

SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE



Software Requirements Specification

ONLINE MEDICAL STORE MANAGEMENT SYSTEM

Submitted by:

Sanjan Sumita Ganesh Shetty – 13070121139

Saumya Shandilya – 13070121143

Rheeya Uppaal – 13070121133

Gauri P Desai-13070121321

Table of Contents

| | |
|---|----|
| Abstract | 3 |
| 1. Introduction | 4 |
| 1.1 Purpose | 4 |
| 1.2 Scope of the Project | 4 |
| 1.3 Objectives | 4 |
| 1.4 Glossary and References | 4 |
| 1.5 Overview | 5 |
| 2. Overall Description | 5 |
| 2.1 Process Model | 5 |
| 2.2 Software Features | 8 |
| 2.3 Functional Requirements | 8 |
| 2.4 Non – Functional Requirements | 9 |
| 2.5 Assumptions and Dependencies | 9 |
| 2.6 Risks | 10 |
| 3. Project Estimation Techniques | 12 |
| 4. Project Design | 13 |
| 3.1. Analysis Models | 13 |
| 3.1.1 Data Flow Diagram | 13 |
| 3.1.2 Entity Relationship Diagram | 19 |

| | |
|-------------------------------------|-----------|
| 3.1.3 State Transition Diagram..... | 20 |
| 3.1.4 Timeline chart..... | 21 |
| 3.2 Software System Attributes..... | 22 |
| 3.2.1 Reliability..... | 22 |
| 3.2.2 Availability..... | 22 |
| 3.2.3 Security..... | 22 |
| 3.2.4 Maintainability..... | 22 |
| 3.2.5 Portability..... | 22 |
| 4. Test Cases..... | 23 |

Abstract

The project aims to create a complete medical shop management software, designed to ease the work load of medical shop professionals. The main features include invoicing, inventory and stock control, accounting, client and vendor management.

This software helps the user to track all the profits, loss, profitable clients and products of medical shop, also serving the purpose of being as a medical shop accounting software. It is a flexible and adaptive software suited to medical shops or stores or pharmacies of any size.

This System will store the information of medicines & customers in the system. It will also store the information about stock details. This system provides security by granting access rights for changing the sensitive data to administrator only. Customized search options are also available in this system. This system can generate all kinds of reports as per the user's requirements.

1 Introduction

The following subsections of the Software Requirements Specifications (SRS) document provide an overview of the entire SRS.

1.1 Purpose

The Software Requirements Specification (SRS) will provide a detailed description of the requirements for the Medical Store Database Management System. This SRS will allow for a complete understanding of what is to be expected of the Store Database to be constructed. The clear understanding of the Database and its' functionality will allow for the correct software to be developed for the end user and will be used for the development of the future stages of the project. This SRS will provide the foundation for the project. From this SRS, the Medical Store's database can be designed, constructed, and finally tested.

1.2 Scope of the Project

The end users of this software are the medical shop professionals who would be working at the shop. The software will be accessed by the employees who are responsible for the accounting, client and vendor management of the shop.

1.3 Objectives

The objective of creating this software is to allow the staff of the medical store to keep a proper track of the stock of the medicines, customer details as well as information of the medicines. The software will be user appropriate, easy to use, provide easy recovery of errors and have an overall end user high subjective satisfaction.

1.4 References and Glossary

This is to state that the above Software Requirement Specification holds true to the efforts put by the group G-SRS. However the following references were of utmost importance to make a successful SRS as they guided us to the right path and specified and explained the parameters necessary to build an SRS. Hard work is good but a right tutor and books with an amalgamation of hard work makes it even better.

The following references have been truly acknowledged.

For implementation of the code,

- Let us C by Yashwant Kanetkar
- Let us C++ by Yashwant Kanetkar
- Word documents shared with us by Prof. Ambika Pawar
- Data Structures by Yashwant Kanetkar

For implementation of Software Requirement Specification,

- Information shared with us by Prof.Pooja Kamat
- “Software Engineering, a practitioners approach” by Roger Pressman, Sixth Edition, 2011
- “UML Distilled”, Martin Fowler, Addison Wesley, Fifth Edition, 2010.
- Sample SRS Format referred from Prof.Pooja Kamat
- Various sources from the internet

1.5 Overview

It is the complete medical shop management software is so designed as to ease the work load of medical shop professionals. The main feature includes invoicing, inventory and stock control, accounting, client and vendor management.

This software helps you to track all the profits, loss, profitable clients and products of medical shop moreover it's a medical shop accounting software. Flexible and adaptive software suited to medical shops or stores or pharmacies of any size.

2 Overall Description

This section includes the process model that has been used to develop the software as well as the functional and non-functional requirements of the database hence making it easier to understand for the developer who will be developing the software as well as the end user.

2.1 *Process Model*

To solve actual problems in an industrial setting, a team of software engineers must incorporate a development strategy that encompasses the process, methods and tools used, and the generic phases of development. This strategy is often referred to as a “*process model*” or a “*software engineering paradigm*”. A process model is selected based on the nature of the project, its application, the methods to be used, the controls and deliverables etc.

All software development can be characterized as a problem solving loop in which four distinct stages are encountered:

1. Status quo - The current state of affairs
2. Problem definition - Identifies the current problem to be solved
3. Technical Development - Solves the problem through the application of some technology
4. Solution Integration - Delivers the results to the internal or external customers

This problem solving loop applies to software engineering work at many different levels of resolution. These loops can also be recursive.

Realistically, it is difficult to compartmentalize activities as neatly because interdependency exists among the phases. However, this simplified view drives a point: regardless of the process model selected, the stages coexist simultaneously.

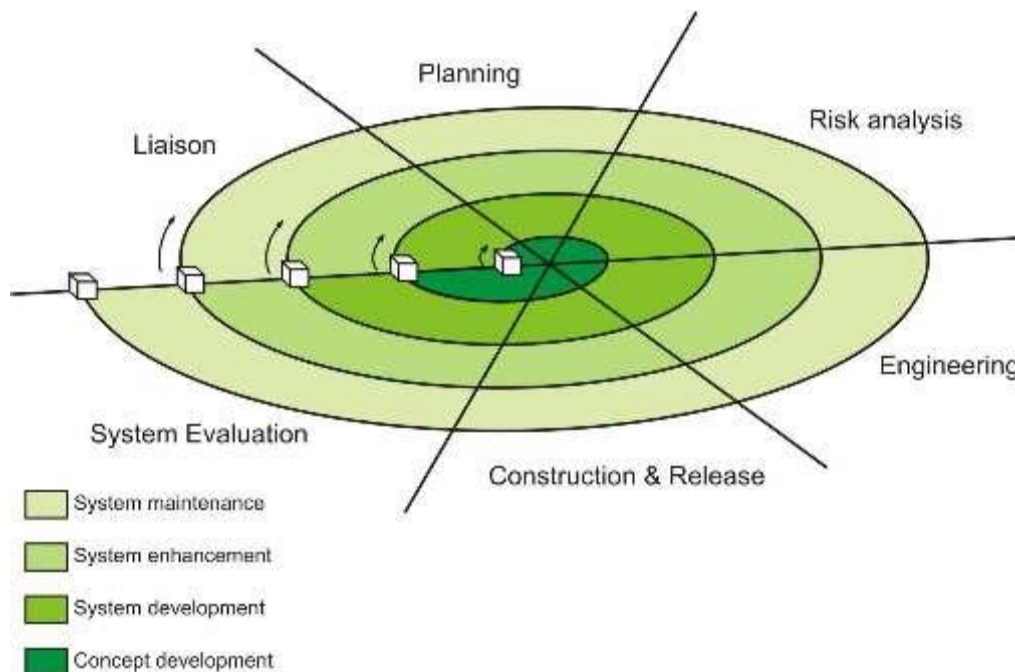
It has been suggested that a “chaos model” describes software development as a “continuum from the user to the developer to the technology.” Software models attempt to bring order to an inherently chaotic activity.

The spiral model is an evolutionary software model that couples the iterative nature of prototyping with the controlled and systematic aspects of the linear sequential model. It provides the potential for rapid development of incremental versions of the software. Using the spiral model, the software is developed in a series of incremental releases. During early iterations, the releases might be something akin to a prototype. During later iterations, more complete versions of the engineered system are produced.

Our spiral model is divided into a number of framework activities or task regions. There are six task regions:

- Customer Communication – Tasks which establish a proper communication between the customer and developer
- Planning – Define resources, timelines and other such information.
- Risk Analysis – Assess technical and management risks.
- Engineering – Build one or more representation of the application.
- Construction and Release – Construct, test, install and provide user support
- Customer Evaluation – Obtain customer feedback based on evaluation of the software representations created during the engineering stage and implemented during the installation stage.

Each of these regions contains a set of tasks called a task set, which have been adapted to the project of the medical management system. Umbrella activities (like software configuration management and software quality assurance) have been applied throughout.



A diagrammatic representation of the spiral model.

Action begins from the first task, Customer Communication (or Liaison), in the innermost circle representing concept development. The action then continues in a clockwise direction around the spiral. With every new turn in the spiral, new parts of the project (System development, followed by enhancement and maintenance) are performed.

Our team adopted the Spiral Process Model for the following reasons:

- Spiral model is a realistic approach to the development of systems and software.
- The developer and customer better understand and react to risks.
- Prototyping can be applied at any stage in the development.

- The model demands a direct consideration of technical risks at all stages of the project, reducing risks before they become problematic. In software like a Medical Management System, managing risks becomes of utmost importance.
- Additional Functionality can be added at a later date.
- Software is produced early in the software life cycle.
- Costs and risk evaluation are important to the project
- Users are unsure of their needs

2.2 *Software Features*

- i) The software is menu driven, with keyboard and mouse navigation.
- ii) The practice is paperless.
- iii) Improve efficiency, productivity.
- iv) Cost effective solution.
- v) Separate login for the HR Manager and Retail Executive.
- vi) Retail Executive maintains records of all the employees working for the store.
- vii) Cashiers can insert, delete, search records in medicine inventory and calculate the day's sale.

2.3 *Functional Requirements*

The software shall implement the following functionalities:

From the Human Resources Manager's perspective:

- i) The administrator shall enter his/her username and password which would be verified and only then shall he/she be logged in to the database.
- ii) The administrator can create a new employee record by providing the employee's id, name, salary and serial no.
- iii) The administrator can add a new record.
- iv) The administrator can search for an existing record (by name) in the database.
- v) The administrator can search for an existing employee record (by serial no.).
- vi) The administrator can search for an existing employee record (by employee id).
- vii) The administrator can delete an existing employee record from the database.
- viii) The administrator can update any record in the database (for e.g. employee's salary) and hence update the complete database.
- ix) The administrator can view the records of the employees.
- x) The administrator can exit from the given program whenever he/she wants.

From the Retail Executive's perspective:

- i) The cashier shall enter his/her username and password which would be verified and only then he/she shall be logged in.
- ii) The cashier can create a record in the inventory.

- iii) The cashier can add a new record to the inventory.
- iv) The cashier can delete a record from the inventory.
- v) The cashier can search for a record in the inventory.
- vi) The cashier can view his entire inventory.
- vii) The cashier can also view the nearest chemists' inventory.
- viii) A calculator is provided which shall allow the cashier to use it as and when required.
- ix) The cashier can exit from the given program whenever he/she wants.

2.4 Non-Functional Requirements

2.3.1 Performance Characteristics

- i) The software shall accommodate 1 user at a time.
- ii) The software shall have a multiuser login system which will allow the admin and the employee to use the same software.
- iii) The software shall be accessible by the end user 24/7 as long as the end user has his/her username and password.
- iv) The software shall not take more than 1 second to begin running.

2.3.2 Safety Requirements

- i) While the end user is accessing the software, no form of virus shall disrupt/harm the computer system.

2.5 Assumptions and Dependencies

It has been assumed that:

- The users are already aware of their passwords.
- Only 2 users can login to the system.
- The users must be specially trained to use the system.
- The user must know how to express an expression in prefix format.
- The employee database is dependent on the HR Manager for its records.
- The system hardware and software requirements are available.

2.6 Risks

A risk is a potential problem – it might happen and it might not.

Two characteristics of risks:

1. Uncertainty – the risk may or may not happen, that is, there are no 100% risks (those, instead, are called constraints)
2. Loss – the risk becomes a reality and unwanted consequences or losses occur

Project risks:

- They threaten the project plan
- If they become real, it is likely that the project schedule will slip and that costs will increase

Technical risks

- They threaten the quality and timeliness of the software to be produced
- If they become real, implementation may become difficult or impossible

Business risks

- They threaten the viability of the software to be built
- If they become real, they jeopardize the project or the product
- Sub-categories of Business risks:
 - Market risk – building an excellent product or system that no one really wants
 - Strategic risk – building a product that no longer fits into the overall business strategy for the company
 - Sales risk – building a product that the sales force doesn't understand how to sell
 - Management risk – losing the support of senior management due to a change in focus or a change in people
 - Budget risk – losing budgetary or personnel commitment

A risk table provides a project manager with a simple technique for risk projection. It consists of five columns:

- Risk Summary – short description of the risk
- Risk Category
- Probability – estimation of risk occurrence based on group input
- Impact – (1) catastrophic (2) critical (3) marginal (4) negligible
- RMMM – Pointer to a paragraph in the Risk Mitigation, Monitoring, and Management Plan

| Risk Summary | Risk Category | Probability | Impact | RMMM |
|--------------------|---------------|-------------|--------|---|
| Power Failure | Known Risk | 0.3 | 3 | Use a back up power supply. Monitor the risk often. |
| System Crashed | Unpredictable | 0.25 | 2 | Keep a backup of all the data in the database as there can be easy retrieval of data |
| Software Corrupted | Unpredictable | 0.2 | 2 | Use a premium antivirus or apply effective cyber security measures as well as restore points to avoid risks |

An effective strategy for dealing with risk must consider three issues
(Note: these are not mutually exclusive i.e they can occur at the same time)

- Risk mitigation (i.e., avoidance)
- Risk monitoring
- Risk management and contingency planning

Risk mitigation (avoidance) is the primary strategy and is achieved through a plan:

- During risk monitoring, the project manager monitors factors that may provide an indication of whether a risk is becoming more or less likely
- Risk management and contingency planning assume that mitigation efforts have failed and that the risk has become a reality
- RMMM steps incur additional project cost

Risks can occur after the software has been delivered to the user.

3. Project Estimation Techniques

2.1 COCOMO (Constructive Cost Estimation Model)

COCOMO (Constructive Cost Estimation Model) was proposed by Boehm [1981].

Basic COCOMO

The basic COCOMO estimation model is given by the following expressions:

$$\text{Effort} = a^1 \times (\text{KLOC})^{a^2} \text{ PM}$$

$$\text{Tdev} = b^1 \times (\text{Effort})^{b^2} \text{ Months}$$

Where

- KLOC is the estimated size of the software product expressed in Kilo Lines of Code,
- a^1, a^2, b^1, b^2 are constants for each category of software products,
- Tdev is the estimated time to develop the software, expressed in months,
- Effort is the total effort required to develop the software product, expressed in person months (PMs).

NOTE-

There are 850 Lines of Code in our given program

Assuming per LOC is Rs.15/- then the estimated cost for the project is Rs.12750/-

We have implemented the Basic COCOMO MODEL (i.e. Organic),

So the effort developed is $2.4 * (0.85)^{1.05} = 2.0235 \text{ PM}$

The time developed is $2.5 * (2.0235)^{0.38} = 3.26 \text{ Month}$

4. Project Design

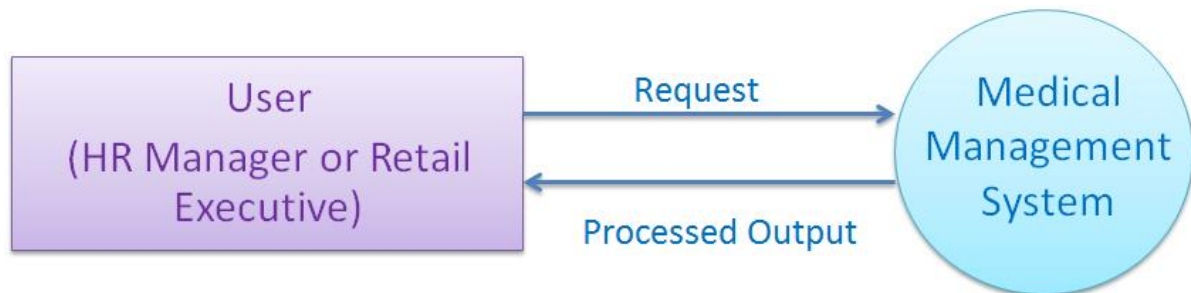
Software design is the process by which an agent creates a specification of a software artifact, intended to accomplish goals, using a set of primitive components and subject to constraints.

4.2 Analysis Models

The software design process consists of three main parts:

- Data modeling – makes use of the Entity Relationship Diagram (ERD)
- Functional modeling - makes use of the Data Flow Diagram (DFD)
- Behavioral modeling – makes use of the State Transition Diagram (STD)

4.1.1 Data Flow Diagram



Copyright © 2015 by the 'G-SRS Team'. All rights reserved.

Data Flow Diagram – Level 0

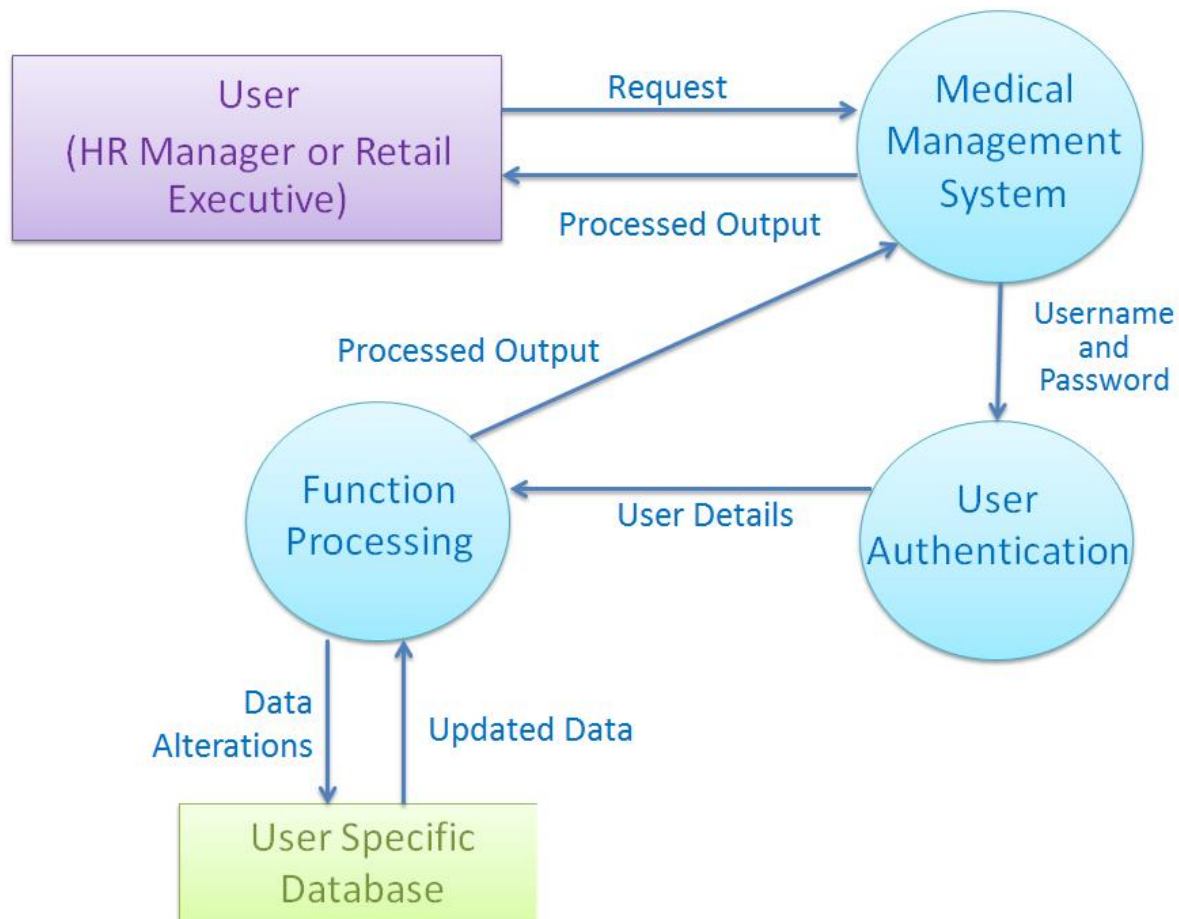
The software can be explained as a user sending a request to the software. The software, consisting of the medical management system process, processes the request, updates the required databases and processes and displays the processed output.

The Level 1 diagram explains the processes in more detail. Major processes and major data stores have been identified. As in Level 1, the user sends a request to the Medical Management System.

The System asks for the Username and Password from the User. This data is sent to the User Authentication Process. The process determines whether the user has been authenticated and checks which user (i.e. HR Manager or Retail Executive) is using the system. It sends these details to the next process.

The Function Processing process accepts the user details and displays a list of functions to the user. Depending on which function has been selected, a different Function Process is executed. The Function Processing sends required data to the Database (which varies, depending on which user is using it), updates the data based on the inputs, and retrieves the updated data.

The process then sends the processed output to the Medical Management System, which returns it to the user.



Copyright © 2015 by the 'G-SRS Team'. All rights reserved.

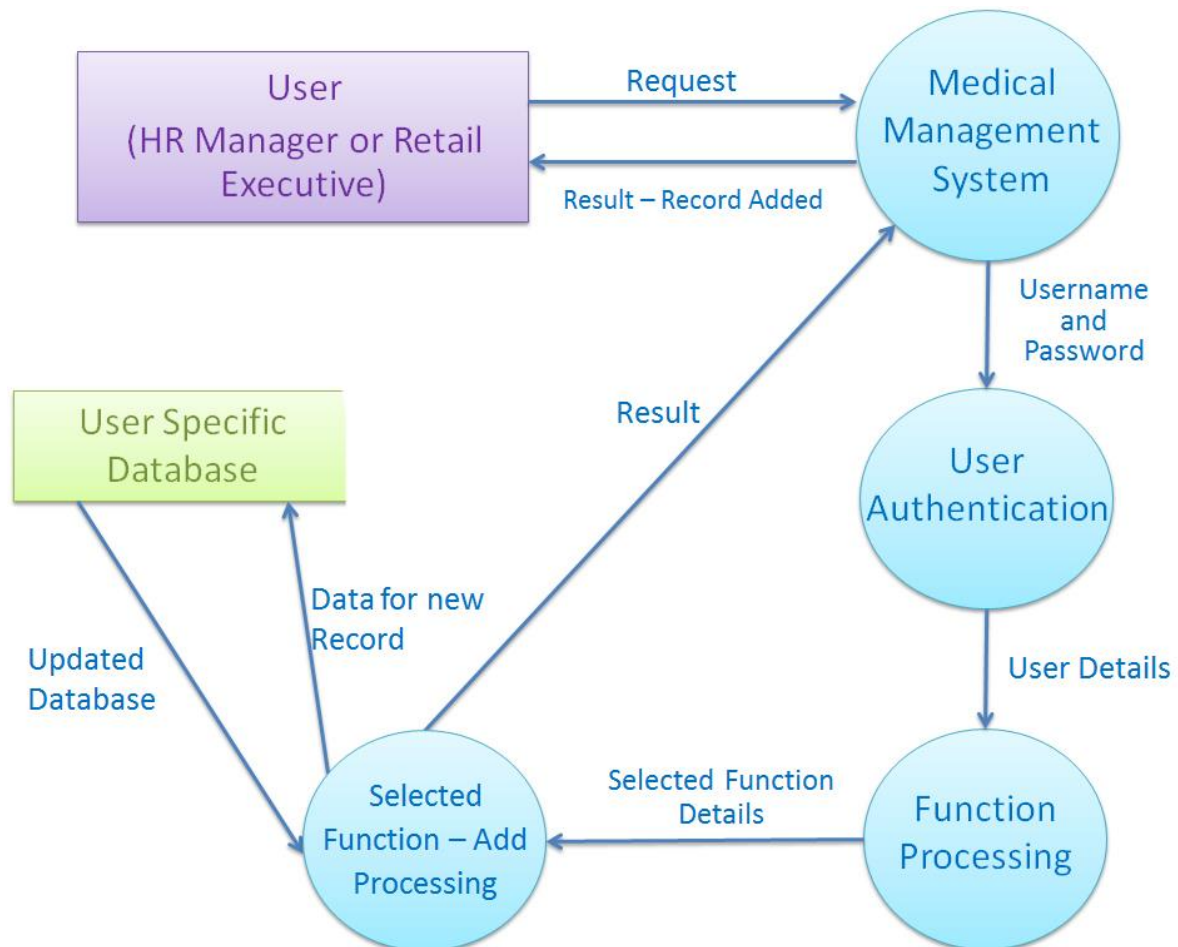
Data Flow Diagram – Level 1

The Level 2 diagram in a Data Flow Diagram looks at each process of the software in detail. Thus, there are many level 2 diagrams, each of which focuses on one process.

The Function Processing can be split into further sub-processes, namely:

1. Insert new record function
2. Delete existing record function

3. Search for an existing record function
4. Display all records function

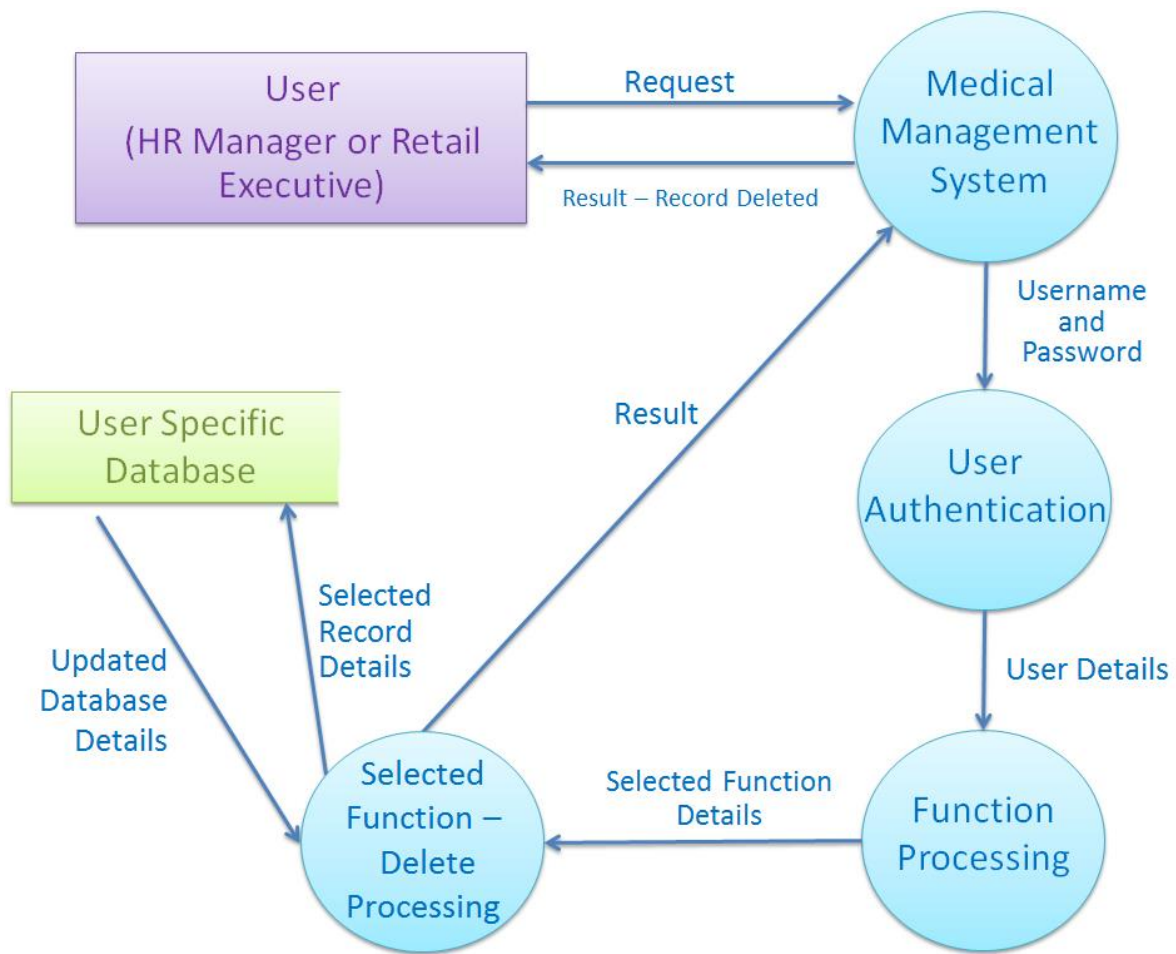


Copyright © 2015 by the 'G-SRS Team'. All rights reserved.

Data Flow Diagram – Level 2 – Insert Function

Until the Function Processing stage, the software functions as described in level 1. After this, the Function Processing determines which of the aforementioned functions have been selected by the user. In this case, the Insert Function has been selected. Thus, the Function Process sends the function details to the Insert Function process.

The insert function process receives the data for the new record from the previous process. It sends the data to the Manager or Executive database (depending on the type of login) and updates the data in it. It procures, from the database, whether the addition of a new record was successful. It then sends this result to the Medical Management System, which further sends it to the user.

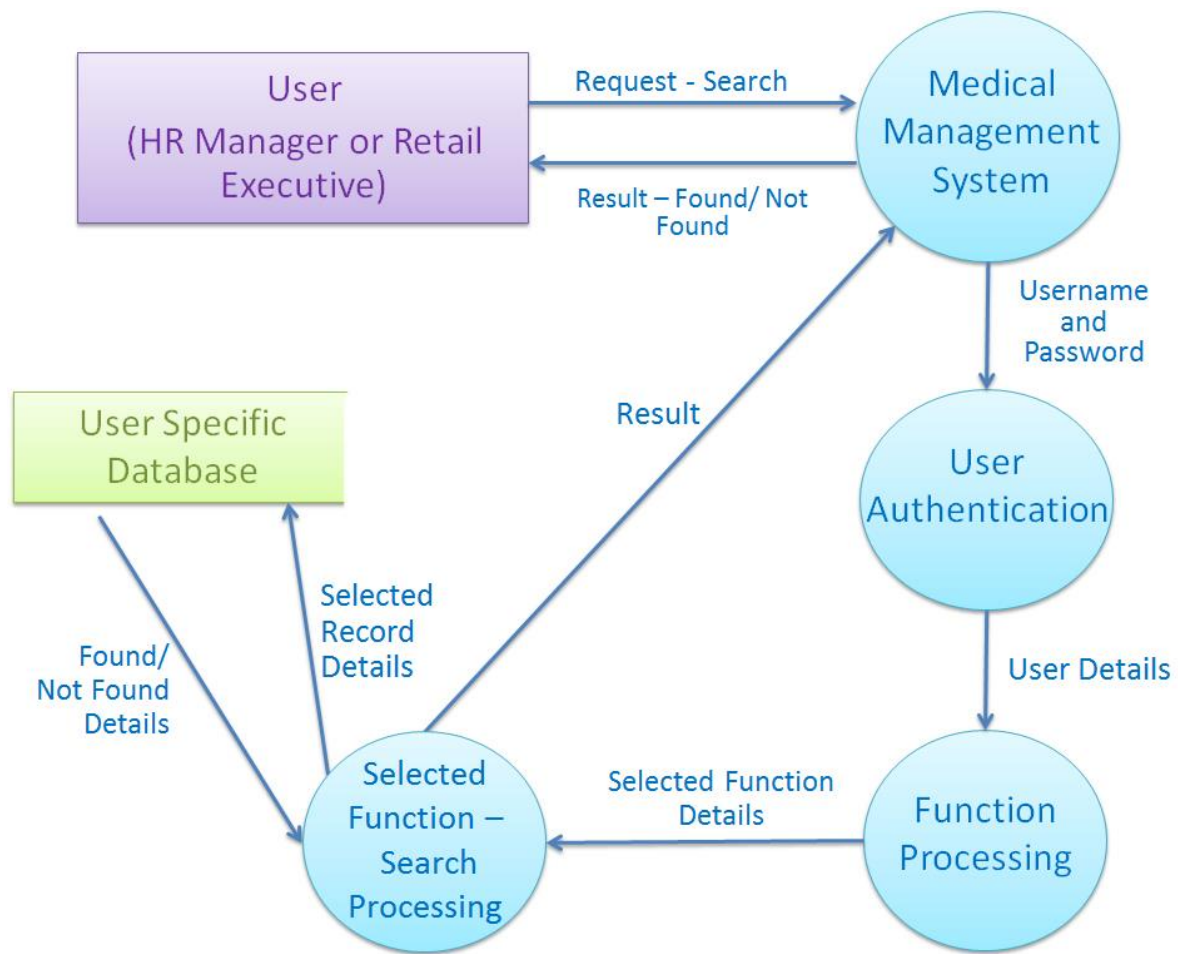


Copyright © 2015 by the 'G-SRS Team'. All rights reserved.

Data Flow Diagram – Level 2 – Delete Function

The Level 2 for the Delete Function works in a similar way as the Level 2 Insert Function. After the reaching the Processing for the Delete Function, the function searches the Database for the required record. If it is found, it deleted the record from the database.

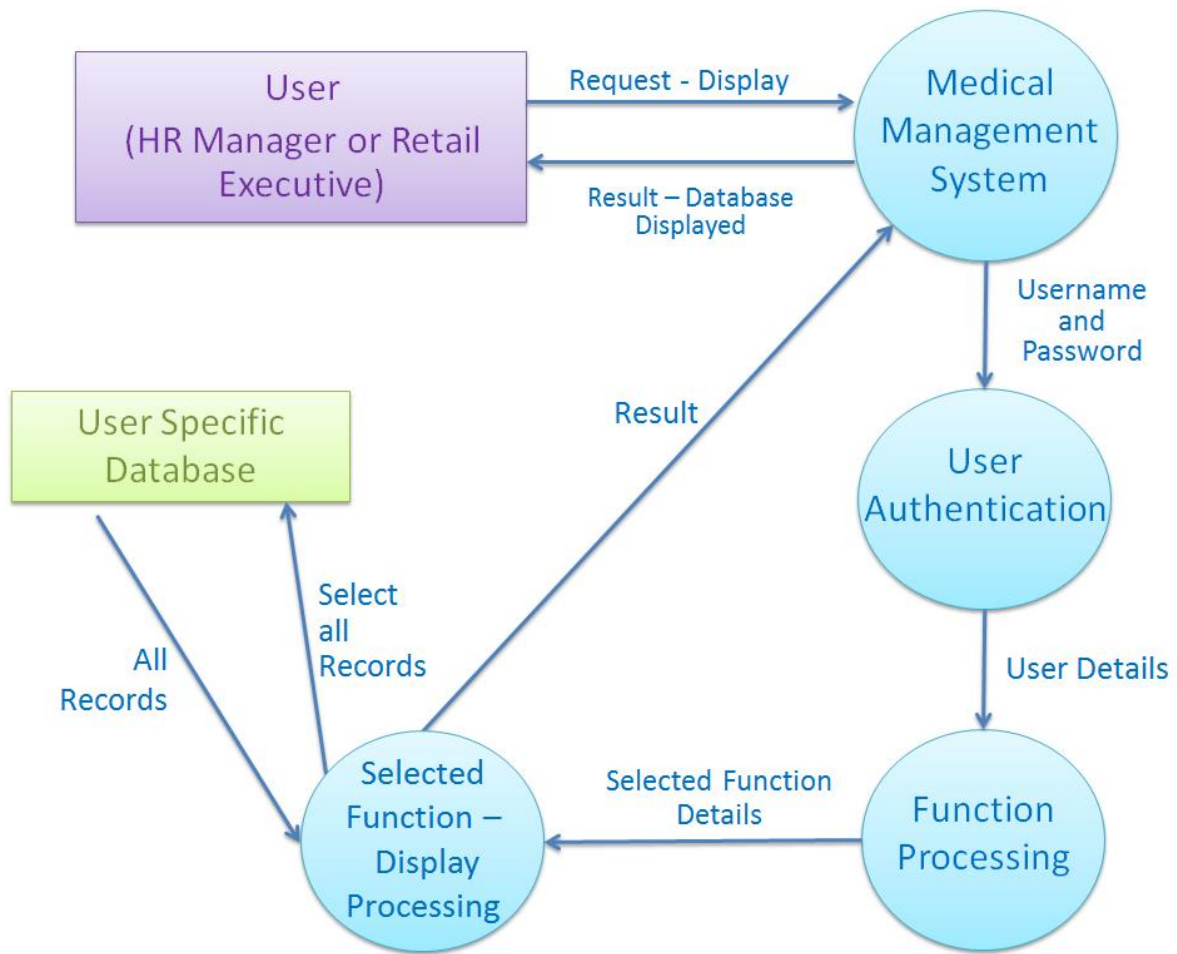
The process checks the result of the delete operation from the database and sends to the Medical Management System, which further sends it to the user.



Copyright © 2015 by the 'G-SRS Team'. All rights reserved.

Data Flow Diagram – Level 2 – Search Function

The Level 2 for the Search Function works in the same way till the Search Function Processing. Here, the process accesses the database by sending the Search Key to the Database to search for the appropriate record. Depending on whether the record is found or not found, the process sends the result to the Medical Management System, which sends it further to the user.

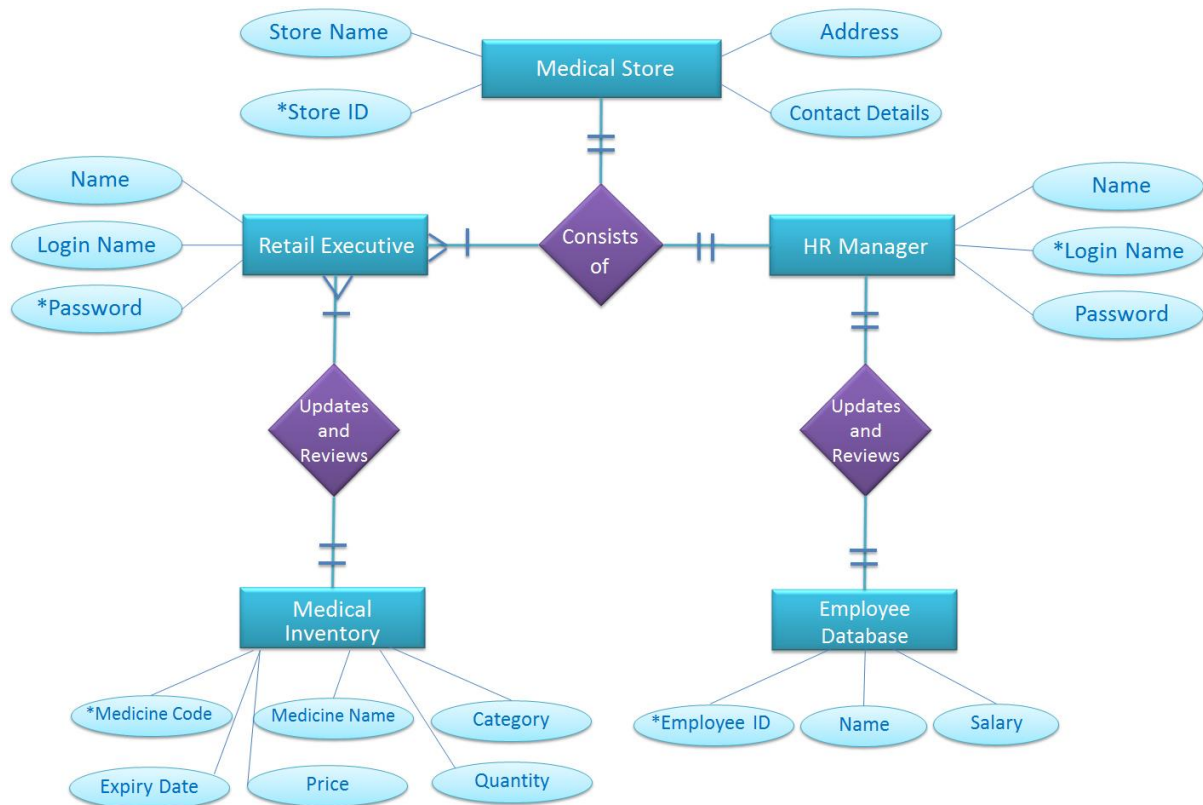


Copyright © 2015 by the 'G-SRS Team'. All rights reserved.

Data Flow Diagram – Level 2 – Search Function

The Level 2 for the Display Function works in the same way till the Display Function Processing. Here, the process accesses all the records of the Database. It then sends all these records to the Medical Management System, which sends them further to the user, thus displaying all records.

4.1.2 Entity Relationship Diagram



Note: All unique attribute of entities have been marked with an asterisk (*).

Copyright  2015 by the 'G-SRS Team'. All rights reserved.

Entity Relationship Diagram for the Medical Management System

1) ENTITIES INVOLVED:

- i) Medical Store.
- ii) Retail Executive.
- iii) H-R Manager.
- iv) Medical Inventory.
- v) Employee Database.

2) ATTRIBUTES OF EACH ENTITY:

- i) Medical Store: - Store name, Address, Contact details, Store id.
- ii) Retail Executive: - Name, Login name, Password.
- iii) H-R Manager: - Name, Login name, Password.
- iv) Medical Inventory: - medicine code, medicine name, category, expiry date, price, quantity.
- v) Employee Database: - Employee id, Name, Salary.

3) PROCESSES INVOLVED:

- i) The medical store CONSISTS OF a retail executive and a H-R Manager.
- ii) The retail executive UPDATES AND REVIEWS the medical inventory.
- iii) The H-R Manager UPDATES AND REVIEWS the employee database.

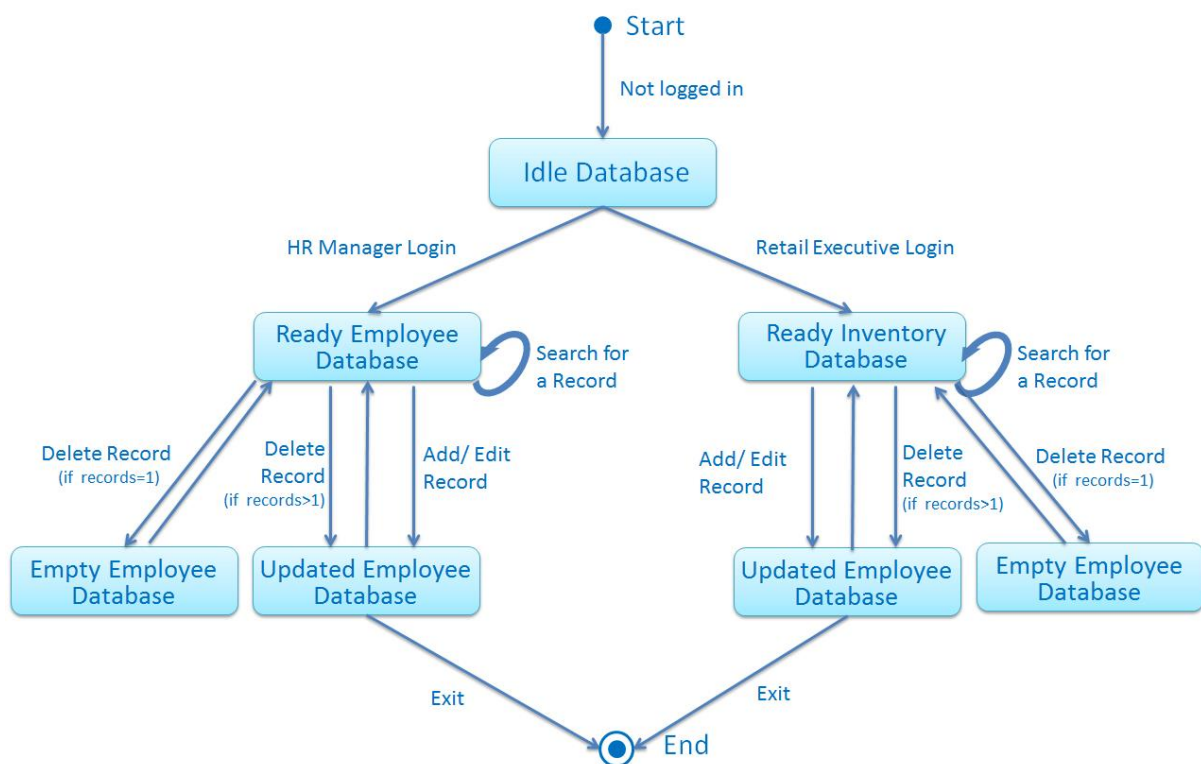
4) CARDINALITY:

- i) The medical store may consist of MANY retail executives.
- ii) The medical store consists of a SINGLE inventory and database.

5) MODALITY:

- i) The medical store can consists of ONLY ONE H-R Manager and ATLEAST ONE retail executive.
- ii) The retail executive is in charge of ONLY ONE inventory.
- iii) The H-R Manager is in charge of ONLY ONE employee database.

4.1.3 State Transition Diagram



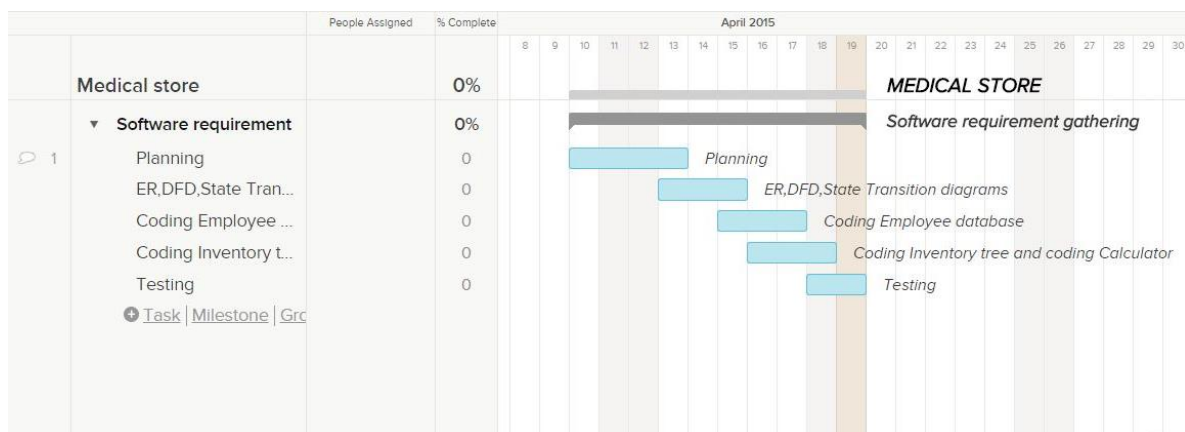
Copyright © 2015 by the 'G-SRS Team'. All rights reserved.

State Transition Diagram for the Medical Management System

The states followed in the state transition diagram can be represented in a sequential manner:

1. Start
2. Login
 - a. HR Login leads to a Ready Employee Database
 - b. Retail Manager Login leads to a Ready Inventory
3. In both states mentioned above, the following actions can be taken:
 - a. Add or Edit a record –
 - This leads to the ‘Updated Employee Database’ state.
 - This then returns to the ‘Ready’ state, being ready for more actions.
 - b. Delete a record –
 - If there are more than one records in the database, the state changes in the same manner as described above
 - If there is only one record, the state changes to ‘Empty Database’. This further leads to ‘Ready’ state, being ready for updating.
 - c. Exit – This leads to the ‘End’ state.

4.1.4 Timeline chart



Timeline Chart for Medical Management System

4.2 Software System Attributes

2.1.1 Reliability

In this case, the Medica developed by the G-SRS is developed in a such a manner that the HR Manager and the executive retailer can perform the required functions with precision as such the addition and deletion of the entity and keeping a record of the employee's sales details and its general information in its databases

2.1.2 Availability

The Medica software should be available on normal business hours of the store to carry out various functions of the given medical store

2.1.3 Security

In this case, Medica software has maintained its integrity as the program developed has made it a point that there can be no unauthorized access to the software and the system remains safe from intruders/Trojan/viruses

However we suggest the HR manager to install a premium antivirus and a firewall to keep the system safe.

2.1.4 Maintainability

In this case, Medica software developed for the online medical store management system ensures about the fact if an error possibly occurred in a given line of code, it would require an minimum effort to resolve a particular error.

3.2.5 Portability

In this case, Medica developed by the G-SRS has taken or looked into the fact that in case the officials feel that the required software should be transferred or upgraded in a new hardware than it is the most feasible to do so and ease of use is definitely taken into account

5. Test Cases

| Test Case ID | Test Condition | Inputs | Precondition | Expected Result | Outputs | Solution |
|--------------|----------------|------------------------------|---|-------------------------|--|---|
| TC-01 | Login | <i>Username and Password</i> | <i>Predefined Username and Password</i> | <i>Login successful</i> | <i>Login possible, but can't change password</i> | <i>Allow user to enter new password</i> |
| TC-02 | Login | <i>Username and Password</i> | <i>Multi-User login possible</i> | <i>Login successful</i> | <i>Only 2 user login possible</i> | <i>Allow more users to log in</i> |
| TC-03 | Calculator | Calculator Expression | Calculation done | Calculation done | Prefix expression calculator | Allow calculation of infix expressions |

The Data Structures used in the software include:

1. Linked List for the Retail Manager database of Inventory

- Variables in every record:
 - i. string emp_name - Stores a string for employee's name.
 - ii. int emp_id - Integer variable for storing employee's unique id number.
 - iii. int emp_srno - Integer variable for storing employee serial number based on performance.
 - iv. float emp_salary - Float variable for storing Employee's salary.
 - v. node1 *next - to move to the next node of list.
- Functions:
 - i. void CreateEmpRecord() - To create n number of employee records as entered by user.
 - ii. void AddEmpRecord(string Ename,int Eid,float Esal,int Esno) - To insert a record at beginning,end or in middle.
 - iii. int SearchEmpRecordid(int Eid) - To search a record by Employee id.
 - iv. int SearchEmpRecordname(string Ename) - To search a record by employee name.
 - v. int SearchEmpRecordSno(int Esno) - To search a record by Employee serial number.
 - vi. void DeleteEmpRecord(int Esno) - To delete employee record based on serial number.

- vii. void UpdateEmpRecord(int Eid) - To update employee records by searching them from employee id.
- viii. void DisplayEmpRecord() - To display employee records present in list.

2. BINARY SEARCH TREE FOR INVENTORY

- Data Members:
 - i. int medicine_serial_no - Variable for storing medicine serial number .
 - ii. string medicine_category - Variable for storing medicine category.
 - iii. float medicine_price - Variable for storing medicine price.
 - iv. int medicine_code - Variable for storing unique code (Primary Key) .
 - v. int quantity_aval - Variable for quantity of a medicine available in inventory.
 - vi. string exp_date - Variable for storing medicine's expiry date.
 - vii. string medicine_name - Variable for storing medicine name.
 - viii. inventory_node* left,*right - node pointers to left and right subtree.
- Functions:
 - i. void insert_item(int n) - To insert a record based on Medicine code using binary search tree property.
 - ii. void delete_item_from_file() - To delete a record from file inventory.
 - iii. void deletenode(int n) - To delete a node from tree based on medicine code.
 - iv. void search_item(int key1) - To search an item in tree based on medicine code.
 - v. inventory_node* search_item(inventory_node *ptr,int key1) - Overloading the search function.
 - vi. int IsEmpty() - To check if tree is empty.
 - vii. void display_inventory() - To display contents of file inventory.
 - viii. void preorder_display_inventory() - To display contents of tree.
 - ix. void preorder_display_inventory(inventory_node*currentnode) - Overloading the preorder display function.

3. BINARY EXPRESSION TREE FOR PREFIX EXPRESSION CALCULATOR

- Data members:
 - i. char data - Variable for storing data of each node(operands:leaf nodes and operators : internal nodes).
 - ii. TreeNode *left, *right - Pointers to left and right subtrees.
- Functions:
 - i. void clear() - Function to clear tree
 - ii. void push(TreeNode *ptr) - Function to push a node to tree.

- iii. `TreeNode *pop()` - Function to pop a node
- iv. `TreeNode *peek()` - Function to get top node
- v. `void insert(char val)` - Function to insert character (data) into tree.
- vi. `bool isDigit(char ch)` - Function to check if character entered is digit (0-9).
- vii. `bool isOperator(char ch)` - Function to check if character entered is operator
- viii. `int toDigit(char ch)` - Function to convert character to digit.
- ix. `double evaluate()` - Function to evaluate (perform arithmetic operations)tree.
- x. `double evaluate(TreeNode *ptr)` - Overloading the function to evaluate tree
- xi. `void buildTree(string eqn)` - Function to build tree from input.
- xii. `void postfix()` - Function to get postfix expression by postorder traversal of tree.
- xiii. `void postOrder(TreeNode *ptr)` - Function to perform post order traversal of tree.
- xiv. `void infix()` - Function to get infix expression by inorder traversal of tree.
- xv. `void inOrder(TreeNode *ptr)` - Function to perform in order traversal of tree.
- xvi. `void prefix()` - Function to get prefix expression by preorder traversal of tree.
- xvii. `void preOrder(TreeNode *ptr)` - Function to perform pre order traversal of tree.

The output of the Program is as follows:

```

Welcome to Medica :

The ultimate medical store management software

Press Enter to continue ....

```

```
***** Login *****
```

```
Enter:
```

```
1 for admin login  
2 for cashier login
```

```
1
```

```
Enter username : ADMIN
```

```
Enter Password : 123456
```

Which operation do you want to perform on Employee database :

Enter your option

- 1 => Add a new record
- 2 => Delete record of particular employee
- 3 => Search Record from Employee id
- 4 => Search Record from Employee name
- 5 => Search Records from Serial(Performance) number
- 6 => Display Employee Records
- 7 => Update Employee Records
- 8 => Create Employee Records
- 9 => Exit from the program

8

Enter the employee's id : 1

Enter the employee's name : ABC

Enter the employee's salary : 12000

Enter the employee's serial no. : 1

Add more employee records : Y or N : Y

Enter the employee's id : 2

Enter the employee's name : CDE

Enter the employee's salary : 13000

Enter the employee's serial no. : 2

Add more employee records : Y or N : Y

Enter the employee's id : 3

Enter the employee's name : FGH

Enter the employee's salary : 15000

Enter the employee's serial no. : 3

Add more employee records : Y or N :

```

***** Menu *****
Which operation do you want to perform on Employee database :
Enter your option
1 => Add a new record
2 => Delete record of particular employee
3 => Search Record from Employee id
4 => Search Record from Employee name
5 => Search Records from Serial<Performance> number
6 => Display Employee Records
7 => Update Employee Records
8 => Create Employee Records
9 => Exit from the program
*****
3
Enter the employee's id : 1
Record found : The record exists
1      ABC      1      12000
Want to perform more operations on employee database : Enter y or n : Y
***** Menu *****
Which operation do you want to perform on Employee database :
Enter your option
1 => Add a new record
2 => Delete record of particular employee
3 => Search Record from Employee id
4 => Search Record from Employee name
5 => Search Records from Serial<Performance> number
6 => Display Employee Records
7 => Update Employee Records
8 => Create Employee Records
9 => Exit from the program
*****
4
Enter the employee's name : ABC
Record found : The record exists already
1      ABC      1      12000
Want to perform more operations on employee database : Enter y or n :

```

```

Which operation do you want to perform on Employee database :

Enter your option
1 => Add a new record
2 => Delete record of particular employee
3 => Search Record from Employee id
4 => Search Record from Employee name
5 => Search Records from Serial(Performance) number
6 => Display Employee Records
7 => Update Employee Records
8 => Create Employee Records
9 => Exit from the program

*****
5

Enter the employee's serial number : 2

Record found : The record exists

2      CDE      2      13000

Want to perform more operations on employee database : Enter y or n : y

***** Menu *****

Which operation do you want to perform on Employee database :

Enter your option
1 => Add a new record
2 => Delete record of particular employee
3 => Search Record from Employee id
4 => Search Record from Employee name
5 => Search Records from Serial(Performance) number
6 => Display Employee Records
7 => Update Employee Records
8 => Create Employee Records
9 => Exit from the program

*****
1

Enter the employee's id : 4

Enter the employee's name : HIJ

Enter the employee's salary : 15000

Enter the employee's serial no. : 4

Record not Found

```

Record not Found

New Record added :

| | | | |
|---|-----|---|-------|
| 1 | ABC | 1 | 12000 |
| 2 | CDE | 2 | 13000 |
| 3 | FGH | 3 | 15000 |
| 4 | HIJ | 4 | 15000 |

Want to perform more operations on employee database : Enter y or n : Y

***** Menu *****

Which operation do you want to perform on Employee database :

Enter your option

- 1 => Add a new record
- 2 => Delete record of particular employee
- 3 => Search Record from Employee id
- 4 => Search Record from Employee name
- 5 => Search Records from Serial<Performance> number
- 6 => Display Employee Records
- 7 => Update Employee Records
- 8 => Create Employee Records
- 9 => Exit from the program

2

Enter the employee's serial number starting from 1 : 1

Record found : The record exists

| | | | |
|---|-----|---|-------|
| 1 | ABC | 1 | 12000 |
|---|-----|---|-------|

Record deleted :

| | | | |
|---|-----|---|-------|
| 2 | CDE | 2 | 13000 |
| 3 | FGH | 3 | 15000 |
| 4 | HIJ | 4 | 15000 |

Want to perform more operations on employee database : Enter y or n :

```

***** Menu *****
Which operation do you want to perform on Employee database :
Enter your option
1 => Add a new record
2 => Delete record of particular employee
3 => Search Record from Employee id
4 => Search Record from Employee name
5 => Search Records from Serial<Performance> number
6 => Display Employee Records
7 => Update Employee Records
8 => Create Employee Records
9 => Exit from the program
*****
7

Enter the employee's id : 2
Record found : The record exists
2      CDE      2      13000

Which field do you want to edit :
Enter :
1.Employee id
2.Employee name
3.Employee Salary
4.Employee srno.
3

Enter Employee Salary : 16700

Record Successfully edited

2      CDE      2      16700
3      FGH      3      15000
4      HIJ      4      15000

Want to perform more operations on employee database : Enter y or n :

```

```

Want to perform more operations on employee database : Enter y or n : Y
***** Menu *****
Which operation do you want to perform on Employee database :
Enter your option
1 => Add a new record
2 => Delete record of particular employee
3 => Search Record from Employee id
4 => Search Record from Employee name
5 => Search Records from Serial<Performance> number
6 => Display Employee Records
7 => Update Employee Records
8 => Create Employee Records
9 => Exit from the program
*****
6

2      CDE      2      16700
3      FGH      3      15000
4      HIJ      4      15000

Want to perform more operations on employee database : Enter y or n : N

```


Want to login from another user account : y or n : Y

***** Login *****

Enter:

1 for admin login
2 for cashier login

2

Enter username : CASHIER

Enter Password : 12345

```

Hello cashier

***** Menu *****
Which operation do you want to perform on Inventory :
Enter
1.Insert a record
2.Delete a record
3.Search a Record
4.View Inventory
5.View nearest dealer's inventory
6.Calculator
7.Exit from program
1
Enter the medicine code for the record to be inserted : 1

***** Insert Record *****

***** Medicine Information Form *****

Enter medicine serial number : 1
Enter medicine category : ANTIPYRETIC
Enter medicine name : CROCIN
Enter medicine expiry date : 2-9-2016
Enter medicine price : 60
Enter medicine quantity : 100

1      1      ANTIPYRETIC      CROCIN      2-9-2016      60      100
Perform more operations on inventory : Y or N : Y

```

```

Perform more operations on inventory : Y or N : Y

***** Menu *****

Which operation do you want to perform on Inventory :
Enter
1.Insert a record
2.Delete a record
3.Search a Record
4.View Inventory
5.View nearest dealer's inventory
6.Calculator
7.Exit from program
1

Enter the medicine code for the record to be inserted : 2

***** Insert Record *****

***** Medicine Information Form *****

Enter medicine serial number : 2
Enter medicine category : ANTISEPTIC
Enter medicine name : DETTOL
Enter medicine expiry date : 3-8-2017
Enter medicine price : 80
Enter medicine quantity : 100

1      1      ANTIPYRETIC      CROCIN      2-9-2016      60      100
2      2      ANTISEPTIC      DETTOL      3-8-2017      80      100
Perform more operations on inventory : Y or N : Y

```

```

Which operation do you want to perform on Inventory :
Enter
1.Insert a record
2.Delete a record
3.Search a Record
4.View Inventory
5.View nearest dealer's inventory
6.Calculator
7.Exit from program
3

Enter medicine code : 2

***** Search Record *****

Element 2 found...
1      1      ANTIPYRETIC      CROCIN      2-9-2016      60      100
2      2      ANTISEPTIC      DETTOL      3-8-2017      80      100

Perform more operations on inventory : Y or N : Y

***** Menu *****

Which operation do you want to perform on Inventory :
Enter
1.Insert a record
2.Delete a record
3.Search a Record
4.View Inventory
5.View nearest dealer's inventory
6.Calculator
7.Exit from program
2

Enter medicine code for the record to be deleted : 1

***** Delete Record *****

The contents of inventory after deletion are :

2      2      ANTISEPTIC      DETTOL      3-8-2017      80      100

```

```

***** Menu *****
Which operation do you want to perform on Inventory :
Enter
1.Insert a record
2.Delete a record
3.Search a Record
4.View Inventory
5.View nearest dealer's inventory
6.Calculator
7.Exit from program
4

The contents of inventory are :

2      2      ANTISEPTIC      DETTOL      3-8-2017      80      100

Perform more operations on inventory : Y or N : Y

***** Menu *****
Which operation do you want to perform on Inventory :
Enter
1.Insert a record
2.Delete a record
3.Search a Record
4.View Inventory
5.View nearest dealer's inventory
6.Calculator
7.Exit from program
5

***** Medicine Inventory *****

1      100      Antiseptics      Dettol      2-02-2017      50      100
2      101      Antipyretic      Crocin      2-10-2016      40      100
3      102      Antacids      Zantac      2-12-2016      30      100
4      103      Analgesics      Aspirin      2-12-2016      20      100
5      103      Anti-biotic      Ofloxacin      2-12-2016      60      100
6      104      Antifertile      Novestrol      2-10-2017      40      100
7      106      Tranquilizer      Equanil      3-8-2016      40      100

Perform more operations on inventory : Y or N : Y

```

```

Perform more operations on inventory : Y or N : Y
***** Menu *****
Which operation do you want to perform on Inventory :
Enter
1.Insert a record
2.Delete a record
3.Search a Record
4.View Inventory
5.View nearest dealer's inventory
6.Calculator
7.Exit from program
6

Expression Tree :
Enter equation in Prefix form: *+234

Prefix : *+234
Infix : 2+3*4
Postfix : 23+4*
Evaluated Result : 20
Perform more operations on inventory : Y or N : N

```

```

Want to login from another user account : y or n : N
-----
Process exited after 1727 seconds with return value 0
Press any key to continue . . .

```