**INTRODUCTION**

**INTRODUCTION**

My project, Online Medical Store is designed to cater to the very special needs of user/patients. The website introduces a new product, markets it, and also provides feedback from the Doctors, Patients and Chemists on its impact both medical and financial.

Online medical system provides online treatment to the patient along with the home delivery of medicines for that patient have to login to the portal. Patients after login, decide whether to get treatment from doctors or hospitals. Once decided, he/she gets list of diseases, choose the disease, and choose their symptoms. Before treatment is shown to patient, he/she is asked to pay for treatment. Once payment is done then he/she gets prescription. Once he/she gets prescription, he can download it or send it to nearby medical store which is registered onto this portal. If he/she sends prescription to medical store then medical store provides home delivery with cash on delivery service. Medical store has to register onto this portal to access the benefit of selling medicines online. We are using PHP as server scripting and Java script as a client script along with MYSQL for data storage to develop this website. We will use Android studio to develop .ask of this website. This website is going to be beneficial for all patients who want treatment at home and it will also increase business of doctors indirectly as well as medical stores. The main objective of this web portal is to give online treatment. Patients get the advantage of high profile doctors online. Patients get treatment at lower cost online. Medical stores will deliver medicines at home. We decided to choose this medical portal, because it is good concept and not available in India. We have done some related work, like, we have visited some doctors and hospitals regarding how our project will be useful for them. They have approved this concept and are ready to help us.

**ABOUT A Medicine Company:**

The marketing of medicines is very different from that of any other product or service. A drug is never advertised, and never sold without a prescription. The product is invariably marketed through the Medical Management through visits to doctors, via Seminars, discussions and demonstrations, or via advertisements in select Medical Magazines.

**The Present System:**

As described earlier, the present system is time consuming, tedious and error prone. Potentially suspicious entries are hard to detect and the job of seeking clarifications is almost impossible owing to the delays involved.

**Time Consuming Process:**

The process of detecting important trends and other relevant data analysis from the filed reports is done manually, and is very tedious and error prone.

.

**OBJECTIVE:**

When was the last time you went to buy medicines and the pharmacist refused to sell them till you gave an original prescription? I’m sure this has happened but certainly not all the time. But, when was the last time you bought medicines, handed your prescription to the pharmacist, and the pharmacist actually stamped your original prescription and returned it to you? I am confident that the answer to this question would be — NEVER! This brings us to the very interesting debate going on between offline and online chemists. Like any technology or business that questions and tries to change the status quo, online pharmacies are facing extreme resistance from their offline counterparts on the following grounds:   Online pharmacies encourage self-medication and illegal purchase of habit-forming drugs; Patients can purchase prescription drugs without an original prescription; the original prescription can be used for repeated drug purchase. With this, let us look at the various regulations that govern the sale of drugs and then try and evaluate if online pharmacies are really a boon or bane.

The main objective of this project is to provide the medicines in an easy way. The website is overall developed in such a manner that one can easily use and interact with it and he/she can get a good medicine according to their needs.

The main objectives are listed below.

1. The main objective of this website is to users (“**Retailers and Whole sellers**”) can buy various categories of medicines online.
2. User can buy medicine at their home without going anywhere.
3. This website provides so many offers for our Customers (i.e., Whole sellers and Retailers).
4. Helps to consume time management.
5. Customers can ask any queries related to our Medicines.

**INITIAL**

**REQUIREMENTS**

**INTIAL REQUIREMENT:**

When a project is started an initial investigation is carried out. During this phase of study users need has recognized and other requirements are determined. Once the problem has been defined a study is carried out to select the best system i.e. a feasible system that meets performance requirements. So Feasibility is the determination of whether or not a project is worth doing and the process followed in making this determination is called a Feasibility Study. In order to conduct the feasibility study we have seven distinct, but inter-related types of feasibility, these are Technical feasibility, Operational feasibility, Economical feasibility, Social feasibility, Management feasibility, Legal feasibility and Time feasibility. Out of these seven three are key feasibilities to consider, these are:

* Technical Feasibility
* Economical Feasibility
* Operational feasibility

## Technical feasibility:

This is concerned with specifying equipment (hardware) and software that will successfully satisfy the user requirement. It considers the following facts:

The facility to produce outputs in a given time

Response time under certain conditions

Ability to process a certain volume of transaction at a particular speed

Facility to communicate data to distant location

While examining technical feasibility, huge importance is given to the configuration of the proposed system. The configuration should give the complete picture about the system’s requirement such that what kind of hardware is required and how these units are interconnected so that they could operate and communicate smoothly. The proposed system can be run on currently existing software and hardware.

**The minimum required configuration of the system is-**

|  |  |
| --- | --- |
| **Software** | **Hardware** |
| WINDOW 10 | core2 dual |
| Front End Sublime Text | 4GB RAM or More |
| Back End Microsoft Access |  |

**Economical feasibility:**

Since cost plays quite an important role in deciding the new system, it must be identified and estimated properly. So economic analysis is the most frequently used technique for evaluating the effectiveness (economical feasibility) of a proposed system. To determine the economical feasibility of the system a cost/benefit analysis is to make. This procedure is to determine the benefits and savings that are expected from a proposed system and compare them with costs. Four facts that plays an important role in deciding economical feasibility of the proposed system are as follows: Cost-saving benefits, Cost-avoidance benefits, Improved-performance benefits, Improved-information benefits, Hence the proposed system is economically feasible.

## Operational feasibility:

It is mainly related to human organizational aspects. The points to be considered are:

What changes will be brought with the system?

What organizational structures are disturbed?

What new skills will be required?

Do the existing staff members have these skills if not, can they be trained.

**User’s requirement:**

**Software Specification:**

User interface would be used to provide information necessary to generate a new claim also update information.

1. Front End - Sublime Text

2. Backend - MS-Access 2007

3. O/S – Windows 10

**Hardware Specification:**

RAM - 1 GB

Mother board - 31 G

Hard Disk - 1 TB

Processor - 1.8 GHz Dual Core

Mouse - Optical

Keyboard - Odyssey

Monitor - T.F.T. 18” Acer

U.P.S. - Multi-tech

**System Analysis**

**SYSTEM ANALYSIS:**

System analysis is the process of gathering and interpreting facts, diagnosing problems and using the information to recommend improvements on the system. System analysis is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minutest detail and analyzed. The system analyst plays the role of an interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the inputs to the system are identified. The outputs from the organization are traced through the various processing that the inputs phase through in the organization. A detailed study of these processes must be made by various techniques like Interviews, Questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now, the existing system is subjected to close study and the problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as a proposal. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This loop ends as soon as the user is satisfied with the proposal.

Since this is small project and is developed for academic purpose system analysis is not required.

**Existing System Description:**

**Existing System Details:**

This website is very easy to operate it provide database about Institute, Courses and main advantage of this project is that it will generate an environment such that more than one person could solve his problem together , customer have to give query for getting desire solution. . This project has been developed on HTML and Ms Access. It has been developed and implemented for regular use by the user. It is very easy to understand and operate. Hence it is/Will be totally user friendly & interactive website.

**Problem Faced with Existing System:**

The manual system seems to be time consuming and less efficient with respect to the following:

1. Maintenance of large amount of data is a tedious job.
2. Storing all relevant documents concerned with the customers is also a cumbersome job.
3. To obtain details of any customer in case of checkout involves a lot of hard work.
4. No way to check the different queries and services, which may arise at various stages of processing.

**Proposed System:**

In order to make the site dynamic and more interactive we have tried to include a database link to our college website. Hence the recruiters have been provided with the facility to post their eligibility criteria, vacancies and salary packages. In response to which a student can submit his willingness to appear for the drive along with his personal details. Provision has also been made to display the latest events and announcements associated with the college online. We have developed our project using the three tier architecture which uses the following languages.

**Feasible Study with Report:**

The end product, i.e. the documentation after feasibility study report document. It contains the following sections:

Covering report, which briefly indicates the management about the nature, general findings and recommendations to be considered.

Table of contents

Narrative explanation of the purpose and scope of the project, the reason for undertaking feasibility study, and the department involved or affected by candidate system

Detail findings outline the methods used in the present system. Effectiveness, efficiency operating costs, description of objectives and general procedures of the candidate system.

Economic justification details point-to-point cost comparisons and preliminary cost estimates for the development and operation of the candidate system. Return on Investment (ROI) is also given.

Recommendations and conclusion suggest to management the most beneficial and cost effective system.

Appendices document are the memos and data compiled during the investigation.

**Software Requirement Specification**

**Software Requirement Specification:**

A requirements specification for a software system is a complete description of the behavior of a system to be developed. It includes a set of [used cases](http://en.wikipedia.org/wiki/Use_case) that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional (or supplementary) requirements. [Non-functional requirements](http://en.wikipedia.org/wiki/Non-functional_requirements) are requirements which impose constraints on the design or implementation (such as [performance engineering](http://en.wikipedia.org/wiki/Performance_engineering) requirements, [quality](http://en.wikipedia.org/wiki/Quality_%28business%29) standards, or design constraints).

Requirement documentation is very important activity after the requirements elicitation and analysis. This is the way to represent the requirements in a consistent format. Requirement document is called **Software Requirement Specification (SRS).**

The software requirement specification is produced at the culmination of the analysis task. This is the way to represent requirements in a consistent format. It is a specification for a particular software product, program or a set of programs that performs certain functions in a specific environment. The function and allocated to software as part of system engineering are refined by establishing a complete information description, a detailed functional and behavioral description, an indication of performance requirements and design constraints, appropriate validation criteria, and other data pertinent to requirements.

Software product, program or set of programs that perform certain functions in a specific environment. There are two important cases regarding SRS: First one, SRS is used to define the needs and expectations of the users. The second one, SRS is written for different purpose and serve as a contract document between customer and developer. This produces the probability of the customer being disappointment with the final product.

IEEE defines a requirement as:

* A condition of capability needed by a user to solve a problem or achieve an objective.
* A condition or capability that must be met or processed by a system to satisfy a contract, standard, specification, or other formally imposed document.

Generally, the SRS is a document that completely describes what the proposed software should do without describing how the software will do it. The basic goal of the requirements phase is to produce the SRS, which describe the complete external behavior of the proposed software.

**Organization of an SRS:**

The most general organization of an SRS is as follow

* Introduction
  + Purpose
  + Scope
  + Definitions
  + System Overview
* Overall Description
  + Product Perspective
  + Product Functions
  + User Characteristics
  + Constraints, Assumptions and Dependencies
* Specific Requirements
  + External interfaces
  + Functions
  + Performance requirements
  + Logical database requirement
  + Design constraints

An SRS must consist of the following features:-

* consistent
* complete
* unambiguous
* modifiable
* verifiable
* traceable
* ranked

**Need for SRS:**

The SRS is needed for the following reasons:

* An SRS establishes the basis for agreement between client and developer.
* An SRS provides a reference for validation of the final product.
* A high- quality SRS is a prerequisite to high–quality software.
* A high- quality SRS reduces the development cost.

**Platform:**

Windows is very powerful scalable Operating System that provides basic file and prints services as well as robust platform for server applications.

Main features are as follows:-

* More extensive network management features.
* Improved Network Performance.
* Enhanced communication features.

**Objective:**

A software requirement specification is literally the conversation of a specific point. It's difficult in this instance to avoid the circular reference. A project's specifications consist of the body of information that should guide the project developers, engineers, and designers through the work of creating the software. A software requirement specification document describes how something is supposed to be done. A specifications document may list out all of the possible error states for a certain form, along with all of the error messages that should be displayed. The specifications may also describe the steps of any functional interaction, and the order in which they should be followed by the user. A requirements document, on the other hand, would state that the software must handle error states reasonably and effectively, and provide explicit feedback to the users.

**The specifications show how to meet this requirement.**

Specifications may take several forms. They can be a straight forward listing of functional attributes, they can be diagrams or schematics of functional relationships or flow logic, or they can occupy some middle ground. Software Requirement Specifications can also be in the form of prototypes, mockups, and models. Project specifications are much more important for determining the quality of the product. Every rule and functional relationship provides a test point. A disparity between the program and its specification is an error in the program if and only if the software requirement specification exists and is correct. A program that follows a terrible specification perfectly is terrible, not perfect.

**Scope:**

Boundaries of software products are defined by a set of Requirements. The software development team designs, implements, tests, and delivers these Requirements to you. A Requirement is an atomic unit of a software product from the viewpoint of the user. As a rule, Requirements are always correct, unambiguous, verifiable, and traceable. Requirements are numbered and prioritized.

All Functional Requirements are then listed in a requirements attributes spreadsheet, where all necessary attributes for each Requirement are maintained. Changes to the project scope can be made only by issuing new Specifications through a process called Change Requests. Each Change Request implies that changes will be made to the Budget, Schedule, and Risks.

This project is most functional and easy to use solution for prioritizing, planning, managing and executing projects. This project can be used in educational institutions where an analysis of various projects allotted to various scholars is necessary, and also in MNC’s with slight modification to judge the performance of employees. One can also gain some experience based on earlier projects in his upcoming assignments.

**Requirement:**

The software requirements specification is produced at the culmination of the analysis task. This is the way to represent requirements in a consistent format. It is a specification for a particular software product , program or a set of programs that performs certain functions in a specific environment .The function and allocated to software as part of system engineering are refined by establishing a complete information description, a detailed functional and behavioral description, an indication of performance requirements and design constraints, appropriate validation criteria, and other data pertinent to requirements.

**Functional Requirement:**

Functional requirements specify which output should be produced from the given input. They describe the relationship between the input and output of the system. For each functional requirement, a detail description of all the data inputs and their sources, the units of measure, and the range of valid inputs must be specified. All the operations to be performed on the input data to obtain the output should be specified. This includes specifying the validity checks on the input and output data, parameters affected by the operations, and equations or other logical operations that must be used to transfer the inputs into corresponding outputs. An important part of the specification is the system behavior in abnormal situations, like input unit (which can occurs in many ways) or error during computations.

The functional requirement must clearly state what the system should do if such situations occurs. Specially, it should specify the behavior of the system for invalid input and invalid outputs. Furthermore, behavior for situations where the input is valid but the normal operations cannot be performed should also be specified. In short, the system for the foreseen inputs and all foreseen system states should be specified. These special conditions are often likely to be overlooked, resulting in the system that is not robust.

**Security Requirement:**

Security requirements are the particularly significant in defense system and many database systems. Security requirement place restrictions on the use of certain commands, control access to data, provide different kind of access requirement for different people, require the use of passwords and cryptography techniques, and maintain a log of activities in system. Given the current security needs even of common systems, they may also require proper assessment of security threats, proper programming techniques, and use of tools to detect flaws like buffer overflow.

For the purpose of security process I have added the login feature into my project so as to keep it safe from the external problem. One can only interact with my website by giving it the suitable i.e. the accurate ID. And this project is made for academic purpose so its security level is quite low.

**Output Requirement:**

In what formats do you need to deliver your information? Once upon a time, the only answer was print. Then for many organizations, the answer shifted to PDF. After that, it was PDF and HTML. Today, it’s PDF, HTML, and mobile-friendly content. So, output requirements include both media (print, online) and deliverable formats (PDF, HTML, mobile apps).

**Software Requirement:**

Software requirement plays a very important role in the making and development of a project, as it provides a suitable language as well as the perfect medium to implement our program or project on the system. Software requirement is very necessary for the implementation of a program.

The Software requirements specification is produced at the culmination of the analysis task. This is the way to represent requirements in a consistent format. It is a specification for a particular software product, program or a set of programs that performs certain functions in a specific environment .The function and allocation to software as part of system engineering are refined by establishing a complete information description, a detailed functional and behavioral description, an indication of performance requirements and design constraints, appropriate validation criteria, and other data pertinent to requirements.

Software product, program or set of programs that perform certain functions in a specific environment. There are two important cases regarding SRS: First one, SRS is used to define the needs and expectations of the users. The second one, SRS is written for different purpose and serve as a contract document between customer and developer. This produces the probability of the customer being disappointment with the final product.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Software Requirement:-** | |  |
|  |  | Operating System : | Window 10 Pro (Genuine) |
|  |  | OS type : | 64-bit OS |
|  |  | Web Languages : | HTML, CSS, BOOTSTRAP |
|  |  | Client side Scripting : | Java Script |
|  |  | Server Side Script : | PHP 5.3.13 (Hypertext Pre Processor) |
|  |  | Web Server : | Apache 2.2.22 |
|  |  | Database : | SQL Server |
|  |  | Control Panel : | XAMPP |
|  |  | Web Browser : | Google Chrome, Microsoft Edge, Firefox |
|  |  | Editors : | Sublime Text |

## Hardware Requirement:-

|  |  |  |
| --- | --- | --- |
|  | Processor : | Intel(R) Core(TM) i3-6006U CPU @2.00GHz |
|  | RAM : | 4.00 GB (3.88 GB Usable) |
|  | System Type : | x64 based Processor |
|  | Hard Disk Drive : | 1TB |
|  | Solid State Drive : | 256GB |

**MODULE DESCRIPTION:**

**Registration Module:**

In registration module there is a simple registration page where all the basic information about user will be submitted, in this module name of user, address, gender, users name, login ,Password are be mentioned. Login email will is unique which is frequently use by user and company admin for users detail. After registration the registration page is jump on log in page.

**Log In Module:**

In log in module a client will able to enter their email and password and they will jump to their profiles and can edit or update their profile, buy medicines and can submit their feedbacks and much more.

**Feedback Module:**

In this module one can gives feedback, complaints and his suggestions to admin and get back a response to their feedback.

**Search Module:**

The system contains one of the most useful and important module where one can search medicines and its information according to their need and prescription and dosage such as how many times can take medicines, medicines expiry and manufacturing date etc.

**Other Modules:**

There are more other modules in the project like contact, about and more user details module that one can view his information and other details related to the website. The whole project is designed in such a way that one can easily find a medicine according to their requirement and can fulfill his needs by using our system.

**SYSTEM DESIGN**

**SYSTEM DESIGN:**

System design provides the understandings and procedural details necessary for implementing the system recommended in the system emphasis is on the translating the performance requirements into design specifications. The design phase is a transition from a user oriented document (system proposal) to a document oriented to the programmers or database personnel.

**System design goes through two phases of the development:**

1. Logical Design
2. Physical Design

A data flow diagram shows the logical flow of the system. For a system it describes the input (source), output (destination), database (data stores), and procedures (data flows) all in a format that meets the user requirements. When analysis prepares the logical system design, they specify the user needs at a level of detail that virtually determines the information flow into and out of the system and the required data resources.

**SOFTWARE PROCESS MODEL:**

**Waterfall Model:**

The waterfall model is a popular version of the systems development life cycle model for software engineering. Often considered the classic approach to the systems development life cycle, the waterfall model describes a development method that is linear and sequential. Waterfall development has distinct goals for each phase of development. Imagine a waterfall on the cliff of a steep mountain. Once the water has flowed over the edge of the cliff and has begun its journey down the side of the mountain, it cannot turn back. It is the same with waterfall development. Once a phase of development is completed, the development proceeds to the next phase and there is no turning back.

The advantage of waterfall development is that it allows for departmentalization and managerial control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process. Theoretically, be delivered on time. Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order, without any overlapping or iterative steps.

The disadvantage of waterfall development is that it does not allow for much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage. Alternatives to the waterfall model include joint application development (JAD), rapid application development (RAD), synchronize and stabilize, build and fix problems. The waterfall model is a sequential design process, often used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of conception,initiation,analysis,design,construction,testing,production,implementation & maintenance.

**The whole project is based on waterfall model and follows the Top-down-designing Approach.**

Integration and module testing

Delivery

Maintenance

Coding and module testing

Design and specification

Feasibility study

Requirements analysis and specification

**Fig:-Waterfall Model**

**DESIGNING APPROACH:**

**Top down design:**

The TOP DOWN approach starts from the highest level component of the hierarchy and proceed through to lower levels. A top down design approach starts by the major component of the system. Decomposing them into their lower level component and iterative until the desired label of detail is achieved. Top down design method is in some form of step wise refinement. Starting from a abstract design in each step the design is refine to more concrete level, until we reach a level were no more refinement is needed.

A system consists of components, which have components of their own; indeed a system is a hierarchy of components. The highest level component corresponds to the total system. The top down approach start from the upper level component of hierarchy and proceeds through the lower level. By contrast a bottom up approach starts with the lowest level component of the hierarchy and proceeds through progressively higher levels to the top level components.

The top down approach has been promulgated by many researches and has been found to be extremely useful for design. Most design methodologies are based on the top down approach.

A top down approach is suitable only if the specifications of the systems are clearly known and the system development is from scratch. However, if a system is to be built from an existing system, a bottom approach is more suitable, as it starts from some existing components.

**HIGH LEVEL**

**DESIGN**

**HIGH LEVEL DESIGN:**

**DATA FLOW DIGRAM:**

A Data Flow Diagram (DFD) is a graphical representation of the information flow and transformation that are applied, as data moves from the input to the output. It is also known as “data flow graph” or a “bubble chart”. The DFD may be used to represent increasing information flow and functions details. A level 0 DFD, also called a fundamental system model, represent an entire software element as a single bubble with the input and the output data directed by the incoming and outgoing arrows. Data flow diagrams are commonly used during problem analysis. Data flow diagram is quite general and nit limited to problem analysis for software requirement specification. DFDs are very useful if understanding a system and can be effectively used during analysis.

A DFD shows the flow of data through the system It views the system as a function that transforms the input into desired output. The DFD aims to capture the transformations that take place within a system to the input data so that eventually the output data is produced. The agent that performs transformation of data from one state to another is called a process. so a DFD shows the movement of data through the different processes.

Named and circle shows the processes and data flows are shows by arrows entering or leaving the circles. A rectangle represent a source or sinks and is a net originator or consumer of the data. A source or sink is typically outside the main system of study.

The DFD should be carefully scrutinized to make sure that all the process in the physical environment are shown in DFD. It should also ensure that none of data flow is actually carrying control information.

**Features of DFD:**

* The exceptional simplicity of the DFD zymology is one reason why data oriented analysis techniques is the most widely used.
* The data flow diagram is a graphical tool that can be very valuable during the system analysis.
* The DFD depicts information flow without explicit notation of control.(e.g. conditions of loops).
* The level 0 data flow diagram should depict the software as a single bubble.
* Primary input/output files should be maintained.
* One bubble at a time should be refined.

There is a natural tendency to over complicate the DFD. This happens when we try to show too many details early.

**Basic Rules of DFD:**

**Here is the basic rules that apply to all DFDs:**

* No internal logic should be shown like loops, if-else, this is not a flow chart.
* In order to keep the diagram uncluttered, you can repeat data stores and external entities.
* No process can have only output data flows (a miracle).
* No process can have only input data flows (black hole).
* Data can’t be moved directly from one store to another without a process.
* Data can’t move directly from an external entity to a data store without a process.
* Data stores can’t be sink (only input data flows) or source (only output data flows) in level 1 DFD.

**DFD of Project**

**Level -0:**

**CONTEXT DIAGRAM:**

Admin

User

Request for login Request for registration

Response Response

**Level -1:**

Check for login Admin

Admin

Response Reply

Add/Edit category Insert Data Category

Response Reply

Add/Edit product Insert Data

Reply Product

Response

Manage Order View Order Order

Response Reply

View Report

Display Data admin/order

**Level -2:**

Admin

Check Detail Admin

Insert Data Product

Reply

Update Info Product

Reply

Remove Data Product

Reply

**Note:** The DFD should be carefully scrutinized to make sure that all the process in the physical environment are shown in DFD. It should also ensure that none of data flow is actually carrying control information.

**E-R Model Concept:**

The entity relationship model is a generalization of primitive commercial systems, which are based on hierarchical and network approaches. The E-R relationship, which is also known as Entity Relationship is based on the theory of real world which consists of a set of basic objects, which are called entities and relationships among these object. An entity exists as an object and is distinguishable from other objects.

**Entity:**

Any distinguishable person, place, thing, event or concept about which information is kept or an object which can be distinctly identified and distinguished and represented in a database or anything about which we store information is called an **Entity.**

**Attribute:**

Attributes describe the entity to which they are associated. A particular instance of an attribute is a value. In other words attributes are the characteristics of an entity type. Attributes can be classified as descriptors or identifiers. A descriptor describes a non-uniquely identify an instance of an entity.

**Relationship:**

It is an association among several entities, in other words it provides relationship between one or more entities.

**Notification For E-R Diagram:**

There is no standard for representing data objects in E-R diagram. Each modeling methodology uses its own notation. All notational style represents entities as rectangular boxes and relationship as lines connecting boxes. Each style uses a special set of symbols to represent the cardinality of a connection. An entity is represented in E-R diagram as a rectangular box enclosing the entity type name. Attribute names are enclosed in ellipses and attached to their entity type by straight lines.

**The basic symbols used for the E-R constructs are:**

|  |  |  |
| --- | --- | --- |
| **Notation** | **Use** | **Symbols** |
| **Lines** | Linking attributes to entity sets to relationship sets |  |
| **Ellipse** | Representing attributes |  |
| **Rectangle** | Representing entity set |  |
| **Diamond** | Represent relationship |  |
| **Nested**  **Ellipse** | Represents  multi valued  attribute |  |
| **Dotted**  **Ellipse** | Represents  derived  attributes | dotted_ellipse.png |
| **Nested**  **Rectangle** | Represents  weak  entities |  |
| **Line with**  **in**  **Ellipse** | Represents  key  attribute |  |
| **Double**  **Lines** | Represents  total  relationship |  |
| **Ellipse and**  **Lines** | Represents  composite  attributes |  |
| **Nested**  **diamond** | Represents  identifying  relationship |  |
| **Triangle** | Represent  is a **/** has a  relationship |  |

**Keys Concept:**

A key is a value which can always be used to uniquely identify an object instance. It becomes important to invent a method to distinguish entity and relationships. The differences between entities must be expressed in terms of attributes.

**Super Key:**

A Super Key is a set of one or more attributes which, taken collectively, allows us to identify uniquely an entity in the entity set.

**Candidate Key:**

Candidate Key is a minimal super key that uniquely identifies a record in a table. Candidate key is also referred to as Surrogate keys.

**Primary Key:**

In a database table an attribute which can be used to uniquely identify the records is called Primary Key. In other words a Primary Key is a key which is a part of candidate key.

**Alternate Key:**

Alternate key is a key which is the part of candidate key but not primary key. In other words if there are multiple candidate keys in a table, then the keys which are not chosen as primary key will be called as alternate key.

**Composite Keys:**

When the key that uniquely identifies the rows of the table is made up of more than one attribute it is called a composite key. In other words if we use multiple attributes to create a primary key, then that primary key is refers to as a Composite Key.

**Foreign Keys:**

A foreign key is column or group of column in a relational database table that provides the link between data in two tables. It acts as cross reference between tables because it references the primary key to other table thereby establishing a link between them. In other words foreign key is a key that uniquely identify records from one table to other table.

**Guideline for Drawing E-R Diagram:**

When gathering information i have to:

* 1. Identify the entities in the system.
  2. Identify the attribute of each entity.
  3. Identify the relationship between the entities and more things like this.

**E- R Diagram of the Project:**

**E-R Diagram of Online Medical Store**

Buys

Has many

Category

Medicines

Cart

Contain

User

Has many

Role

**Table Schema:**

Table schema defines the structure of back end tables and its short details. The main job of this table schema is to keep a short description of data types, its values and use of this with their unique Primary Key.

**Table Name: - Log In**

**Table Description: -**

This table stores the **email** and **password** of admins and defines id as primary key and also used to validate the users.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Constraints** | **Data Type** | **Size** |
| ID | Primary Key | Int | 10 |
| User Name | Not Null | Varchar | 20 |
| Password | Not Null | Varchar | 20 |

**Table Name: - Registration**

**Table Description: -**

The tables store the registration of users and buys and see medicines with their details. Here we define **a id** as primary key that we can easily access the full address easily.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Constraints** | **Data Type** | **Size** |
| ID | Primary Key | Int | 10 |
| Fname | Not Null | Varchar | 10 |
| Lname | Not Null | Varchar | 10 |
| E-mail | Not Null | Varchar | 40 |
| Password | Not Null | Varchar | 20 |
| Mobile No. | Not Null | Numeric | 10 |

**Table Name: - Feedback**

**Table Description: -**

This table stores the information of feedback given by clients and job providers and can be identified by the unique **id** here we have also stored the email of the person who submitted the feedback that we can easily identify the person.

Feedback can be given by both Job seeker and Job Providers and will get response back from Admin.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Constraints** | **Data Type** | **Size** |
| Id | Primary Key | Integer | 10 |
| Type | Not Null | Varchar | 10 |
| Name | Not Null | Varchar | 20 |
| Email | Not Null | Varchar | 40 |
| contact no | Not Null | Numeric | 10 |
| Date | Not Null | Numeric | 10 |
| Description | Not Null | Varchar | 80 |

**Table Name: - Medicine info**

**Table Description: -**

This table stores information about various medicines and details about them such as name, mfg\_date, exp\_date etc. Here we are considering **id** as primary key to uniquely identify the medicines and their names etc.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Constraints** | **Data Type** | **Size** |
| ID | Primary Key | Int | 10 |
| Name | Not Null | Varchar | 20 |
| Cost | Not Null | Numeric | 10 |
| Mfg date | Not Null | Numeric | 10 |
| Expiry date | Not Null | Numeric | 10 |
| Category | Not Null | Varchar | 10 |

**Table Name: - Order info**

**Table Description: -**

This table stores information about various orders and details about them such as date, amount, quantity etc. Here we are considering **id** as primary key to uniquely identify the orders and their names etc.

Here we stored the **id** of each order that we will access from medicine info table corresponding to their details.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Constraints** | **Data Type** | **Size** |
| Order id | Primary Key | Int | 10 |
| Date | Not Null | Numeric | 10 |
| Amount | Not Null | Numeric | 10 |
| Quantity | Not Null | Numeric | 10 |

## Low level design:

**Modulation:**

A system is considered modular if it consists of discrete component show that each component can be implemented separately, and a change to one component has minimal impact on other components.

Each function in each abstraction has a single, well defined purpose.

* Each function manipulates no more than one major data structure.
* Function share global data selectively. It easy to identify all routines that share a major data structure.
* Function that manipulates instances of abstract data types are encapsulated with the data structure be in manipulated.

**Structure chart:**

The structure chart is one the most commonly used methods for system design. Structures charts are used during architectural design to document hierarchical structure, parameters and interconnection in a system.

**BIBLIOGRAPHY & REFERENCES**

**BOOKS:-**

1. Jennifer Niederst Robbins “Learning Web Design” 5th  Edition June 2012,
2. Pankaj Jalote “An approach to Software Engineering”, 6th edition, 2015, Narosa Publishing House.
3. Leon & Leon, “Database Management System”, Vikas Publishing House.
4. Tim Suhering “Teach Yourself PHP” SAMS Publication
5. Jason Smith “Build-and-Design-a-Website-HTML-CSS”, etc.

**WEBSITES:**

1. <http://www.cs.bham.ac.uk/~rxb/HTML_text/hci/Schedule/Requirements.html>
2. <https://en.wikipedia.org/wiki/Systems_analysis>
3. <https://www.tutorialspoint.com/software_testing_dictionary/software_requirement_specification.htm>
4. <https://www.linkedin.com/pulse/what-system-design-why-important-development-process-fareed>
5. <http://www.the-software-experts.com/e_dta-sw-process.php>
6. [https://www.smartdraw.com/entity-relationship-diagram](https://www.smartdraw.com/entity-relationship-diagram/)
7. <https://www.smartdraw.com/data-flow-diagram/>
8. <https://www.visual-paradigm.com/tutorials/data-flow-diagram-dfd.jsp>
9. <https://www.tutorialspoint.com/sdlc/sdlc_overview.htm>