

CONTENTS

<u>S.no</u>	<u>Content Name</u>	<u>Page No</u>
1.	Introduction	4
1.1	A brief overview of the HCMS	
1.1.1	Advantages to the End Users	
1.2	Organization Profile	
1.3	Goal and Need	
1.4	General Methodology in Developing Software Project	
1.4.1	Requirement Analysis Phase	
1.4.2	Design Phase	
1.4.3	Development Phase	
1.4.4	Coding Phase	
1.4.5	Testing Phase	
2.	Requirement Analysis	12
2.1	Introduction	
2.2	Data Collection	
2.2.1	Observation	
2.3	Software Requirement Specification Document	

2.3.1 Introduction	
2.3.2 Problem Definition	
2.3.3 Hardware requirements	
2.3.4 Software requirements	
2.3.5 Design Constraints	
3. System Analysis	18
3.1 Module description	
3.2 Feasibility analysis	
3.3 Studying the existing system	
3.4 Proposed system	
4. Design Phase	27
4.1 Introduction.	
4.2 Flow charts	
4.3 Data flow diagrams	
5. Development Phase	34
5.1 Features of Java language	
5.2 Ms-Access	
6. Testing Phase	57
7. Screens	67
8. Conclusion	68
9. Bibliography	70

ABSTRACT

The main intention of introducing this system is to reduce the manual work at Health center counters. Every sort of task is performed by the system, such as registering different types of persons (i.e employees , students and others) ,enquiries, and complaints etc. reducing much paper work and burden of file storage. Also the latest information is right available for the officials and executives wherever they require. The system also facilitates the pharmacist to enquire about the drugs and about the stock to be ordered and about the expiry date .

Where the system must be placed?

There are a lot of benefits to the Health center by placing the system at their registration and at drug store office . At the same time the patients are also benefited using this system. They can get the work done within no time.

How to use the system?

Using the system is as simple as using the personal computer. Since end user computing is developing in our country, It is beneficial to both Health center and the patients. Every step is clearly defined and help is provided through out the application to the user. Even the exceptions are handled well to avoid confusion.

How is it beneficial to the Health Center?

The health center can get much out of the system. The system is used to enter the patient details and to enter the details about the health center and the details about the in-patient and out-patient in detail and about the reports of the patients . This system represents the patient by the **OP** number and this is main criteria how the patient is provided by the free services . The drug information and the specifications is also provided in this Health Center Management System.



1. INTRODUCTION

1.1. BRIEF OVERVIEW OF HEALTH CENTER MANAGEMENT SYSTEM

To develop a Health Center 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

1.1.1 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

1.2

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

Health Center Management system

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:

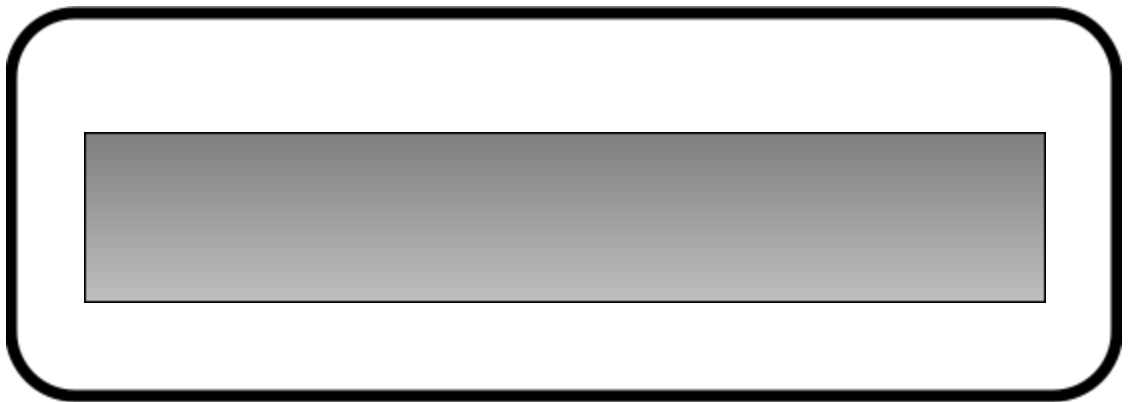
The development phase includes choosing of a suitable s/w to solve the particular problem given. The various facilities and the sophistication in the selected s/w give a better development of the problem.

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.



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 through 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 be 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 Center Management System project has been divided into four modules. They are

1. Registration
2. Drug Store
3. Case Records
4. Daily Entries

Module-One Registration

This module consists of the following sub modules viz.,

Inserting the Records of the persons sent from the different departments

Viewing the Records

Module-Two Drug Store

This module is divided into three sub modules. They are

Inserting Drugs

Updating Drugs

About Drugs

Module-Three Case Records

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

In-patient Record (Student)

In-patient Record (Employee)

Out-patient Record (Student & Employees)

Module-Four Daily Entries

This module again is divided into two sub modules. They are

Pharmacy daily Entries

Nurse daily Entries

Software Requirements:

Platform	-	Windows (2000/XP) /Unix/Solaris
Software	-	JSDK 5.0, Ms-Access.

Hardware Requirements:

Processor	-	Intel celeron class Processor with 2.0 GHz
RAM	-	256 MB
Hard Disk	-	40 GB
Keyboard	-	101 keys
Mouse	-	Any pointing device

Design Constraints:

This Health Center 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.



SYSTEM ANALYSIS

Analysis is the detailed study of the various operations performed by a system and their relationships within and outside of the system. A key question is: What must be done to solve the problem? One aspect of analysis is defining the boundaries of the system and determining whether or not candidate system should consider other related systems. During analysis, data are collected on the available files, decision points, and transactions handled by the present system.

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.

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

- 1 Registration
- 2 Drug stores
- 3 Case Records
- 4 Daily Entries

1. Registration

This module has been divided into two sub modules. They are
Inserting the new records

Retrieving the record

1.1. Inserting the new records

The information of the students and employees are send from the different departments to provide the free service. All the records which are send from the departments are entered into the database and each member is represented through the unique number known as the opno. Inserting of the new records include the opno i.e which is allotted uniquely for providing the free services, Name, Age, Sex, Address , Family members of the employee and other Beneficiaries under the S.V. Health Center.

1.2 Retrieving the details

The records which are entered into the database are retrieved to check whether that person who has come to take the service at the health center is eligible to take the service are not. This is done by entering the opno that has allotted to the patient, if the data is retrieved from the database the person is eligible to take the service and by this ,if person is the employee then in some cases he/she is charged like wise the x-ray charges and the bed charges for the employee. The students are not charged in any cases all the services are provided for the free of the cost.

2 Drug Stores

This module is divided into three modules. They are

2.1 Drug Entry

2.2 Updating Drug Entry

2.3 Stock Details

2.1 Drug Entry

This is an entry done by entry operator after getting the stock from the medical transcribers and the information is stored in the database. The information is in the form starting with the Drug name, in-stock, Expiry date1 and Expiry date2. The Drug name is uniquely identified such that the database is maintained without any complexity. Here the expiry date is represented in the form of month-year, but not in the form mm / dd / yy. The stock is first stored in the main drug stores and every week it is shifted to the pharmacy store by watching the stock at the pharmacy house.

2.2 Updating Drug Entry

This is an entry done by same entry operator after the drugs all entered in the database. The drugs should be shifted to the pharmacy house by checking the stock at the pharmacy house. First the Main drug store maintains the stock and the drugs are shifted to the pharmacy house periodically. This periodical shifting of the drugs are updated such that the drug house knows how much of the stock is present in the drug store, Here the drug stores mainly considers the expiry date as the main criteria the drugs which have the nearer expiry date are shifted to the pharmacy house.

2.3 Stock Details

The stock details are so important such that we know how many drugs are in the main store, what are the expiry dates of the drugs. When the stock is taken it will minimum of two and maximum of five expiry dates. So it is better to watch the expiry dates so that the drugs are shifted to the pharmacy house according to the drugs which are having the early expiry date.

3 Case Records

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

- 3.1 Students In-Patients Records
- 3.2 Employees & Beneficiaries In-Patients Records.
- 3.3 Out – Patients Records (Students & Beneficiaries)

3.1 Students In-Patients Records

The information about the in-patients should be notified such that the treatment for the patient is done in the proper way. The students who are in-patients are maintained separately because everything to the student is served freely i.e there is no bed charges, no charge for the scanning, no charge for the X-Rays and every thing is supplied freely. There may be special cases that are used for case studies how that disease has occurred and what are the precautions are to be taken to overcome that disease. The diseases that affect the other patients are shifted to the Isolation ward such that disease is not affected to the other patients.

3.2 Employees & Beneficiaries In-patient Records

The employees and other beneficiaries are maintained separately such that there are applicable for some charges .These

employees have the bed charges, X-Ray and Scanning charges. These are calculated according to the days present in the hospital. The medicines are freely served to the patient only in some special cases the patient is charged.

3.3 Out-Patients Record

The patients are served for the common problems which are not serious, the treatment to all the beneficiaries and students are provided freely. There are separate wards for the ladies and gents and in some special cases patients are sent to the chief doctor.

4 Daily Records

Daily records are then divided into two sub modules. They are

4.1 Pharmacy Daily Entries

Pharmacy daily entries are the entries which are given to the out-patients and these are noted at the end of the day. Daily entries will give the information about how many drugs are needed every day. This will then take the sufficient number of drugs from the main stores to the Pharmacy house.

4.2 Nurse Daily Entries

Nurse daily entries that are entered with the operator when the drug has been administered to the patient, The date, time what type of the drugs applied to the patient are to be noted in certain period of time.

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:

- ✓ Economic Feasibility
- ✓ Operational Feasibility
- ✓ Technical Feasibility

1. **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.

2. **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.

3. **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.

Disadvantages

1. The work is done manually so that it takes much time to recognize the patient at the registration office.
2. More number of labors is needed.
3. We can't asses the calculations accurately.
4. Amount of time is more needed when the more patients come to registration counter. It counts time factor.
5. As there are thousands of patients records; Searching process is a difficult task.

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

Health Center Management system

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

Advantages

1. A fast and more efficient service to all patients . As there are thousands of patients records; Searching process is an easy task.
2. Saving in staff time in entering and manipulating data.
3. Easy input, deletion and manipulation of lot, patients details.
4. Simple correction of input errors and we can asses the calculations accurately.

Disadvantage

1. Loss of data when electronic fluctuations occur.

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.



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 FLOW CHARTS

Before solving a problem with the help of a computer, it is essential to plan the solution in a step-by-step manner. Such a planning is represented symbolically with the help of flow chart. It is an important tool of system analysts and

Programmers for tracing the information flow and the logical sequence in data processing Logic is the essence of a flow chart.

A flow chart is the symbolic representation of step-by-step solution of a given problem, and it indicates flow of entire process, the sequence of the data input, operations, computations, decisions, results and other relevant information.

Pertaining to a particular problem, a flow chart helps us in the complete understanding of the logical structure of a complicated problem and in documenting the method used. It would be seen that the flow chart is a very convenient method of organizing the logical steps and deciding what, when and how to proceed with various processes. The logic should be depicted in the flow charts. Computerization of the data without a flow chart is like constructing the building without a proper design and detailed drawings.

Kinds of the Flow Charts

1. System Flow Chart

The system analyst to describe data flow and operations for the data processing cycle uses these. A system flow chart defines the broad processing in the organizations, showing the origin of the data, filing structure, processing to be performed, output that is to generate and necessity of the offline operation.

2. Program Flow Chart (or) Computer Procedure flow chart

The programmers to describe the sequence of operations and the decision of a particular problem normally use these. A program flow chart plans the Program structure and also serves the purpose of documentation for a program, which is to be retained and used at a later date either by the original programmer or others.

Advantages:

Apart from, the DFDS the flow charts has been helping the programmer to develop the programming logic and to serve as the documentation for a Completed program, it has the following advantages

1. They help for the easy understanding of the logic of a Process or a procedure
2. It is a better communicating tool than writing in words.
3. It is easy to find the conditions, which are responsible For the actions.
4. It is an important tool for planning and designing the New system.
5. It clearly indicates the role-played at each level.
6. It provides an overview of the system and also demonstrates the relationship between the various steps.
7. It facilitates troubleshooting.
8. It promotes logical accuracy.

Disadvantages:

1. Communication lines are not always easy to show.
2. The charts are sometimes complicated.
3. Reproduction is difficult.

- 4. They are hard to modify.

UML DIAGRAMS

TABLES

Drugs Table:-

DrugName	Istock	Ostock	ExDate1	ExDate2	Total

In-Patient Record :-

Opno	Name	Age	Sex	DoB	DoA	DoD

Out-Patient Record:-

Opno	Name	Age	Sex	Diagnosis

Registered :-

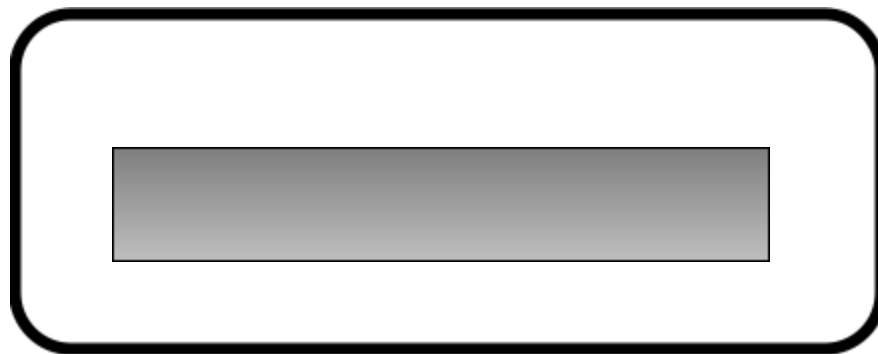
Opno	Name	Age	Sex	Occupation	Diagnosis

Registration :-

Opno	Name	Age	Sex	Occupation	Address	Family members

Pharmacy Entries:-

Opno	Drug Name1	No. of Tablets issued DN1	Drug Name2	No. of Tablets issued DN2



DEVELOPMENT PHASE

FEATURES OF LANGUAGE

JAVA

When the chronicle of computer languages is written, the following will be said: B led to C, C evolved into C++, and C++ set the stage for Java. To understand the reasons that drove the creation, the forces that shaped it, and the legacy that it inherits. Like the successful compute languages that came before, Java is a blend of the best elements of its rich heritage combined with the innovative concepts required by its unique environment.

JAVA'S FEATURE SET:

Although the fundamental forces that necessitated the invention of Java are portability and security, other factors also played an important role in molding the final form of the language and are given by

- ☐ Simple
- ☐ Object Oriented
- ☐ Portable
- ☐ Robust
- ☐ Security
- ☐ Multithreaded
- ☐ Interpreted and High performance
- ☐ Distributed

- ☐ Dynamic
- ☐ Garbage Collection
- ☐ No Pointers
- ☐ Simplicity

Simple

Java was designed to be easy for the professional programmer to learn and use effectively. Java will be even easy if we already through in the concept of object oriented. Some of confusing concepts are left out of Java or implemented in a cleaner, more approachable manner.

Object Oriented

The Java programming language is object oriented, which makes program design focus on what you are dealing with rather than on how you are going to do something. This makes it more useful for programming in sophisticated projects because one can break the things down into understandable components. A big benefit is that these components can then be reused.

Object oriented languages use the paradigm of classes. In simple term, a class includes both the data and functions to operate on the data. You can create an instance of a class, also called an object, which will have all the data members and functionality of its class. Because of this, you can think of a class as being like template, with each object being a specific instance of a particular type of class.

The class paradigm allows one to encapsulate data so that those using the data cannot see specific data values and function implementation. Encapsulation makes it possible to make the changes in code without breaking other programs that use that code. If for example the implementation of a function is changed, the change is invisible to another programmer who invokes that function, and it does not affect his/her program, except hopefully to improve it.

Java includes inheritance, or the ability to derive new classes from existing classes. The derived class, also called a subclass, inherits all the data and functions of the existing class, referred to as parent class. A subclass can add new data members to those inherited from the parent class. As far as methods are concerned, the subclass can reuse the inherited methods, as it is, change them, and/or add its own new methods.

Portable

One of the biggest advantages Java offers is that it is portable. An application written in Java will run on all the major platforms. Any computer with a Java based browser can run the applications or applets written in the Java programming language. A programmer no longer has to write one program to run on a Macintosh, another program to run on a Windows machine, still another to run on UNIX machine, and so on. In other words, with Java, developers write their programs only once. The virtual machine is what gives Java its cross platform capabilities. Rather than being compiled into machine language, which is different for each operating system and computer architecture, Java code is compiled into byte codes.

With other languages, the program code is compiled into a language the computer can understand. The problem is that other computers with different

machine instruction set cannot understand that language. Java code on the other hand is compiled into byte code rather than a machine language. These byte codes go to the Java virtual machine, which executes them directly or translates them into the language that is understood by the machine running it.

In summary, these means that with the JDBC API extending Java, a programmer writing Java code can access all the major relational databases on any platform that supports the Java virtual machine.

Robust

The multi platform environment of the web places extraordinary demands on a program, because the program must execute reliably in a variety of systems. Thus, the ability to create robust programs was given a high priority in the design of Java. However, it also checks our code at run time. In fact, many space hard-to-down bugs that often turn up in hard-to reproduce run time situations are simply impossible to create in Java.

To better understand how java is robust, two of the reasons are mainly considered for program failure memory management mistakes and mishandling exceptional conditions. Memory management can be a difficult, tedious task in traditional programming environments. Java virtually eliminates these problems by managing memory allocation and deal location.

Exceptional conditions in traditional environment often arise in situations such as division by zero of “file not found” and they must be managed with clumsy and hard-to-read constructs. Java helps in this area by providing object-oriented exception handling.

Security

Every time that we downloaded a “normal” program, we are risking a viral infection. Prior to Java most users did not download executable programs frequently, and those who did scan them for viruses prior to execution. Even so, most users still worried about the possibilities of infecting their systems with a virus.

In addition to viruses, another type of malicious program exists that must be guarded against. This type of program can gather private information, such as credit card numbers, bank account balances and passwords, by searching the contents of our computer’s local file system. Java answers both of this concern by providing a “firewall” between a networked application and our computer.

Multithreaded

Java was designed to meet the real-world requirement of creating interactive, networked programs. To accomplish this, java supports multithreaded programming, which allows us to write programs that do many things simultaneously. The Java run-time system comes with an elegant yet sophisticated suplicated solution for multi process synchronization that enables us to construct smoothly running interactive systems.

Interpreted and high performance

Java enables the creation of cross-platform programs by compiling into an intermediate representation called Java byte code. This code can be interpreted on any system that provides a Java virtual machine. Most previous attempts at cross-platform solutions have done so at the expense of performance. Other interpreted system, such as BASIC, Tcl and PEAL, suffer from almost insurmountable performance deficits. Java, however, was designed to perform well on very low power CPUs.

Distributed

Java is built with network communications in mind. It has a comprehensive library of routines for dealing with network protocols such as TCP/IP, HTTP and FTP. As a result, Java application can open and access objects across the Internets with the same ease that programmers normally expects while accessing a local file system. You don't have to worry about implementing the details of network yourself. Java comes with everything needed for truly distributed computing.

Dynamic

Fundamentally, distributed computing environment must be dynamic. Java was designed to adapt in a constantly evolving environment. It is capable of incorporating new functionality regardless of where that functionality comes from. The local computer system, the local and wide area networks, and the internet are all potential contributors.

Garbage Collection

Automatically takes care of allocating and de-allocating memory, a huge potential source of errors. If an object is no longer being used (has no references to it), then it is automatically removed from memory, or "Garbage Collected". Programmers don't have to keep track of what has been allocated and de-allocated them, which makes their job a lot easier, but more importantly it stops memory leaks.

No Pointers

Eliminates big source errors. By using object references instead of memory pointers, problems with pointer arithmetic are eliminated, and problems with inadvertently accessing the wrong memory address are greatly reduced.

Simplicity

Makes Java easier to learn and use correctly. Java keeps it simple by having just one way to do something instead of having several alternatives, as in some languages. Java also stays lean by not including multiple inheritances, which eliminates the errors and ambiguity that arise when you create a subclass that inherits from two or more classes.

To replace capabilities, multiple inheritances provide, Java lets you add functionality to a class through the use of interfaces.

5.2 Introduction to JDBC:

JDBC™ is a Java™ API for executing SQL statements. (As a point of interest, JDBC is a trademarked name and is not an acronym; nevertheless, JDBC is often thought of as standing for "Java Database Connectivity".) It consists of a set of classes and interfaces written in the Java programming language. JDBC provides a standard API for tool/database developers and makes it possible to write database applications using a pure Java API.

Using JDBC, it is easy to send SQL statements to virtually any relational database. In other words, with the JDBC API, it isn't necessary to write one program to access a Sybase database, another program to access an Oracle database, another program to access an Informix database, and so on. One can write a single program using the JDBC API, and the program will be able to send SQL statements to the appropriate database. And, with an application written in the Java programming language, one also doesn't have to worry about writing different applications to run on different platforms. The combination of Java and JDBC lets a programmer write it once and run it anywhere.

Java being robust, secure, easy to use, easy to understand, and automatically downloadable on a network, is an excellent language basis for database applications. What is needed is a way for Java applications to talk to a variety of different databases. JDBC is the mechanism for doing this.

JDBC extends what can be done in Java. For example, with Java and the JDBC API, it is possible to publish a web page containing an applet that uses

information obtained from a remote database. Or an enterprise can use JDBC to connect all its employees (even if they are using a conglomeration of Windows, Macintosh, and UNIX machines) to one or more internal databases via an intranet. With more and more programmers using the Java programming language, the need for easy database access from Java is continuing to grow.

MIS managers like the combination of Java and JDBC because it makes disseminating information easy and economical. Businesses can continue to use their installed databases and access information easily even if it is stored on different database management systems. Development time for new applications is short. Installation and version control are greatly simplified. A programmer can write an application or an update once, put it on the server, and everybody has access to the latest version. And for businesses selling information services, Java and JDBC offer a better way of getting out information updates to external customers.

5.2.1 JDBC does the following things:

- ☐ Establish a connection with a database
- ☐ Send SQL statements
- ☐ Process the results.

The following code fragment gives a basic example of these three steps:

```
Class.forName ("oracle.jdbc.driver.OracleDriver");  
Connectioncon=DriverManager.getConnection("jdbc:oracle:thin:@192.168.0.  
2:1521:tit");  
Statement stmt=con.createStatement ();
```

5.2.2 CONNECTION

A connection object represents a connection with a database. A connection session includes the SQL statements that are executed and the results that are returned over the connection. A single application can have

one or more connections with a single database, or it can have connections with many different databases.

5.2.2.1 OPENING A CONNECTION

The standard way to establish a connection with a database is to call the method `DriverManager.getConnection`. This method takes a string containing a URL. The Driver Manager class, referred to as the JDBC management layer, attempts to locate a driver that can connect to the database represented by the driver classes, and when the method `getConnection` is called, it checks with each driver in the list until it finds one that can connect using this URL to actually establish the connection.

The standard syntax for JDBC URLs is:

Jdbc: < sub protocol>: < sub name>

A JDBC URL has three parts, which are separated by colons:

- jdbc-the protocol. The protocol in a JDBC URL is always jdbc.
- <Sub protocol>-usually the driver or the database connectivity mechanism, which may be supported by one or more drivers. A prominent example of a sub protocol name is “oracle”, which has been reserved for URLs that specify “thin”-style data source names.
- <Sub name>- a way to identify the database. The sub names can vary, depending on the sub protocol, and it can have a sub name with any internal syntax the driver writer chooses. The point of a sub name is to give enough information to locate the database.

5.2.2.2 SENDING STATEMENT

Once a connection is established, it is used to pass SQL statements to its underlying database. JDBC does not put any restrictions on the kinds of SQL statements that can be sent; this provides a great deal of flexibility, allowing the use of database-specific statements or even non-SQL statements. It requires, however, that the user be responsible for making sure that the underlying database can process the SQL statements being sent and suffer the consequences if it cannot.

5.2.3 DRIVER MANAGER

The Driver Manager class is the management layer of JDBC, working between the user and the drivers. It keeps track of the drivers that are available and handles establishing a connection between a database and the appropriate driver. In addition, the driver manager class attends to things like driver login time limits and the printing of log and tracing messages. The only method in this class that a general programmer needs to use directly is `DriverManager.getConnection`. As its name implies, this method establishes a connection to a database.

5.3 JDBC IS A LOW-LEVEL API & A BASE FOR HIGHER-LEVEL APIS

JDBC is a "low-level" interface, which means that it is used to invoke (or "call") SQL commands directly. It works very well in this capacity and is easier to use than other database connectivity APIs, but it was designed also to be a base upon which to build higher-level interfaces and tools. A higher-level interface is "user-friendly," using a more understandable or more convenient API that is translated behind the scenes into a low-level interface such as JDBC. At the time of this writing, two kinds of higher-level APIs are under development on top of JDBC:

1. An embedded SQL for Java. At least one vendor plans to build this. DBMSs implement SQL, a language designed specifically for use with databases. JDBC requires that the SQL statements be passed as Strings to Java methods. An embedded SQL preprocessor allows a programmer to instead mix SQL statements directly with Java: for example, a Java variable can be used in a SQL statement to receive or provide SQL values..
2. A direct mapping of relational database tables to Java classes. Java Soft and others have announced plans to implement this. In this "object/relational" mapping, each row of the table becomes an instance of that class, and each column value corresponds to an attribute of that instance. As interest in JDBC has grown, more developers have been working on JDBC-based tools to make building programs easier, as well. Programmers have also been writing applications that make accessing a database easier for the end user. For example, an application might present a menu of database tasks from which to choose. With the requested input typed in, the application then automatically invokes the necessary SQL commands. With the help of such an application, users can perform database tasks even when they have little or no knowledge of SQL syntax.

5.3.1 JDBC VERSUS ODBC AND OTHER APIS

At this point, Microsoft's ODBC (Open Database Connectivity) API is probably the most widely used programming interface for accessing relational databases. It offers the ability to connect to almost all databases on almost all platforms. So why not just use ODBC from Java?

The answer is that you can use ODBC from Java, but this is best done with the help of JDBC in the form of the JDBC-ODBC Bridge, which we will cover shortly. The question now becomes, "Why do you need JDBC?" There are several answers to this question:

1. ODBC is not appropriate for direct use from Java because it uses a C interface. Calls from Java to native C code have a number of drawbacks in the security, implementation, robustness, and automatic portability of applications.
2. A literal translation of the ODBC C API into a Java API would not be desirable. For example, Java has no pointers, and ODBC makes copious use of them, including the notoriously error-prone generic pointer "void *". You can think of JDBC as ODBC translated into an object-oriented interface that is natural for Java programmers.
3. ODBC is hard to learn. It mixes simple and advanced features together, and it has complex options even for simple queries. JDBC, on the other hand, was designed to keep simple things simple while allowing more advanced capabilities where required.
4. A Java API like JDBC is needed in order to enable a "pure Java" solution. When ODBC is used, the ODBC driver manager and drivers must be manually installed on every client machine. When the JDBC driver is written completely in Java, however, JDBC code is automatically installable, portable, and secure on all Java platforms from network computers to mainframes.

In summary, the JDBC API is a natural Java interface to the basic SQL abstractions and concepts. It builds on ODBC rather than starting from scratch, so programmers familiar with ODBC will find it very easy to learn JDBC. JDBC retains the basic design features of ODBC; in fact, both interfaces are based on the X/Open SQL CLI (Call Level Interface). The big difference is that JDBC builds on and reinforces the style and virtues of Java, and, of course, it is easy to use. More recently, Microsoft has introduced new APIs beyond ODBC: RDO, ADO, and OLE DB. These designs move in the same direction as JDBC in many ways, that is, in being an object-oriented database interface based on classes that can be implemented on ODBC. However, we did not see compelling functionality in any of these interfaces to make them an alternative basis to ODBC, especially

with the ODBC driver market well established. Mostly they represent a thin veneer on ODBC. This is not to say that JDBC does not need to evolve from the initial release; however, we feel that most new functionality belongs in higher-level APIs such as the object/relational mappings and embedded SQL mentioned in the previous section.

5.3.2 TWO-TIER AND THREE-TIER MODELS

The JDBC API supports both two-tier and three-tier models for database access. In the two-tier model, a Java applet or application talks directly to the database. This requires a JDBC driver that can communicate with the particular database management system being accessed. A user's SQL statements are delivered to the database, and the results of those statements are sent back to the user. The database may be located on another machine to which the user is connected via a network. This is referred to as a client/server configuration, with the user's machine as the client, and the machine housing the database as the server. The network can be an intranet, which, for example, connects employees within a corporation, or it can be the Internet.

In the three-tier model, commands are sent to a "middle tier" of services, which then send SQL statements to the database. The database processes the SQL statements and sends the results back to the middle tier, which then sends them to the user. MIS directors find the three-tier model very attractive because the middle tier makes it possible to maintain control over access and the kinds of updates that can be made to corporate data. Another advantage is that when there is a middle tier, the user can employ an easy-to-use higher-level API which is translated by the middle tier into the appropriate low-level calls. Finally, in many cases the three-tier architecture can provide performance advantages.

Until now the middle tier has typically been written in languages such as C or C++, which offer fast performance. However, with the introduction of optimizing compilers that translate Java byte code into efficient machine-specific code, it is becoming practical to implement the middle tier in Java.

5.3.3 JDBC DRIVER TYPES

The JDBC drivers that we are aware of at this time fit into one of four categories:

1. JDBC-ODBC Bridge plus ODBC driver: The Java Soft bridge product provides JDBC access via ODBC drivers. Note that ODBC binary code, and in many cases database client code, must be loaded on each client machine that uses this driver. As a result, this kind of driver is most appropriate on a corporate network where client installations are not a major problem, or for application server code written in Java in three-tier architecture.
2. Native-API partly-Java driver: This kind of driver converts JDBC calls into calls on the client API for Oracle, Sybase, Informix, DB2, or other DBMS. Note that, like the bridge driver, this style of driver requires that some binary code be loaded on each client machine.
3. JDBC-Net pure Java driver: This driver translates JDBC calls into a DBMS- independent net protocol, which is then translated to a DBMS protocol by a server. This net server middleware is able to connect its pure Java clients to many different databases. The specific protocol used depends on the vendor. In general, this is the most flexible JDBC alternative. In order for these products to also support Internet access, they must handle the additional requirements for security, access through firewalls, and so on, that the Web imposes.
4. Native-protocol pure Java driver: This kind of driver converts JDBC calls into the network protocol used by DBMSs directly. This allows a direct call from the client machine to the DBMS server and is a practical solution for Intranet access.

Eventually, we expect that driver categories 3 and 4 will be the preferred way to access databases from JDBC. Driver categories 1 and 2 are interim solutions

where direct pure Java drivers are not yet available. There are possible variations on categories 1 and 2 (not shown in the table below) that require a connector, but these are generally less desirable solutions. Categories 3 and 4 offer all the advantages of Java, including automatic installation (for example, downloading the JDBC driver with an applet that uses it).

The following chart shows the four categories and their properties:

DRIVER CATEGORY	ALL JAVA?	NET PROTOCOL
1 – JDBC-OCBC Bridge	No	Direct
2 – Native API as basis	No	Direct
3 – JDBC-Net	Yes	Requires Connector
4 – Native protocol as basis	Yes	Direct

5.3.4 OBTAINING JDBC DRIVERS

At the time of this writing, there are dozens of drivers in Category 1: ODBC drivers that can be used with JavaSoft's bridge. There are currently about a dozen Category 2 drivers built on top of native APIs for DBMSs. There are a few Category 3 drivers. Currently there are at least two Category 4 drivers, but by the end of 1997, we expect that there will be Category 4 drivers for all of the major DBMSs.

To get the latest information on drivers, check the JDBC web page at [http:// java.sun.com/products/jdbc](http://java.sun.com/products/jdbc). The first vendors with Category 3 drivers available were SCO, Open Horizon, Visigenic, and Web Logic. Java Soft and Intersolv, a leading database connectivity vendor, worked together to produce the JDBC-ODBC Bridge and the JDBC Driver Test Suite.

5.4 Introduction to MS Access:

5.4.1 About Data Access Pages:

A data access page is a web page that you can use to add, edit, view or manipulate current data in a Microsoft Access database or an SQL server database .You can create pages that are used to enter and edit data, similar to Access forms. You can also create page that display records grouped hierarchically, similar to Access reports.

Collect and distribute data in several ways. You can use pages to add, edit and view data within a Microsoft Access database or Microsoft Access projects. You can use them on them on the internet or on an intranet, and you can send them in e-mail.

5.4.2 View grouped records interactively:

On grouped pages, you can view just the details you want by expanding and collapsing group headers. You can also sort and filter records.

5.4.3 Analyze data and make projections:

You can organize data in different ways using a Pivot table list, make projections and do complex calculations using a spreadsheet control and view data graphically in a chart.

5.4.4 Display HTML text:

You can store HTML code in fields in your database and displays it as formatted HTML text on the page. For example, if a value in field includes the HTML tag that formats text as italic, <I> Text</I>, you can use a bound HTML control on the page to display the value in italic text.

5.4.5 Use familiar design tools:

In design view, create pages using toolbars, the tool box, themes and other features that are similar to the tools you use to create forms and reports.

5.4.6 Microsoft Access project:

Easily create a client/server application: A Microsoft Access project (.adp) is new type of Access file that provides efficient, native-mode access to a Microsoft SQL server data base through the OLE DB component architecture. Using an Access project, you can easily create a client / server application

Work with an Access project working with a Microsoft Access project is very simply to working with an Access database .The process of creating forms, reports data access pages, macros and modules is the same. Once you connect to an SQL Server database, you can view, create modify and delete tables,

views, stored procedures and database diagrams using the Microsoft SQL server design tools.

Use Microsoft data engine (MSDE): MSDE is a new technology that provides local data storage compatible with SQL Server 7.0. Think of MSDE as a client /server data engine alternative to the file server Microsoft Jet database engine. It is designed and optimized for use on smaller computer systems, such as a single user computer or small working sever.

Upsize data and objects by using the upsizing Wizard: The upsizing wizard upsize a Microsoft Access database (.mdb) to anew or existing Microsoft SQL Server 6.5 and 7.0 data base or new Microsoft Access project(.adp) but upsizing the data and data definitions and migrating database objects.

Working with data and database design; Microsoft Access provides many new features that make working data and designing a database even easier.

- Use record-level locking a Microsoft Access database now supports record-level locking, in addition to page-level locking (with locks all records on a4k page).You enable the locking level with the new database option ,Open databases using record level locking (Tools menu ,Options command ,Advanced Tab).The actual level that is used depends on how the Access database is programmed.
- Find and replace you can now move freely between the find and replace dialog boxes and the data in the views or window.
- View related data in sub-datasheet to view and edit related or joined data in a table, query o r form datasheet or in a sub form all from the same view. for example , in a Northwind sample database the suppliers table has a one-to-many relationship with the products

table's far each row of the suppliers table in datasheet view ,you can view edit the related rows of the products table in a sub-datasheet.

- Automatically fix errors caused by renaming fields name autocorrect automatically corrects common side effects that occur when you rename forms, reports, tables, queries, fields, text boxes or other controls.
- Take advantage of Unicode support use the characters of any language that Unicode supports in your data .use Unicode compression to offset the effects of Unicode's increased storage space requirements.
- Work with the euro: To easily display euro amounts with other currencies, you can use the euro setting(#,##,## #) of the format property to indicate a euro amount .Alternatively ,you can enter the euro symbol by pressing ALT+0128 on the numeric keypad with NUM LOCK on. When you paste or import data with that contains the euro symbol from Microsoft Excel into Microsoft access, Access stores the euro symbol regardless of the currency symbol defined in Regional Settings in Windows Control Panel. Finally, you can use the Euro Convert function to convert one currency to another by using the euro as an intermediary. More about working with the euro.
- Print relationships: Print a report of the relationship in your access database as they appear in the Relationship Window
- Use the keyboard to manage relationships: use the keyboard to create, edit and delete relationship and joins.

- Use Microsoft activeX data objects (ADO): use Microsoft activeX data objects (ADO) to access and manipulate data in a database server through any OLE DB provider.

5.4.7 Database Window:

The database window in Microsoft access provides a variety of options for viewing and manipulating database objects.

- Use the Database window toolbar: quickly find commands for creating opening, or managing database objects.
- Use the objects bar view database object in the objects bar –its vertical orientating makes it easier to use
- Organize database objects into groups' click the Groups bar to view your groups, which can contain shortcuts to database objects of different types.
- Use new object shortcuts in the Database window, quickly create a new database object by using a wizard, or open a new database object in design view.
- Customize how you select and open objects in the databases window if you choose, change the default behavior so that you select a database object by resting the pointer over it, and open an object by single-clicking it.
- Select an object by typing its name. For example, select the Shipper's table while viewing the list of table objects by typing it.

Securing, Marinating, and Converting a database:

- Secure your access database with the User-level security Wizard. The User-level security wizard is now much easier to use and is the preferred method for defining user-level on a Microsoft access database for most common security schemes,
- Protect your code with visual Basic for application password a Visual Basic applications password that you create in the Visual Basic Editor; they are no longer protected by user-level security.

- Use the improved Compact utility Compact Microsoft Access databases and Microsoft access projects with an improved utility that combines compacting and repairing into a single process, and is safer and more effective.
- Compact automatically select Compact on Close to automatically compact a Microsoft Access databases or Microsoft access project when you close it.

5.4.8 Forms and Reports:

Microsoft access provides new features to make it faster and easier to create great-looking forms and reports.

- Groups' text boxes and other controls use the Group command on the Format Menu to group related text boxes and other controls on form or report.
- Define conditional formatting rules for text boxes and other controls Use the conditional Formatting command on Format menu to define the font color, font size, control background color, and other visual information that gives feedback to users as they enter data on a form.
- Create forms and report for SQL server database in a Microsoft access project, create forms and reports for a SQL server Database with the same tools that you use to create them in a Microsoft access Database.
- Distribute reports to users who do not have Microsoft Access Export Microsoft access reports to report snapshot (.snp) file format. You can use snapshot viewer to view, print, and mail report snapshots.

5.4.9 Replication:

Replication commands on the Microsoft Access Tools menu in Microsoft Access database (.mdb) and in a Microsoft access project (.adp) allow you to create replicas and synchronize them on demand as you work in Microsoft Access.

- Replication in an access project: replication data in an access project by creating publications and subscriptions.
- Jet and replications Objects (JRO): JRO includes methods and properties developers can use the programmatically replicate and synchronize database and Access projects.
- Web Server replication: work on files even when you are no longer connected to a web server.
- Replica Priority: a replica in a replica set is assigned a priority when it is created. The highest priority replica wins in the case of synchronization conflict.
- Prevent deletes option: Prevent users from deleting records in a replica.
- Local and anonymous replicas: create local replicas and anonymous replicas, Local and anonymous replicas can synchronize only with their parent, global replica. Anonymous replicas are recommended for Internet applications if you need a large number of replicas.
- Conflict viewer: the conflict Viewer is the default tool in Access to reconcile and resolve synchronization conflict
- Row Level Tracking Property: when this table property is set to true, it indicates that conflicts are tracked based at the row-level of a table. Conflicts are tracked at the column-level by default.
- Conflict Function Property: this property is used to replace the Microsoft access Conflict viewer with a customized procedure that assists users in resolving synchronization conflicts.
- Retention period: controls the number of day's non synchronized records are retained in the system tables.

- Synchronization conflicts: A single mechanism is now used to record and resolve conflicts and errors, making it easier to resolve conflicts. Whenever a conflict occurs, a winning change is selected and applied in all replicas and the losing change is recorded as a conflict at all replicas.
- Column-level conflict resolution: Conflicts are determined at the column level. Changes to the same record in two different replicas cause a synchronization conflict only if the same column or field is changed.



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 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.

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

TESTING STRATEGIES

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.

Recovery Testing:

It is a system test that forces the system to fail in a variety of ways and verifies that the recovery is properly performed.

Security Testing:

Attempts to verify the protection mechanisms built into the system.

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.

SCREENS



CONCLUSION

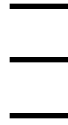
The need for the Health Center to computerize the application processing and servicing the Patients request through automated modules is most necessary and now inevitable.

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 Center 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.



BIBLIOGRAPHY

The following are the details of books and the sites which have details regarding the Swings technology which is used in this project.

Java 2 Core by sun publications

The Complete Reference, (Herbert Schildt)

Web Sites:

www.java.sun.com

www.JavaGalaxys.com

www.oreilly.com