WEB BASED HEALTH CARE SYSTEM

Project Report submitted in partial fulfillment of the requirement

for the degree of

Bachelor of Technology.

in

Computer Science & Engineering

Under the Supervision of

MS. NISHTHA AHUJA

By

Manish Kumar Pachoriya
101237

To

Jaypee University of Information and Technology

Certificate

This is to certify that project report entitled "Web based health care system" Submitted by Manish Kumar Pachoriya (101237) in partial fulfillment for The award of degree of Bachelor of Technology in Computer Science & Engineering to Jaypee University of Information Technology, Waknaghat , Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University Or Institute for the award of this or any other degree or diploma.

Date: **Supervisor's**

Name **Ms. Nishtha Ahuja**

(CSE Deptt.)

Acknowledgement

I would like to thank many people who have helped us directly or indirectly in the successful completion of the project. I would like to thank one and all the core of my heart.

First of all I would like to express my deep sense of gratitude towards my project Guide **Nishtha Ahuja (** CSE Deptt.**)** for always being available whenever I required her guidance as well as for motivating throughout the project work .
I would like to express my deep gratitude towards other teachers and other staff for giving their valuable suggestions and co-operation for doing my project.
I am deeply thankful to Brig. Balbir singh , Director of Jaypee University of information Technology Waknaghat for providing necessary facilities during the execution of this project.

I would like to thank all my friends for their help and constructive criticism during this project period. Finally , I am very much indebted to my parent for their moral support and encouragement to achieve higher goals. I have no words to express my gratitude and still I am very thankful to my parent who have give me their every support.

SIGNATURE

Manish kumar pachoriya

{ 101237}

TABLE OF CONTENT

CHAPTER 1 : INTRODUCTION

1.1. BRIEF OVERVIEW OF HEALTH CARE MANAGEMENT SYSTEM

To develop a Health care Management system, we take care of patient registration, drug information and concerns such as drug enquiries and complaints.

The current manual system is slow laborious and error prone to computerize the same for quicker efficient results and customer satisfaction

1.2 ADVANTAGES TO BOTH END USERS & DEVELOPERS

The system is useful in various ways as the information about the patients who are taking the free services from the health center all the details are already stored in the database , so the service is done in no time . All the information about the drugs are also maintained in the database

1.3 GOAL AND NEED:

GOAL: With every going day the need to be where the inflow of out patient request exceeds that which can be handled manually. Hence computerization of OP receipt request and maintenance of the drugs through the computerization brings better satisfaction and service oriented ness.

Quicker processing of OPNO receipt would mean better service to the patients . It would also help in the complexity of maintaining the records manually and thus less time is wasted on rework. Proper maintenance of the drug information timely dispatching of the drugs from the main stores to the pharmacy and also maintenance of daily dispatching of the drugs to the out patients from the pharmacy to the out patients . Towards this achievement the computerization of the Health Center will help greatly in maintaining pf proper information about the out patients who are eligible for the free services and the patients who are not eligible for the free services , drug information , patients records ,and daily dispatching of the drugs to various patients .

NEED: To develop a Heath Center Management system as from manual system to computerized system, and to take care of Records of the various departments in the health center.The current manual system is slow laborious and error prone to computerize the same for quicker efficient results .

1.4 GENERAL METHODOLOGY IN DEVELOPING S/W PROJECT :

The general methodology in developing a system is involved in different phases, which describe the system's life cycle model for developing software project. The concept includes not only forward motion but also have the possibility to return that is cycle back to an activity previously completed. This cycle back or feedback may occur as a result of the failure with the system to meet a performance objective or as a result of changes in redefinition of system activities. Like most systems, the life cycle of the computer based system also exhibits distinct phases.

Those are,

1. REQUIREMENT ANALYSIS PHASE
2. DESIGN PHASE
3. DEVELOPMENT PHASE
4. CODING PHASE
5. TESTING PHASE

1.4.1. REQUIREMENT ANALYSIS PHASE :

This phase includes the identification of the problem, in order to identify the problem, we have to know information about the problem, the purpose of the evaluation for problem to be known. We have to clearly know about the client's requirements and the objectives of the project.

SYSTEM ANALYSIS PHASE :

Feasibility analysis involves the benefits of various approaches and the determination of the alternative approaches a\through methods like questionnaires and interviews etc., different data about the project is collected and the data through out the project is represented in the form of UML Diagrams.

1.4.2 DESIGN PHASE :

S/W design is a process through which the requirements are translated into a representation of a s/w. One of the software requirements have been analyzed and specified, the s/w design involves three technical activities : design, coding generation and testing. The design of the system is in modular form i.e., the s/w is logically partitioned into components that perform specific functions and sub functions. The design phase leads to modules that exhibit independent functional characteristics. It even leads to interfaces that reduce the complexity of the connections between modules and with the external environment. The design phase is of main importance because in this activity, decisions ultimately affect the success of s/w implementation and maintenance.

1.4.3 DEVELOPMENT PHASE:

The development phase includes choosing of a suitable s/w to solve the particular problem given. The various facilities and the sophistication in the selected s/w give a better development of the problem**.**

1.4.4 CODING PHASE :

The coding phase is for translating the design of the system produced during the design phase into code in a given programming language, which can be executed by a computer and which performs the computation specified by the design.

1.4.5 TESTING PHASE :

Testing is done in various ways such as testing the algorithm, programming code, sample data debugging is also one of following the above testing.

CHAPTER 2 : REQUIREMENT ANALYSIS

2.1 DATA COLLECTION:

Observation of the Existing System:

In the typical Health Center Management System is developed to make all the sections computerized . The entire process is very time consuming and involves tones of paper work- mostly manually, which is both error prone and time consuming.

The new system would have the patients are requesting for the receipts at the Registration office by showing the ID given to the patients that are given at there respected departments at the university level , Just by submitting the opno to the database the person is confirmed that the person is eligible or not . If the person is eligible the service is provided to the person. In the Drug Store the maintenance of the drug information that how many drugs are there in the store and how much is dispatched to the Pharmacy house and how many are in the main stores and what content of the drugs are to be ordered? All this information is in the drug store. The information that how many out- patients that have visited the health center and how many patients are In-patients this information is stored in the Case Records, and the daily dispatching of the drugs from the pharmacy to the patients are maintained in the Daily Records.

2.2 SYSTEM REQUIREMENT SPECIFICATION DOCUMENT

What is SRS?

Software Requirement Specification (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software project is initiated by the client needs. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase.)

The SRS phase consists of two basic activities:

1) Problem/Requirement Analysis:

The process is order and more nebulous of the two, deals with understand the problem, the goal and constraints.

2) Requirement Specification:

Here, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity.

The Requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic goal of this phase.

ROLE OF SRS:

The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium though which the client and user needs are accurately specified. It forms the basis of software

development. A good SRS should satisfy all the parties involved in the system.

2.3 INTRODUCTION :

2.3.1 PURPOSE :

The purpose of this document is to describe all external requirements for The Health Center. It also describes the interfaces for the system.

2.3.2 SCOPE :

This document is the only one that describes the requirements of the system. It is meant for the use by the developers , and will also by the basis for validating the final delivered system. Any changes made to the requirements in the future will have to go through a formal change approval process. The developer is responsible for asking for clarifications, where necessary, and will not make any alterations without the permission of the client.

2.3.1 PROJECT DEFINITION:

The Health care Management System project has been divided into four modules. They are,

1. .Diseases
2. Healthy living
3. Health Corner
4. Tool and Resources
5. Tool & resources
6. Alternative medicine

1. Diseases:

This module consists of the following sub modules viz.,

1.1. Alzheimer's Disease
1.2. Allergy
1.3. Arthritis
1.4. Breast Cancer
1.5. Cerebral palsy
1.6. Diabetes
1.7. Hearing Impairment
1.8. Infertility
1.9. Leptospirosis
1.10. Migraine
1.11. TB
1.12. Alcoholism& Deaddicition
1.13. Amoebiasis
1.14. Asthma
1.15. Cataract
1.16. Acidity
1.17. Appendicitis

1.18. Autism
1.19. Chicken Pox
1.20. HIV and AIDS
1.21. Heart Disease
1.22. Malaria
1.23. Viral Fever

2. Healthy living:

This module is divided into 14 sub modules. They are

2.1. Beauty House
2.2. Diet& Nutrition
2.3. First Aid
2.4. Health Calendar
2.5. Immunization
2.6. Oral care
2.7. Personal Hygine
2.8. Child care
2.9. Eye care
2.10. Fitness
2.11. Home Hygiene
2.12. Insurance
2.13. Legal Matters
2.14. Organ donation

3. Health Corner:

This module has been divided into three sub modules. They are

3.1.Aids corner

3.2. Cancer Corner

3.3. Cardiac Corner

3.4. Diabetes Corner

3.5. Mental Health Corner

3.6. Neuro Center

3.7. Surgeon's Corner

3.8. Women's Corner

4. Tool and Resources:

This module again is divided into sub modules. They are

4.1. Health Calculators

4.2.Medical calculators

4.3.Online Consultation

4.4.Medical record

4.5.Health Record

5. Community and Advice

This module again is divided into sub modules. They are

5.1 Spiritual Support

5.2 Expert speak

5.3 Health Longe

6. Alternative Medicine

This module again is divided into sub modules. They are

6.1 Acupuncture

6.2 Atma Gnana

6.3 Ayurveda

6.4 Homeopathy

6.5 Reiki

6.6 Siddha

6.7 Yoga

Software Requirements:

Platform - Windows (2000/XP) /Unix/Solaris

Languages used - HTML , CSS , JAVASCRIPT , MYSQL

Hardware Requirements:

RAM - 256 MB

Hard Disk - 40 GB

Keyboard - 101 keys

Mouse - Any pointing device

Design Constraints:

This Health Care Management System require huge resources as Hundreds of the patients will require the services instantly, quick response time are needed. The database should also be very large and robust to maintain very huge patients and drugs data.

CHAPTER 3: SYSTEM ANALYSIS

3.1. MODULE DESCRIPTION

This section attempts to describe each module of the project in brief, and the detailed description of each of these modules is spread throughout this document.

3.2. FEASIBILITY ANALYSIS

Feasibility study is an important phase in the software development process. It enables the developer to have an assessment of the product being developed. It refers to the feasibility study of the product in terms of outcomes of the product, operational use and technical support required for implementing it.

Feasibility study should be performed on the basis of various criteria and parameters. The various feasibility studies are:

1. Economic Feasibility
2. Operational Feasibility
3. Technical Feasibility

Economic Feasibility:

It refers to the benefits or outcomes we are deriving from the product as compared to the total cost we are spending for developing the product. If the benefits are more or less the same as the older system, then it is not feasible to develop the product.

Operational Feasibility:

It refers to the feasibility of the product to be operational. Some products may work very well at design and implementation but may fail in the real time environment. It includes the study of additional human resource required and their technical expertise.

Technical Feasibility:

It refers to whether the software that is available in the market fully supports the present application. It studies the pros and cons of using a particular software for the development and its feasibility. It also studies the additional training needed to be given to the people to make the application work.

Implementation Plan:

The main plan for the system developed is to mimic the existing system as it is in the proposed system.

Study of the Existing System

The existing system is very complex as every work is done manually. By using the present system, work is done manually. So, each and every work takes much time to complete. Whenever the doctor needs the information it is very difficult for the employee to search for that particular opno details and the drug information to be ordered. Every time we should search the records at the shelves.

THE PROPOSED SYSTEM:

The present system has obvious problems, inhibiting growth and more usage of man power. The present system which has been proposed is very easy to work .The computerization of the every department in the health center will reduce the work that is done manually. The man power is reduced to the maximum extent. The patients at the registration office are registered within no time, because every time there is no need search for the particular opno in the shelf's .The drugs information are maintained without any complexity and all the calculations are made automatically by this system there is no need for the calculations

Goals and Objectives:

1. Service should be provided to patients in an efficient manner.
2. OPNO number receipt is issued instantly when patient apply for OP receipt.
3. Enquiry details about the drugs are to be maintained in the proper way etc.
4. Daily records are maintained such that the drugs are taken from the MAIN STORES are dispatched in the proper way.
5. Each and every patient record should be maintained in systematic manner so that the searching process will be easy.

CHAPTER 4 : DESIGN PHASE

4.1. INTRODUCTION :

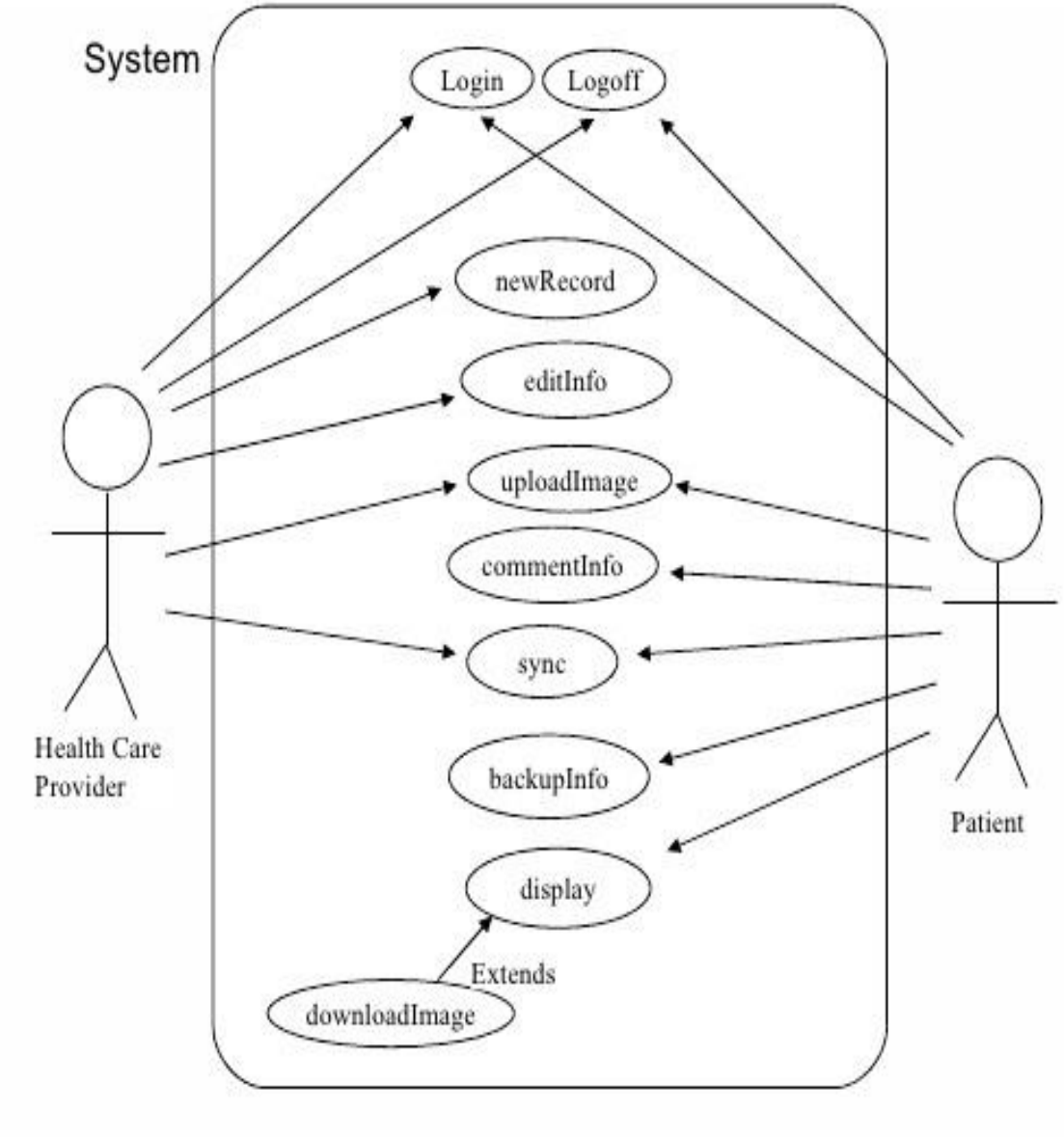
Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

Once the software requirements have been analyzed and specified the software design involves three technical activities design, coding, generation and testing that are required to build and verify the software.

The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer's requirements into finished software or a system.

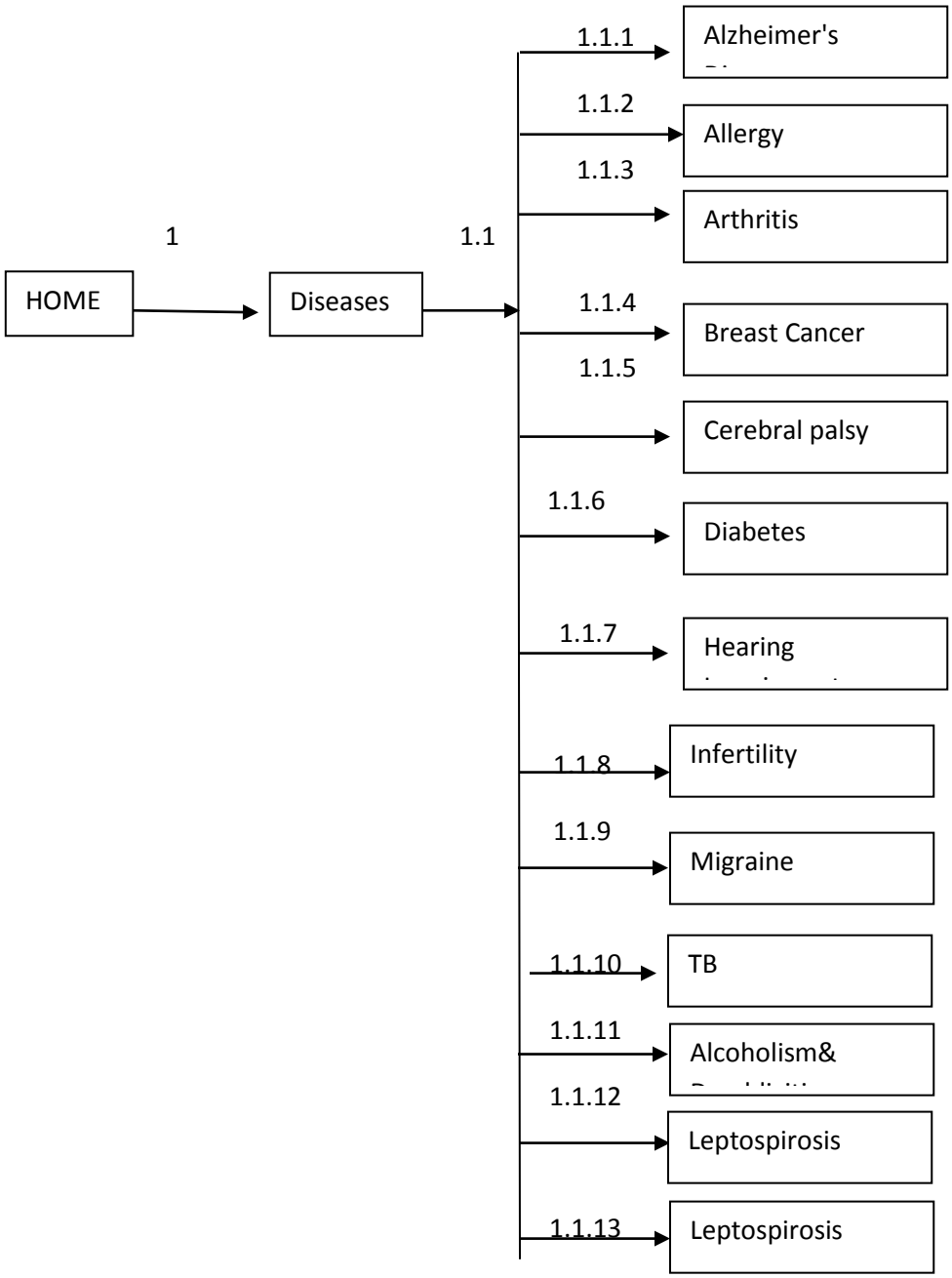
Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data.

4.2 USECASE DIAGRAM

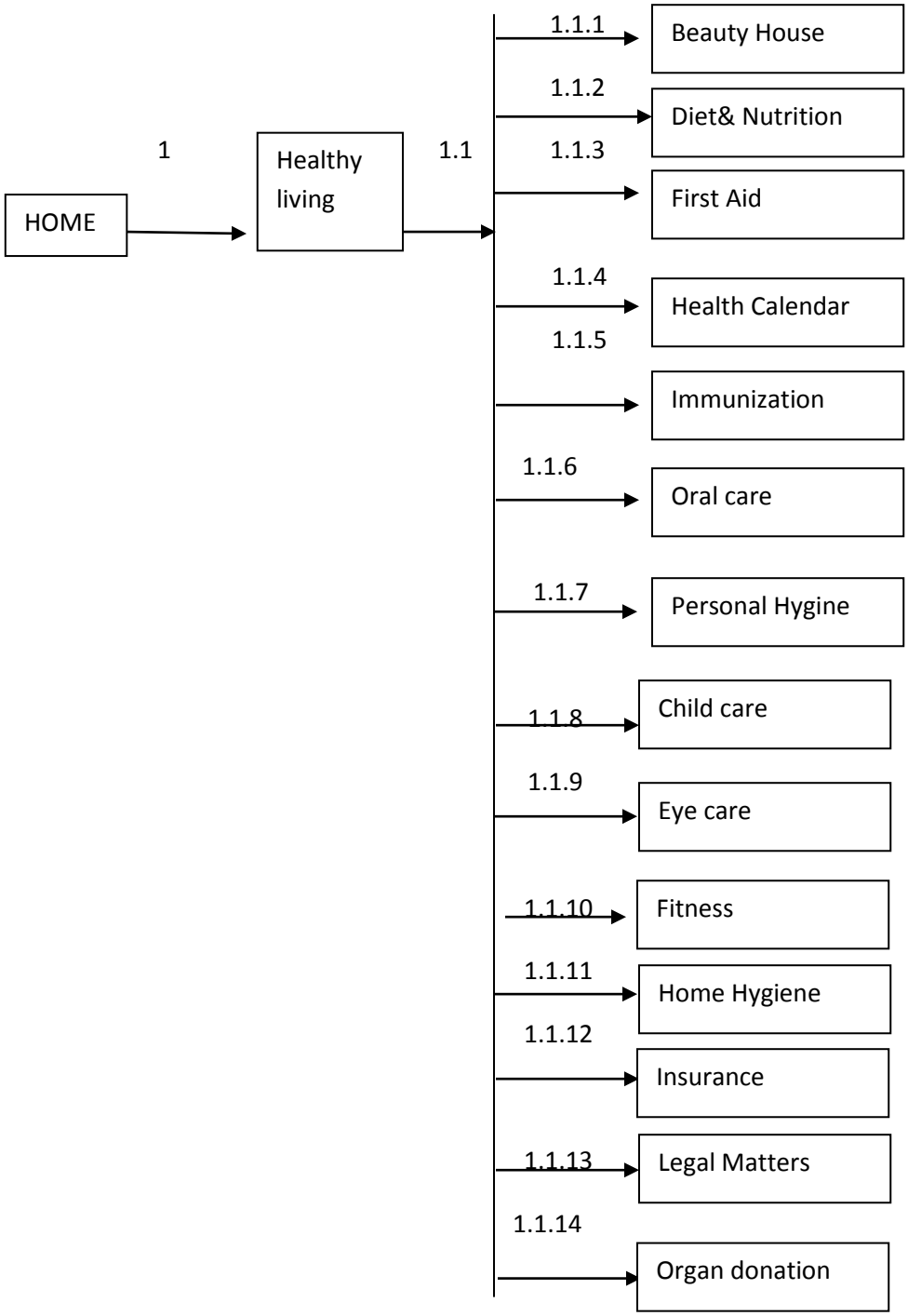


4.3 COLLABORATION DIAGRAM

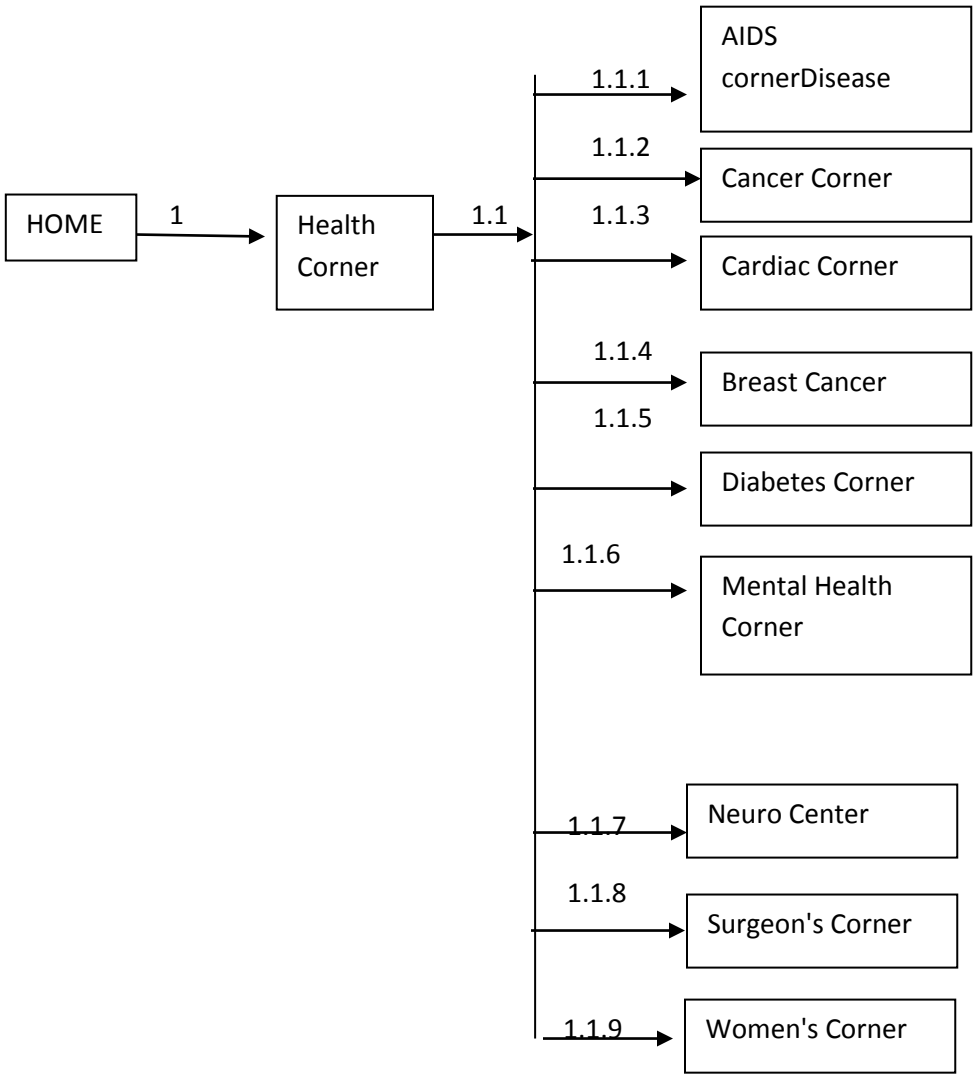
Collarbotion Diagram 1:



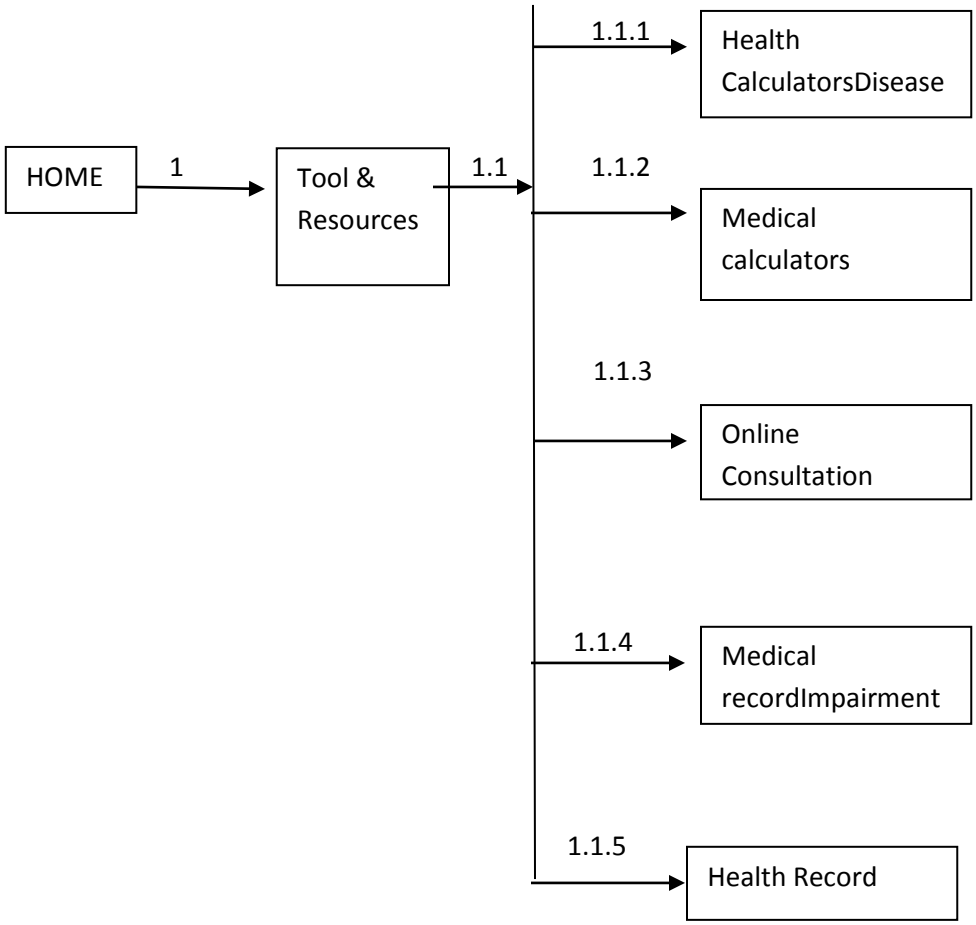
Collaboration diagram 2:



Collaboration diagram 3:



Collaboration diagram 4:



CHAPTER 5: DEVELOPMENT PHASE

5.1 FEATURES OF LANGUAGE

5.1.1 HTML

What is HTML?

HTML is a language for describing web pages.

- HTML stands for Hyper Text Markup Language
- HTML is not a programming language, it is a markup language
- A markup language is a set of markup tags
- HTML uses markup tags to describe web pages

HTML Tags

HTML markup tags are usually called HTML tags

- HTML tags are keywords surrounded by angle brackets like <html>
- HTML tags normally come in pairs like and
- The first tag in a pair is the start tag, the second tag is the end tag
- Start and end tags are also called opening tags and closing tags

HTML Documents = Web Pages

- HTML documents describe web pages
- HTML documents contain HTML tags and plain text
- HTML documents are also called web pages

The purpose of a web browser (like Internet Explorer or Firefox) is to read HTML documents and display them as web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page:

What You Need

You don't need any tools to learn HTML at W3Schools.

- You don't need any HTML editor
- You don't need a web server
- You don't need a web site

Editing HTML

In this tutorial we use a plain text editor (like Notepad) to edit HTML. We believe this is the best way to learn HTML.

However, professional web developers often prefer HTML editors like FrontPage or Dreamweaver, instead of writing plain text.

.HTM or .HTML File Extension?

When you save an HTML file, you can use either the .htm or the .html file extension. We use .htm in our examples. It is a habit from the past, when the software only allowed three letters in file extensions.

With new software it is perfectly safe to use .html.

HTML Headings

HTML headings are defined with the <h1> to <h6> tags.

<h1>This is a heading</h1>
<h2>This is a heading</h2>
<h3>This is a heading</h3>

HTML Elements

An HTML element is everything from the start tag to the end tag:

Start tag *	Element content	End tag *
	This is a paragraph	
	This is a link	

Empty HTML Elements

HTML elements with no content are called empty elements. Empty elements can be closed in the start tag.

 is an empty element without a closing tag (the
 tag defines a line break).

In XHTML, XML, and future versions of HTML, all elements must be closed.

Adding a slash to the start tag, like
, is the proper way of closing empty elements, accepted by HTML, XHTML and XML.

Even if
 works in all browsers, writing
 instead is more future proof.

HTML Attributes

- HTML elements can have attributes
- Attributes provide additional information about an element
- Attributes are always specified in the start tag

Attributes come in name/value pairs like: name="value"

HTML Text Formatting

This text is bold

This text is big

This text is italic

```
This is computer output
```

This is _{subscript} and ^{superscript}

The HTML Style Attribute

The purpose of the style attribute is:

To provide a common way to style all HTML elements.

Styles was introduced with HTML 4, as the new and preferred way to style HTML elements. With HTML styles, styles can be added to HTML elements directly by using the style attribute, or indirectly in separate style sheets (CSS files).

HTML Hyperlinks (Links)

A hyperlink (or link) is a word, group of words, or image that you can click on to jump to a new document or a new section within the current document.

When you move the cursor over a link in a Web page, the arrow will turn into a little hand.

Links are specified in HTML using the <a> tag.

The <a> tag can be used in two ways:

1. To create a link to another document, by using the href attribute
2. To create a bookmark inside a document, by using the name attribute

Link text

HTML Links - The name Attribute

The name attribute specifies the name of an anchor.

The name attribute is used to create a bookmark inside an HTML document.

Bookmarks are not displayed in any special way. They are invisible to the reader.

Useful Tips Section

HTML The <img> Tag and the Src Attribute

In HTML, images are defined with the <img> tag.

The <img> tag is empty, which means that it contains attributes only, and has no closing tag.

To display an image on a page, you need to use the src attribute. Src stands for "source". The value of the src attribute is the URL of the image you want to display.

Syntax for defining an image:

HTML Tables

Tables are defined with the <table> tag.

A table is divided into rows (with the <tr> tag), and each row is divided into data cells (with the <td> tag). td stands for "table data," and holds the content of a data cell. A <td> tag can contain text, links, images, lists, forms, other tables, etc.

HTML Lists:

Unordered Lists

An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.

The list items are marked with bullets (typically small black circles).

Ordered Lists

An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.

The list items are marked with numbers.

Definition Lists

A definition list is a list of items, with a description of each item.

The <dl> tag defines a definition list.

The <dl> tag is used in conjunction with <dt> (defines the item in the list) and <dd> (describes the item in the list):

HTML Forms

HTML forms are used to pass data to a server.

A form can contain input elements like text fields, checkboxes, radio-buttons, submit buttons and more. A form can also contain select lists, textarea, fieldset, legend, and label elements.

HTML Forms - The Input Element

The most important form element is the input element.

The input element is used to select user information.

An input element can vary in many ways, depending on the type attribute. An input element can be of type text field, checkbox, password, radio button, submit button, and more.

The most used input types are described below.

Text Fields

<input type="text" /> defines a one-line input field that a user can enter text into:

5.1.2 CSS :

What Exactly *is* CSS?

Cascading Style Sheets —or what everyone calls *CSS*— is a web language that you can use in conjunction with HTML to endow it with many, *many* extra properties. Basically, CSS will turn your plain old HTML into a smorgusbord of special effects. For instance, if you've ever wondered how to:

- Remove the underlining of hyperlinks
- Change the borders on text boxes in web page forms
- Make your web page background fixed in Firefox
- Set the margin around web pages to 0 (or anything you like)
- Apply padding to only one side of a table cell (or any block element for that

...and more, then CSS is the right ingredient for you.

Why You Should Use CSS

If you're a serious web designer then here's some very practical reasons why you *should* use CSS:

Easily edit the formatting of multiple web pages:

This is the #1 most compelling reason why you should use CSS in your web pages (although some web standards gurus will probably disagree). If you want to keep a uniform look across many web pages and you want to be able to change that look instantly without having to go through each and every page one by one, an external style sheet is the ticket for you. An external style sheet (a.k.a., *linked* style sheet) is a file that contains only CSS code and will allow you to change the formatting to multiple web pages at once. to put this into perspective, let's say that you have a website with 100 pages (or more) and you decided to make the headings green on every page. With plain old HTML, you'd be using heading tags along with the font element to produce something like this:

```
<h1><font color="green">Your Heading</font></h1>
```

But hang on a sec. It's a couple of weeks later and suddenly you've decided that you want to make the headings on all your web pages *maroon* instead of green. With HTML, you'd have to laboriously plow through all 100 web pages, one by one, to change the value of the color attribute on your headings tags (or use some kind of potentially dangerous multiple file search-and-replace function).

On the other hand, if all your web pages were using an external style sheet, you'd be able to change the heading color on all 100 pages in about 5 seconds by editing just a single .css file. Now does that sound like something you'd like to be able to do?

Make your web pages load faster:

If you're using the same external style sheet to build all the web pages at your site, when users come to visit, their web browsers will *cache* that style sheet (keep it in memory) so that it doesn't have to be reloaded every time they click through to another page. In essence, once the style sheet is cached, all the pages at your site will load faster in the user's browser. And we all know how important speed is on the internet, right?

Make table less designs:

As of this writing, using HTML tables to layout web pages is a method that is still very much alive, but if you've dabbled in this then you've probably heard tell of a so-called 'table less' design.

And what is it?

Well essentially, a table less design is a web page that uses HTML and CSS to create a multi-column layout *without* using HTML tables (yes, this is possible). And it's a sure bet that everybody and their monkey's uncle that mentions 'tableless design' will also quickly follow it with some kind of pious reference to supporting 'web standards'

Support web standards:

And last but not least, yes, using CSS in your web pages will support the international efforts being made (ostensibly) to clean up the internet by promoting so-called 'semantic markup' or web pages created using just structural HTML elements and attributes. All the presentational HTML elements and attributes should be replaced with CSS. The idea here is that HTML elements and attributes should define only how a web page is structured (e.g., this is a paragraph, this is a list item, this is a table defining genuine tabular data instead of page layout). All HTML elements and attributes normally used for presentation (e.g., this is Times New Roman font, this is red, this is aligned to the left) should be relegated to CSS.
Then, since different style sheets can be applied to the same web page depending on the circumstances, the web page becomes much more versatile and becomes more accessible to different mediums such as small screen devices, print devices,

How to Use CSS - An Overview

Essentially, CSS is applied to the HTML elements in your web pages and you can do this using any of three different methods:

• Inline Styles

This applies CSS directly to each individual HTML element on your web page using the style attribute. This is the easiest CSS to learn and visualize. It's also handy for quickly testing out CSS properties you're not familiar with or for applying styles that you're not likely to repeat elsewhere.

• Embedded Style Sheets

This applies CSS to the current page. It primarily consists of a list of CSS *rule sets* placed between a set of <style>...</style> tags which themselves are inserted in between the <head>...</head> tags of your web page. Here the real power of CSS starts to come in to play as you can apply the same styles to multiple HTML elements at once on the current page.

• External Style Sheets

This is a list of CSS *rule sets* that are stored in an external file that carries the .css extension. It applies styling to all web pages that are linked to it using the link element in the <head>...</head> tags. This is the most powerful form of CSS as a single .css file can control the formatting of many web pages at once.

Any one or all three of the above methods can be applied to the same web page at the same time. Each method not only defines a manner of applying CSS but also carries with it a certain precedence which gives web authors some control over which style applies when more than one method targets the same HTML element.

Generally speaking, this CSS pecking order can be described as follows:

• Inline styles are the boss

When more than one method applies to the same HTML element, the inline style wins and its style rules are applied to the HTML element.

• Embedded style sheets are second in command

When no inline style exists for any particular HTML element, rule sets in the embedded style sheet which target that HTML element are applied.

• External style sheets come up the rear

In the absence of any inline styles or rule sets in an embedded style sheet for any particular HTML element, rule sets in an external style sheet which target that HTML element are applied.
The precedence of each method thus 'cascades' down through this pecking order and hence you have *Cascading* Style Sheets.

Inline Styles

Using *inline styles* is the best way to start learning CSS as it applies the styling directly to individual HTML elements using the style attribute. This is a no-nonsense, self-contained method that is easy to visualize and leaves little room for error and therefore lends itself to experimenting with new CSS properties and values that you are not familiar with. Inline styles are also good for style declarations that you don't use very often and would just ultimately clutter up your embedded or external style sheets for no good reason. Here's the syntax you use to create an inline style:

```
<element-name style="property: value;">
```

5.1.3 JAVA SCRIPT

What is JavaScript?

- JavaScript was designed to add interactivity to HTML pages
- JavaScript is a scripting language
- A scripting language is a lightweight programming language
- JavaScript is usually embedded directly into HTML pages
- JavaScript is an interpreted language (means that scripts execute without preliminary compilation)
- Everyone can use JavaScript without purchasing a license

Are Java and JavaScript the same?

NO!

Java and JavaScript are two completely different languages in both concept and design!

Java (developed by Sun Microsystems) is a powerful and much more complex programming language - in the same category as C and C++.

What can a JavaScript do?

- **JavaScript gives HTML designers a programming tool -** HTML authors are normally not programmers, but JavaScript is a scripting language with a very simple syntax! Almost anyone can put small "snippets" of code into their HTML pages
- **JavaScript can put dynamic text into an HTML page -** A JavaScript statement like this: document.write("<h1>" + name + "</h1>") can write a variable text into an HTML page
- **JavaScript can react to events -** A JavaScript can be set to execute when something happens, like when a page has finished loading or when a user clicks on an HTML element
- **JavaScript can read and write HTML elements -** A JavaScript can read and change the content of an HTML element
- **JavaScript can be used to validate data -** A JavaScript can be used to validate form data before it is submitted to a server. This saves the server from extra processing
- **JavaScript can be used to detect the visitor's browser** - A JavaScript can be used to detect the visitor's browser, and - depending on the browser - load another page specifically designed for that browser
- **JavaScript can be used to create cookies** - A JavaScript can be used to store and retrieve information on the visitor's computer

The Real Name is ECMAScript

JavaScript's official name is ECMAScript.

ECMAScript is developed and maintained by the ECMA organization.

ECMA-262 is the official JavaScript standard.

The language was invented by Brendan Eich at Netscape (with Navigator 2.0), and has appeared in all Netscape and Microsoft browsers since 1996.

The development of ECMA-262 started in 1996, and the first edition of was adopted by the ECMA General Assembly in June 1997.

The standard was approved as an international ISO (ISO/IEC 16262) standard in 1998.

The development of the standard is still in progress.

Where to Put the JavaScript

JavaScripts in a page will be executed immediately while the page loads into the browser. This is not always what we want. Sometimes we want to execute a script when a page loads, or at a later event, such as when a user clicks a button. When this is the case we put the script inside a function, you will learn about functions in a later chapter.

Scripts in <head>

Scripts to be executed when they are called, or when an event is triggered, are placed in functions. Put your functions in the head section, this way they are all in one place, and they do not interfere with page content.

Scripts in <body>

If you don't want your script to be placed inside a function, or if your script should write page content, it should be placed in the body section.

Using an External JavaScript

If you want to run the same JavaScript on several pages, without having to write the same script on every page, you can write a JavaScript in an external file. Save the external JavaScript file with a .js file extension. **Note:** The external script cannot contain the <script></script> tags!

CHAPTER 6: TESTING

6.1. LIFE CYCLE MODELS

The stage of planning the development process involves defining a define, develop, test, deliver, operate, and maintain a software product. Different lifecycle models emphasize different aspects of the lifecycle and no single lifecycle model is suitable for all software products. A lifecycle model that is understood and accepted by all concerned parties improves project communication and enhances project manageability, resource allocation, cost control, and product quality.

The Phased Life Cycle Model

The phased lifecycle model represents software lifecycle as a series of successive activities. Each phase requires well-defined input information, utilizes well-defined processes and results in well-defined products. The phased model consists of following phases.

Analysis, Design, Implementation, System Testing and Maintenance

This model is sometimes called the Waterfall Chart, the implication being that the products cascade from one level to another in smooth progression.

Analysis	Design	Implementation	System Testing	Maintenance

Planning, User needs Definition	Design Details	Code, debug and Test	Integration & Acceptance	Enhance, Fix Adapt

The **Analysis Stage** consists of Planning and Requirements definition Major include understanding the customer's problem, performing A feasibility study, developing a recommended solution strategy, determining the acceptance criteria and planning development process. The products of planning are a System definition and a project plan.

The **Software Design** follows analysis. Design is concerned with its software components, specifying relationships among components specifying some structure, maintaining a record of design decisions and providing blueprint implementation phase. Design consists of detailed design and Architectural design.

The **implementation** phase of software development involves translation design specification into source code, and debugging, documentation and unit testing the source code. To enhance the quality of the software the methods are structured control constructs, built in and user defined data types, secure type checking, flexible scope rules

exception handling mechanism, concurrency constructs and separates compilation modules.

System Testing involves two kinds of testing integration testing and acceptance testing. Developing a strategy for integrating the components of a software system into a functioning whole requires careful planning so that modules are available for integration when needed. Acceptance testing involves planning and execution of various tests in order to demonstrate that the implemented system satisfies the requirements document.

The **Maintenance** phase comes after the acceptance by the customer and release of the system for production work. Maintenance activities include enhancements of capabilities, adaptation of software to new processing environments, and correction of software bugs.

This project follows the Phased Life Cycle Model or the Water Fall model to a large extent.

The analysis stage consisted of listening to the needs and requirements of all departments obtaining the required format of the system as desired by them, taking the required data to be stored for future use etc., In the design stage the structure of the system was designed and all the required screens were formatted. This was then shown to the medical officer's approval and the system was built. Implementation phase was also done at as they provided a computer with all the required software and with required configuration. The coding and debugging was done even after this stage certain changes were made as made as requested by the guide. The testing was done to check for any errors or bugs or unwanted behavior in the system. Individual modules as well as the whole system were tested separately.

6.2 SOFTWARE TESTING

Software Testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding, Testing presents an interesting anomaly for the software engineer.

Testing Objectives include:

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

Testing Principles:

- All tests should be traceable to end user requirements
- Tests should be planned long before testing begins
- Testing should begin on a small scale and progress towards testing in large
- Exhaustive testing is not possible
- To be most effective testing should be conducted by a independent third party

6.3 TESTING STRATERGIES

A Strategy for software testing integrates software test cases into a series of well planned steps that result in the successful construction of software. Software testing is a broader topic for what is referred to as Verification and Validation. Verification refers to the set of activities that ensure that the software correctly implements a specific function Validation refers he set of activities that ensure that the software that has been built is traceable to customer's requirements

Unit Testing:

Unit testing focuses verification effort on the smallest unit of software design that is the module. Using procedural design description as a guide, important control paths are tested to uncover errors within the boundaries of the module. The unit test is normally white box testing oriented and the step can be conducted in parallel for multiple modules.

Integration Testing:

Integration testing is a systematic technique for constructing the program structure while conducting test to uncover errors associated with the interfacing. The objective is to take unit tested methods and build a program structure that has been dictated by design.

Top-down Integration:

Top down integrations an incremental approach to construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main control program. Modules subordinate to the main program are incorporated in the structure either in the breath-first or depth-first manner.

Bottom-up Integration:

This method as the name suggests, begins construction and testing with atomic modules i.e., modules at the lowest leveling the program structure. Because the modules are integrated in the bottom up manner the processing required for the modules subordinate to a given level is always available and the need for stubs is eliminated.

Validation Testing:

At the end of integration testing software ids completely assembled as a package. Validation testing is the next stage which can be defined as successful when the software functions in the manner reasonably expected by the customer. Reasonable expectations are those defined in the software requirements specifications. Information contained in those sections form a basis for validation testing approach.

System Testing:

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that all system elements have been properly integrated to perform allocated functions.

Performance Testing:

This method is designed to test runtime performance of software within the context of an integrated system..Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. Testing is the exposure of the system to trial input to see whether it produces correct output.

Testing Phases:

Software testing phases include the following:

Test activities are determined and test data selected.

The test is conducted and test results are compared with the expected results.

There are various types of Testing:

Unit Testing:

Unit testing is essentially for the verification of the code produced during the coding phase and the goal is test the internal logic of the module/program.

This project is thoroughly tested by exposing it to the various test cases regarding correct event generation, as this project passed all the tests its quality is completely assured.

Integration Testing:

All the tested modules are combined into sub systems, which are then tested. The goal is to see if the modules are properly integrated, and the emphasis being on the testing interfaces between the modules. On this project integration testing is done mainly while implementing menus in a sample application such as Browser for Mobiles.

System Testing:

It is mainly used if the software meets its requirements. The reference document for this process is the requirement document.

Acceptance Testing:

It is performed with realistic data of the client to demonstrate that the software is working satisfactorily.

Testing Methods:

Testing is a process of executing a program to find out errors. If testing is conducted successfully, it will uncover all the errors in the software. Any testing can be done basing on two ways:

White Box Testing:

It is a test case design method that uses the control structures of the procedural design to derive test cases. using this testing a software Engineer can derive the following test cases:

Exercise all the logical decisions on either true or false sides. Execute all loops at their boundaries and within their operational boundaries. Exercise the internal data structures to assure their validity.

Black Box Testing:

It is a test case design method used on the functional requirements of the software. It will help a software engineer to derive sets of input conditions that will exercise all the functional requirements of the program. Black Box testing attempts to find errors in the following categories:

Incorrect or missing functions

Interface errors

Errors in data structures

Performance errors

Initialization and termination errors

By Black Box Testing we derive a set of test cases that satisfy the following criteria:

Test cases that reduce by a count that is greater than one, the number of additional test cases that must be designed to achieve reasonable testing.

Test cases that tell us something about the presence or absence of classes of errors rather than errors associated only with a specific test at hand.

Test Approach :

Testing can be done in two ways:

Bottom up approach

Top down approach

Bottom up Approach:

Testing can be performed starting from smallest and lowest level modules and proceeding one at a time. For each module in bottom up testing a short program executes the module and provides the needed data so that the module is asked to perform the way it will when embedded with in the larger system. When bottom level modules are tested attention turns to those on the next level that use the lower level ones they are tested individually and then linked with the previously examined lower level modules.

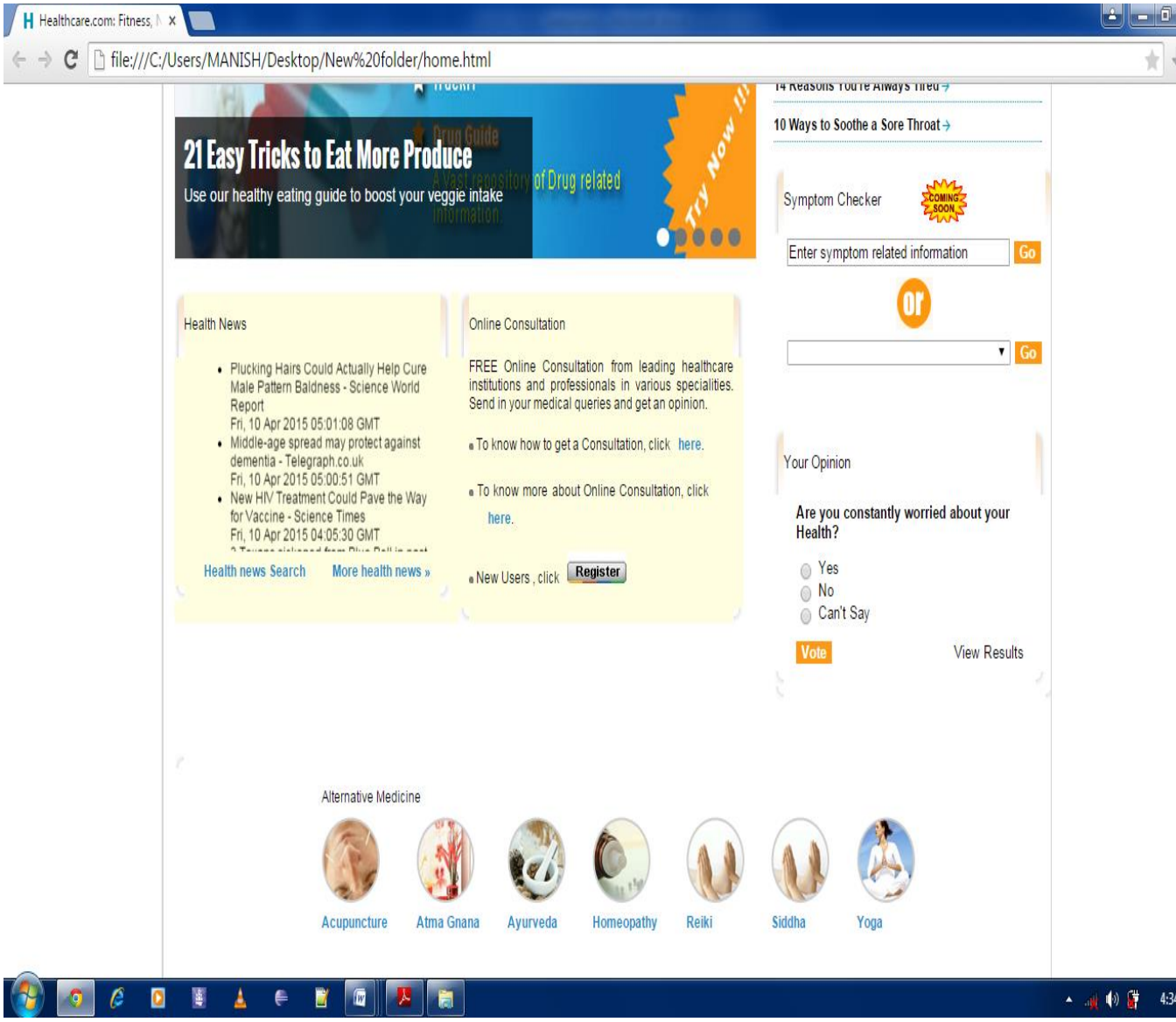
Top down approach:

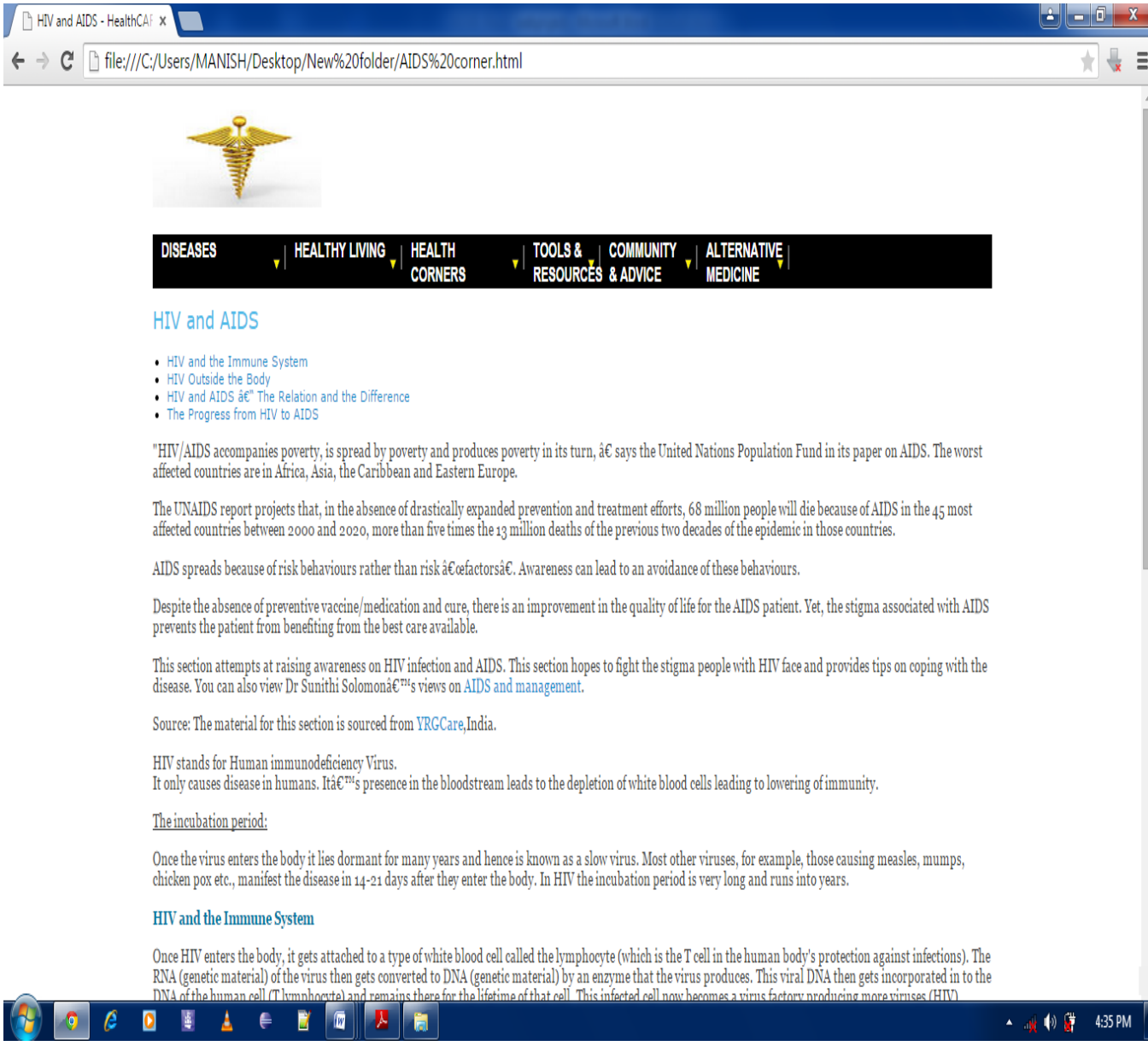
This type of testing starts from upper level modules. Since the detailed activities usually performed in the lower level routines are not provided stubs are written. A stub is a module shell called by upper level module and that when reached properly will return a message to the calling module indicating that proper interaction occurred. No attempt is made to verify the correctness of the lower level module.

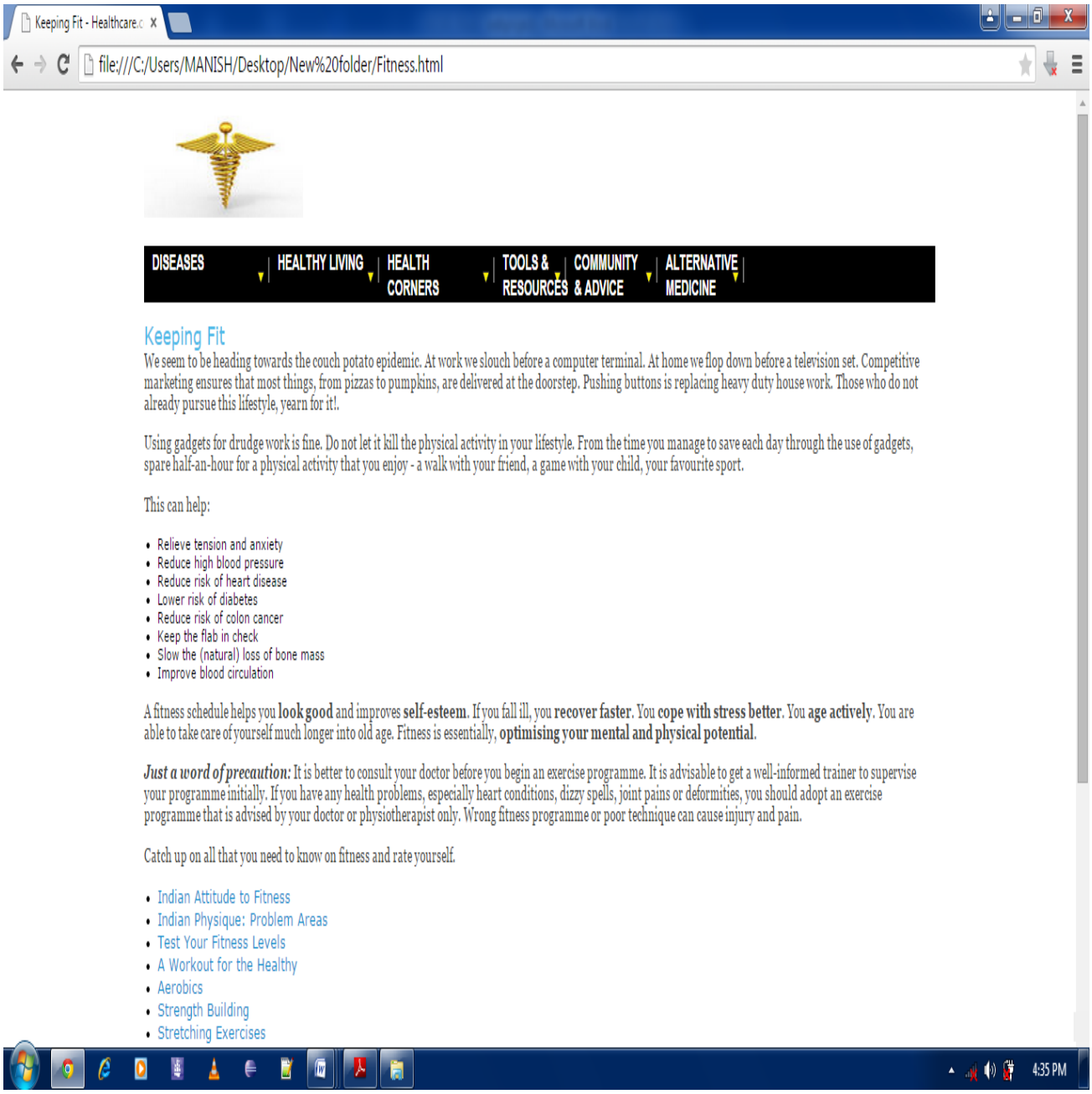
CHAPTER 7: SCREENS

Home page


Healthcare.com: Fitness,
file:///C:/Users/MANISH/Desktop/New%20folder/home.html
Newsletters | Sweepstakes | Subscribe Now to Health Magazine
Search
DISEASES
HEALTHY LIVING
HEALTH CORNERS
TOOLS & RESOURCES
COMMUNITY & ADVICE
ALTERNATIVE MEDICINE
Beauty House
Diet& Nutrition
First Aid
Health Calendar
Immunization
Oral care
Personal Hygine
Child care
Eye care
Fitness
Home Hygiene
Insurance
Legal Matters
Organ donation
th Tools & Resources
h Calculators
cal Calculators
e Consultation
e Health & Medical Record
HT
Drug Guide
A Vast repository of Drug related information
21 Easy Tricks to Eat More Produce
Use our healthy eating guide to boost your veggie intake
Try Now !!!
Connect with Health
Subscribe
Give a Gift
Free Newsletters and More
Enter Email address
Sign up
MOST POPULAR
16 Ways to Lose Weight Fast
Best and Worst Nuts for Your Health
24 Fat-Burning Ab Exercises (No Crunches!)
The Top Fat-Burning Foods
14 Reasons You're Always Tired
10 Ways to Soothe a Sore Throat
Symptom Checker
COMING SOON
Enter symptom related information
Go
or
Health News
Online Consultation
4:33 PM


Healthcare.com: Fitness,
file:///C:/Users/MANISH/Desktop/New%20folder/home.html
21 Easy Tricks to Eat More Produce
Use our healthy eating guide to boost your veggie intake
Drug Guide
A vast repository of Drug related information
Try Now !!!
14 Reasons You're Always Tired
10 Ways to Soothe a Sore Throat
Symptom Checker
COMING SOON
Enter symptom related information
Go
or
Go
Health News
Plucking Hairs Could Actually Help Cure Male Pattern Baldness - Science World Report
Fri, 10 Apr 2015 05:01:08 GMT
Middle-age spread may protect against dementia - Telegraph.co.uk
Fri, 10 Apr 2015 05:00:51 GMT
New HIV Treatment Could Pave the Way for Vaccine - Science Times
Fri, 10 Apr 2015 04:05:30 GMT
Health news Search
More health news »
Online Consultation
FREE Online Consultation from leading healthcare institutions and professionals in various specialities. Send in your medical queries and get an opinion.
To know how to get a Consultation, click here.
To know more about Online Consultation, click here.
New Users , click
Register
Your Opinion
Are you constantly worried about your Health?
Yes
No
Can't Say
Vote
View Results
Alternative Medicine
Acupuncture
Atma Gnana
Ayurveda
Homeopathy
Reiki
Siddha
Yoga


HIV and AIDS - HealthCAF
file:///C:/Users/MANISH/Desktop/New%20folder/AIDS%20corner.html
DISEASES
HEALTHY LIVING
HEALTH CORNERS
TOOLS & RESOURCES
COMMUNITY & ADVICE
ALTERNATIVE MEDICINE
HIV and AIDS
HIV and the Immune System
HIV Outside the Body
HIV and AIDS â€" The Relation and the Difference
The Progress from HIV to AIDS
"HIV/AIDS accompanies poverty, is spread by poverty and produces poverty in its turn, â€ says the United Nations Population Fund in its paper on AIDS. The worst affected countries are in Africa, Asia, the Caribbean and Eastern Europe.
The UNAIDS report projects that, in the absence of drastically expanded prevention and treatment efforts, 68 million people will die because of AIDS in the 45 most affected countries between 2000 and 2020, more than five times the 13 million deaths of the previous two decades of the epidemic in those countries.
AIDS spreads because of risk behaviours rather than risk â€œfactorsâ€. Awareness can lead to an avoidance of these behaviours.
Despite the absence of preventive vaccine/medication and cure, there is an improvement in the quality of life for the AIDS patient. Yet, the stigma associated with AIDS prevents the patient from benefiting from the best care available.
This section attempts at raising awareness on HIV infection and AIDS. This section hopes to fight the stigma people with HIV face and provides tips on coping with the disease. You can also view Dr Sunithi Solomonâ€™s views on AIDS and management.
Source: The material for this section is sourced from YRGCare,India.
HIV stands for Human immunodeficiency Virus.
It only causes disease in humans. Itâ€™s presence in the bloodstream leads to the depletion of white blood cells leading to lowering of immunity.
The incubation period:
Once the virus enters the body it lies dormant for many years and hence is known as a slow virus. Most other viruses, for example, those causing measles, mumps, chicken pox etc., manifest the disease in 14-21 days after they enter the body. In HIV the incubation period is very long and runs into years.
HIV and the Immune System
Once HIV enters the body, it gets attached to a type of white blood cell called the lymphocyte (which is the T cell in the human body's protection against infections). The RNA (genetic material) of the virus then gets converted to DNA (genetic material) by an enzyme that the virus produces. This viral DNA then gets incorporated in to the
4:35 PM


Keeping Fit - Healthcare.c
file:///C:/Users/MANISH/Desktop/New%20folder/Fitness.html
DISEASES
HEALTHY LIVING
HEALTH CORNERS
TOOLS & RESOURCES
COMMUNITY & ADVICE
ALTERNATIVE MEDICINE
Keeping Fit
We seem to be heading towards the couch potato epidemic. At work we slouch before a computer terminal. At home we flop down before a television set. Competitive marketing ensures that most things, from pizzas to pumpkins, are delivered at the doorstep. Pushing buttons is replacing heavy duty house work. Those who do not already pursue this lifestyle, yearn for it!.
Using gadgets for drudge work is fine. Do not let it kill the physical activity in your lifestyle. From the time you manage to save each day through the use of gadgets, spare half-an-hour for a physical activity that you enjoy - a walk with your friend, a game with your child, your favourite sport.
This can help:
Relieve tension and anxiety
Reduce high blood pressure
Reduce risk of heart disease
Lower risk of diabetes
Reduce risk of colon cancer
Keep the flab in check
Slow the (natural) loss of bone mass
Improve blood circulation
A fitness schedule helps you look good and improves self-esteem. If you fall ill, you recover faster. You cope with stress better. You age actively. You are able to take care of yourself much longer into old age. Fitness is essentially, optimising your mental and physical potential.
Just a word of precaution: It is better to consult your doctor before you begin an exercise programme. It is advisable to get a well-informed trainer to supervise your programme initially. If you have any health problems, especially heart conditions, dizzy spells, joint pains or deformities, you should adopt an exercise programme that is advised by your doctor or physiotherapist only. Wrong fitness programme or poor technique can cause injury and pain.
Catch up on all that you need to know on fitness and rate yourself.
Indian Attitude to Fitness
Indian Physique: Problem Areas
Test Your Fitness Levels
A Workout for the Healthy
Aerobics
Strength Building
Stretching Exercises
4:35 PM

CHAPTER 8 : CONCLUSION

The need for the Health care system to computerize the application processing and servicing the Patients request through automated modules is most necessary and now inevitable.

As we have already seen that the need cannot be emphasized for the further development of this system is only timely and helpful to Health Center, the system defined in the above script is up to date and caters to all kinds of request faced by the Health Center employees requirements to provide the better service to the patients, being developed in java it is also flexible modularized highly parameterized and hence can be easily deployed by any other application because of its componentized approach.

Based on the various parameters and properties files everything from the look and feel to the functionalities can be customized. Thus this project is developed from the beginning with reuse in mind and implicitly uses several design patterns. The architecture of this project is such that it suits the diverse and distributed nature of the Health Center Applications.

The features provided by the (Health Care Management System) are in no means comprehensive but by all means full filling all important functionalities of Health Center services. Inclusion of further functionalities as days go by can be easily done because the project has been developed in a layered architecture.

Plug-in modules would easily add new features which change with the times and being performance oriented the project will not face any issues. It is also extensible and

scalable as all applications should be thus it can be said that it will meet surges of huge employee and patient requests that may come up in the near future.

CHAPTER 9 : REFERENCE

[1]. Thomas Powell ,"HTML & CSS: The Complete Reference", Fifth Edition

[2]. Thomas Powell ," Javascript- The Complete Reference", 2nd Edition

[3]. Jarrod ," Developing Web Components".

[4]. Duckett, Jon. *Accessible XHTML and CSS Web Sites Problem Design Solution*, Wrox, 2005.

[5]. Bartlett, Kynn. *Sams Teach Yourself Cascading Style Sheets in 24 Hours, Second Edition*, Sams,2006.

[6]. Crockford, Douglas. *JavaScript: The Good Parts*, O'Reilly & Associates, 2008.

[7]. "Html ", http://www.w3schools.com/html/default.asp

[8]. "CSS", http://www.w3schools.com/css/default.asp

[9]. "JAVASCRIPT",http://www.w3schools.com/js/default.asp

[10]. "Tutorial" http://www.afterhoursprogramming.com/

[11]. http://www.health.com/health/