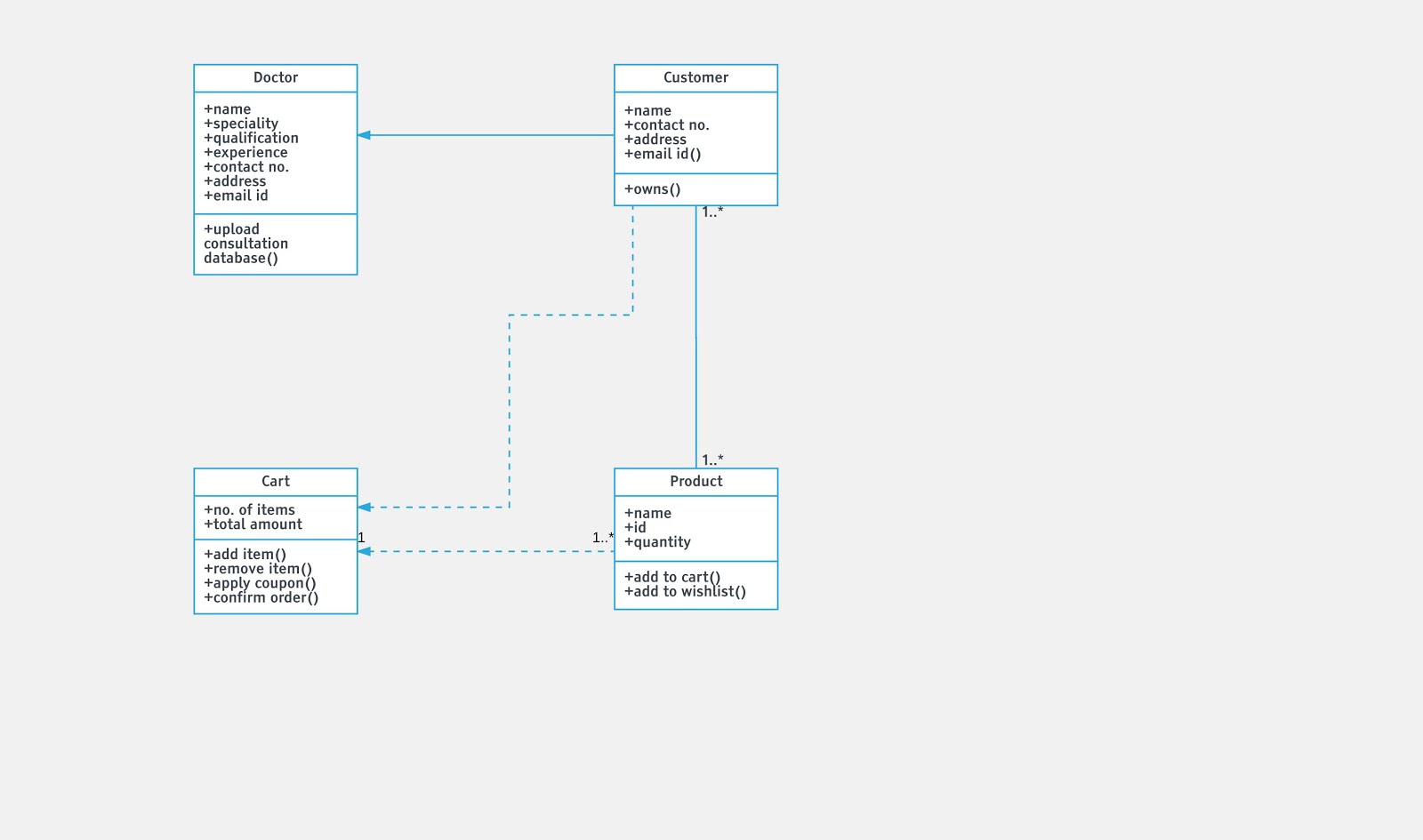


Fig 3.4 Class Diagram

3.5 SEQUENCE DIAGRAM

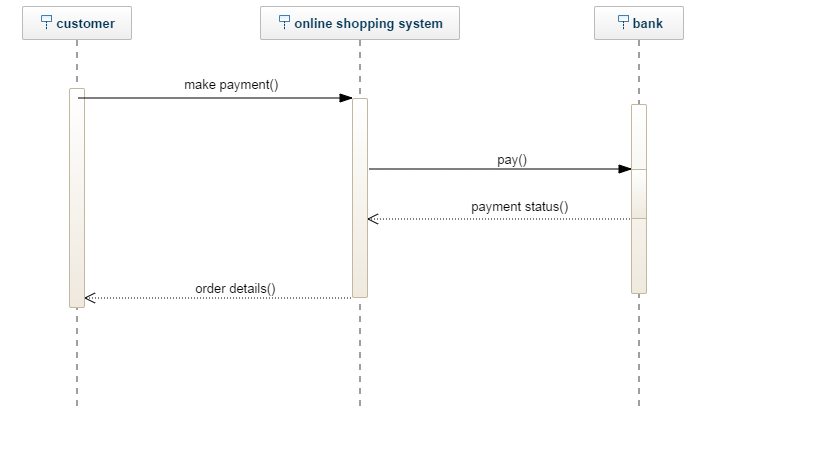


Fig 3.5 Sequence Diagram

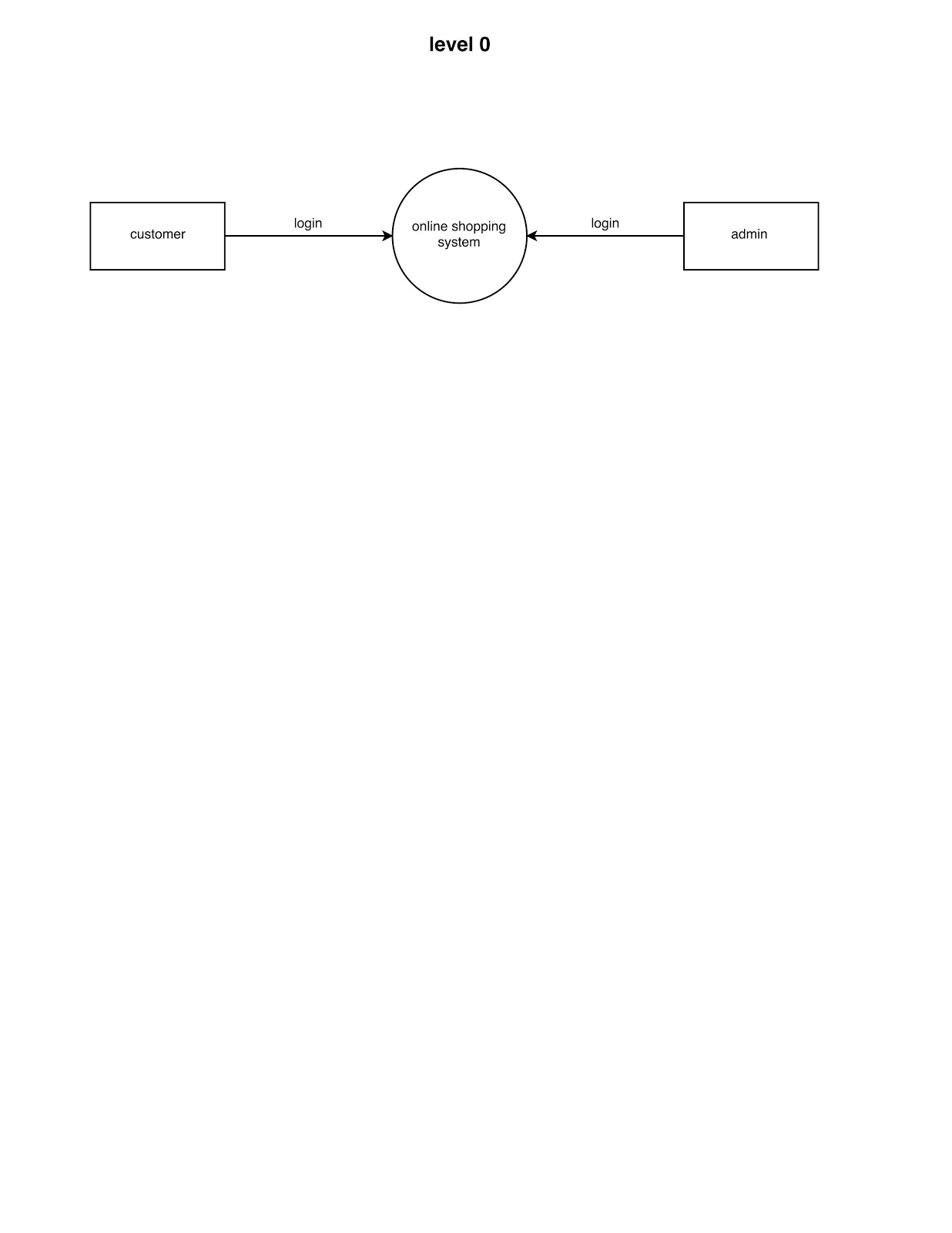
3.6 DFD diagram

Fig 3.6.1 Level 0 DFD

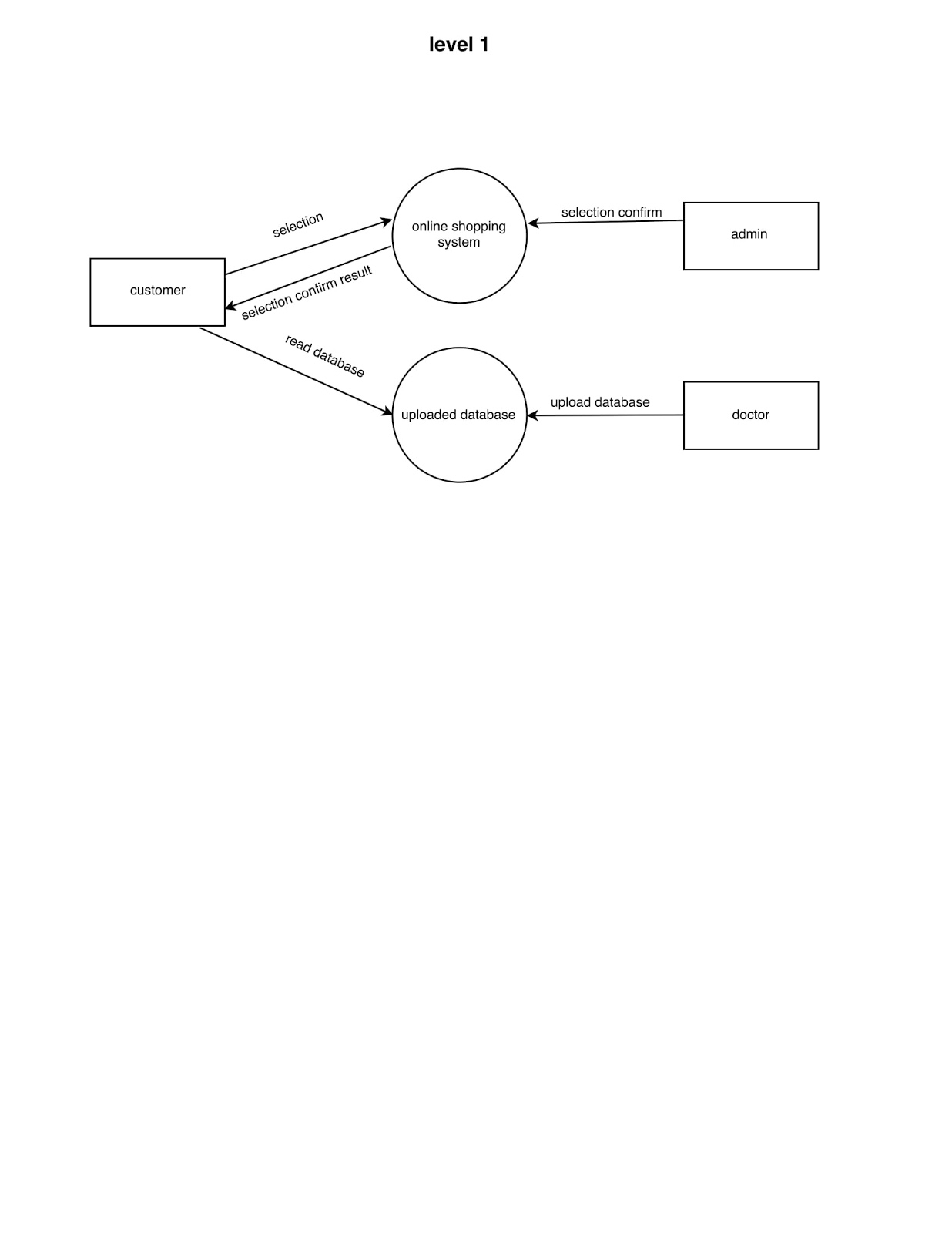


Fig 3.6.2 Level 1 DFD

3.7 Activity diagram

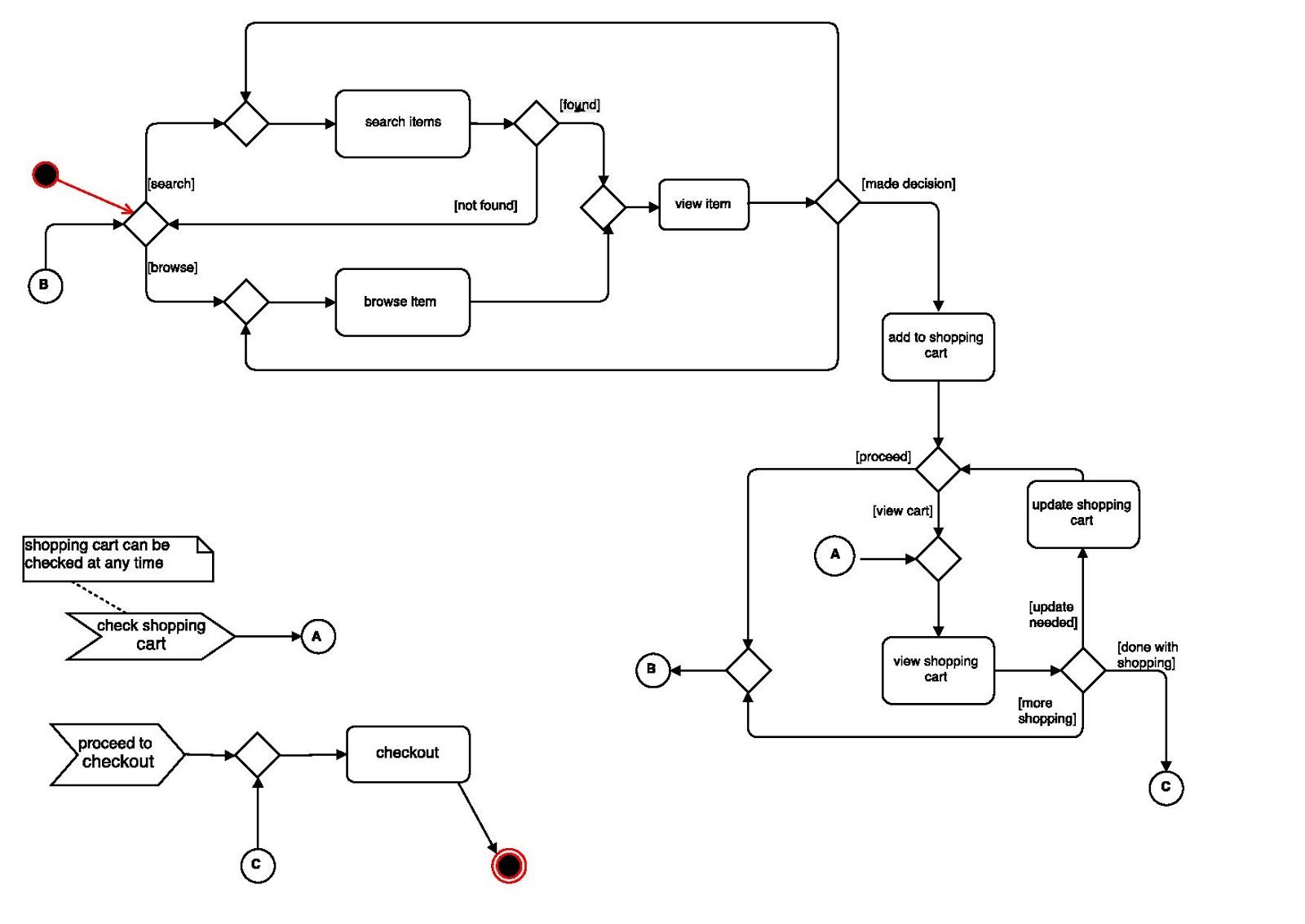


Fig 3.7 Activity diagram

**CHAPTER 4 – Software Test Documentation (STD)**

**1. INTRODUCTION**

**1.1 System Overview:**

Any visitor can use our website but a visitor needs to register to our website first. Any visitor after registration becomes a User. Then the user has privileges to read blogs posted by others and also he/she can write blog for us. We provide flexibility to read and write multiple categories of contents in our website.

**1.1**  **Test Approach:**

Tests can be conducted based on two approaches –

·         Functionality testing

·         Implementation testing

When functionality is being tested without taking the actual implementation in concern it is known as black-box testing. The other side is known as white-box testing where not only functionality is tested but the way it is implemented is also analyzed.

**1.2**  **Testing Objectives:**

The main objective of testing is to uncover as many errors as possible. It can be carried out in different phases and in each section, verification is conducted by taking review. Same in our project, we test each & every functionality & their performance, so we can reduce our flaws.

**2. TEST PLAN**

**2.1 Features to be tested:**

* **Invalid Registration problem**
* **Login failure problem**
* **Search For the Doctor**
* **Search for the illness**

**2.2 Features not to be tested:**

* **Functioning of static pages**
* **Functioning of links to some informative website**
* **No product Check-out**
* **Search for a product**

**3. TEST CASES**

A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing test cases can also help find problems in the requirements or design of an application. In order to fully test that all the requirements of an application are met, there must be at least two test cases for each requirement: one positive test and one negative test. If a requirement has sub-requirements, each sub-requirement must have at least two test cases. Keeping track of the link between the requirement and the test is frequently done using a [traceability matrix](https://en.wikipedia.org/wiki/Traceability_matrix). Written test cases should include a description of the functionality to be tested, and the preparation required to ensure that the test can be conducted.

**Table 3.1: User Login Form**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case** | | | | | |
| **Test scenario name** | | | **Login Form.** | | |
| **Description** | | | **This scenario covers the functionality of the user who wants to login to the system.** | | |
| **Module Name** | | | **Normal User** | | |
| **Status** | | | **Created** | | |
| **Tester Information** | | | | | |
| **Date of Test** | | | **4th May 2016** | | |
| **Time of Test** | | | **10:25 Hours** | | |
| **Operating System** | | | **Windows 7/8/10** | | |
| **ID** | **Test Case** | **User Input** | | **Expected Result** | **Test Result** |
| **User\_**  **Login 1** | **Login** | **User must have input all the valid information** | | **Home page will open** | **Successful** |
| **User has input some invalid information** | | **The error messages will be displayed to the user on those check fields.** | **Successful** |

**Table 3.2: User Registration Form**

|  |  |
| --- | --- |
| **Test Case** | |
| **Test scenario name** | **Registration  Form** |
| **Description** | **This scenario covers the functionality of the user who wants to create an account.** |
| **Module Name** | **Normal User** |
| **Status** | **Created** |
| **Tester Information** | |
| **Date of Test** | **4th May 2017** |
| **Time of Test** | **12:10Hours** |
| **Operating System** | **Windows 7/8/10** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Test Case** | **User Input** | **Expected Result** | **Test Result** |
| **User\_registration 2** | **Registration form** | **User must have input all the valid information** | **It will get the window to fill the registration form** | **Successful** |
| **User has input some invalid information** | **The error messages will be displayed to the user on those check fields.** | **Successful** |

**Table 6.4: Exploring Illness**

|  |  |
| --- | --- |
| **Test Case** | |
| **Test scenario name** | **Exploring Illness** |
| **Description** | **This scenario covers the functionality of Exploring illness module** |
| **Module Name** | **Normal User** |
| **Status** | **Created** |
| **Tester Information** | |
| **Date of Test** | **5th May 2016** |
| **Time of Test** | **10:22 Hours** |
| **Operating System** | **Windows 7/8/10** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Test Case** | **User Input** | **Expected Result** | **Test Result** |
| **Explore\_illness  6** | **Explore illness** | **User must enter the illness , he/she needs to learn about** | **It will provide the  user with the link to the page which has required data of the illness** | **Successful** |
| **User has entered wrong name for the illness** | **No data will be made available** | **Successful** |

**·**

**Table 6.4: Searching doctors**

|  |  |
| --- | --- |
| **Test Case** | |
| **Test scenario name** | **Searching Doctors** |
| **Description** | **This scenario covers the functionality of Searching for doctors module** |
| **Module Name** | **Normal User** |
| **Status** | **Created** |
| **Tester Information** | |
| **Date of Test** | **5th May 2016** |
| **Time of Test** | **11:22 Hours** |
| **Operating System** | **Windows 7/8/10** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Test Case** | **User Input** | **Expected Result** | **Test Result** |
| **Explore\_illness  6** | **Explore illness** | **User must enter the symptoms ,he/she are encountering** | **It will provide the  user with the link to the page which tells the user the type of specialists they must refer. A new windoww that gives the catalogue of such specialists.** | **Successful** |
| **User has entered wrong symptoms** | **No data will be made available** | **Successful** |

**·**

**3.1 Unit Testing-**

In unit testing the individual components are tested independently to ensure their quality. It is a level of software testing where individual units/components of software are tested. The purpose is to validate that each unit of the software performs as it designed.

**3.2  Functional Testing:**

The main focus of functional testing is:

·      Checking of all the functionalities that they are working properly or not.

·      At a given request proper response is given by the system or not.

·      Main purpose of the client is fulfil or not by the project(developer).

**3.3  System Testing:**

The main focus of such testing is to test:-

·      System functions and performance.

·      System reliability and recoverability.

·      System installation.

·      System behaviour in the special conditions.

·      System user operations.

·      Hardware and Software integration and collaboration.

**3.4  Integration Testing:**

A group of dependent components are tested together to ensure their quality of their integration unit. The objective is to take unit tested components and build a program structure that has been dictated by software designer.

**3.5  Validation Testing:**

In validation testing the main focus is to uncover errors in:-

·      System input/output.

·      System function and information data.

·      System interfaces with external parts.

·      User interface.

·      System behaviour & performance.

**3.6  Top Down Integration:**

Top down testing is an incremental approach in which modules are integrated by moving down through the control structure. Modules subordinate to the main control module are incorporated into the system in either a depth first or breadth first manner.

**3.7  Bottom Up Integration:**

In bottom up integration the modules at the lowest levels are integrated at first, then integration is done by moving upward through the control structure. Low level modules are combined into clusters that perform a specific sub-function.

**4. CONCLUSION & FUTURE SCOPE**

Web Dispensary is a website through which users can go drug shopping , search for doctors and learn for their illness. The future holds a online portal for patients to directly connect with doctors at real time.