**Project Report**

**On**

**“Hospital Management System”**

Submitted in the Partial fulfillment of the requirement for the Award of Degree of

**Bachelor of Technology**

in

**COMPUTER SCIENCE & ENGINEERING**

Batch

(2017-2021)



**Subject-Software Engineering**

**(ACCS-16504)**

|  |  |
| --- | --- |
| **Submitted to**  **Er. Bhuvnesh Kumar** | **Submitted by**  Manish sharma (1701294)  Lakshdeep salaria (1701286)  Kamlesh tak(17012)  MD Immamudin(17012) |

# Introduction to subject

# Software Engineering

Software engineering is an engineering discipline that’s applied to the development of software in a systematic approach (called a software process).

It’s the application of theories, methods, and tools to design build a software that meets the specifications efficiently, cost-effectively, and ensuring quality.

It’s not only concerned with the technical process of building a software, it also includes activities to manage the project, develop tools, methods and theories that support the software production.

*Not applying software engineering methods results in more expensive, less reliable software, and it can be vital on the long term, as the changes come in, the costs will dramatically increase.*

Different methods and techniques of software engineering are appropriate for different types of systems. For example, games should be developed using series of prototypes, while critical control systems require a complete analyzable specification to be developed.

## Computer Science Vs Software Engineering

Computer science focuses on the theory and fundamentals, like algorithms, programming languages, theories of computing, artificial intelligence, and hardware design, while software engineering is concerned with the activities of developing and managing a software.

# Software Engineers

The job of a software engineer is difficult. It has to balance between different people involved, such as:

* **Dealing with users**: User don’t know what to expect exactly from the software. The concern is always about the ease of use and response time.
* **Dealing with technical people**: Developers are more technically inclined people so they think more of database terms, functionality, etc.
* **Dealing with management**: They are concerned with return on their investment, and meeting the schedule.

**Dual Role of Software:**

**1. As a product –**

* It delivers the computing potential across network of Hardware.
* It enables the Hardware to deliver the expected functionality.
* It acts as information transformer because it produces, manages, acquires, modifies, displays, or transmits information.

**2. As a vehicle for delivering a product –**

* It provides system functionality (e.g., payroll system)
* It controls other software (e.g., an operating system)
* It helps build other software (e.g., software tools)

**Objectives of Software Engineering:**

1. **Maintainability –**  
   It should be feasible for the software to evolve to meet changing requirements.
2. **Correctness –**  
   A software product is correct, if the different requirements as specified in the SRS document have been correctly implemented.
3. **Reusability –**  
   A software product has good reusability, if the different modules of the product can easily be reused to develop new products.
4. **Testability –**  
   Here software facilitates both the establishment of test criteria and the evaluation of the software with respect to those criteria.
5. **Reliability –**  
   It is an attribute of software quality. The extent to which a program can be expected to perform its desired function, over an arbitrary time period.
6. **Portability –**  
   In this case, software can be transferred from one computer system or environment to another.
7. **Adaptability –**  
   In this case, software allows differing system constraints and user needs to be satisfied by making changes to the software.

**Program vs Software Product:**

1. Program is a set of instruction related each other where as Software Product is a collection of program designed for specific task.
2. Programs are usually small in size where as Software Products are usually large in size.
3. Programs are developed by individuals that means single user where as Software Product are developed by large no of users.
4. In program, there is no documentation or lack in proper documentation.  
   In Software Product, Proper documentation and well documented and user manual prepared.
5. Development of program is Unplanned, not Systematic etc but Development of Software Product is well Systematic, organised, planned approach.
6. Programs provide Limited functionality and less features where as Software Products provides more functionality as they are big in size (lines of codes) more options and features.

# Types of Software Testing

### Introduction:-

Testing is a process of executing a program with the aim of finding error. To make our software perform well it should be error free.If testing is done successfully it will remove all the errors from the software.

### Principles of Testing:-

(i) All the test should meet the customer requirements  
(ii) To make our software testing should be performed by third party  
(iii) Exhaustive testing is not possible.As we need the optimal amount of testing based on the risk assessment of the application.  
(iv) All the test to be conducted should be planned before implementing it  
(v) It follows pareto rule(80/20 rule) which states that 80% of errors comes from 20% of program components.  
(vi) Start testing with small parts and extend it to large parts.

### Types of Testing:-

#### 1. Unit Testing

It focuses on smallest unit of software design. In this we test an individual unit or group of inter related units.It is often done by programmer by using sample input and observing its corresponding outputs.  
Example:

a) In a program we are checking if loop, method or

function is working fine

b) Misunderstood or incorrect, arithmetic precedence.

c) Incorrect initialization

#### 2. Integration Testing

The objective is to take unit tested components and build a program structure that has been dictated by design.Integration testing is testing in which a group of components are combined to produce output.

Integration testing is of four types: (i) Top down (ii) Bottom up (iii) Sandwich (iv) Big-Bang  
Example

(a) Black Box testing:- It is used for validation.

In this we ignores internal working mechanism and

focuses on **what is the output?**.

(b) White Box testing:- It is used for verification.

In this we focus on internal mechanism i.e.

**how the output is achieved?**

#### 3. Regression Testing

Every time new module is added leads to changes in program. This type of testing make sure that whole component works properly even after adding components to the complete program.  
Example

In school record suppose we have module staff, students

and finance combining these modules and checking if on

integration these module works fine is regression testing

#### 4. Smoke Testing

This test is done to make sure that software under testing is ready or stable for further testing  
It is called smoke test as testing initial pass is done to check if it did not catch the fire or smoked in the initial switch on.  
Example:

If project has 2 modules so before going to module

make sure that module 1 works properly

#### 5. Alpha Testing

This is a type of validation testing.It is a type of acceptance testing which is done before the product is released to customers. It is typically done by QA people.  
Example:

When software testing is performed internally within

the organization

#### 6. Beta Testing

The beta test is conducted at one or more customer sites by the end-user of the software. This version is released for the limited number of users for testing in real time environment  
Example:

When software testing is performed for the limited

number of people

#### 7. System Testing

In this software is tested such that it works fine for different operating system.It is covered under the black box testing technique. In this we just focus on required input and output without focusing on internal working.  
In this we have security testing, recovery testing , stress testing and performance testing  
Example:

This include functional as well as non functional

testing

#### 8. Stress Testing

In this we gives unfavorable conditions to the system and check how they perform in those condition.  
Example:

(a) Test cases that require maximum memory or other

resources are executed

(b) Test cases that may cause thrashing in a virtual

operating system

(c) Test cases that may cause excessive disk requirement

#### 9. Performance Testing

It is designed to test the run-time performance of software within the context of an integrated system.It is used to test speed and effectiveness of program.  
Example:

Checking number of processor cycles.

This article is contributed by **Kritka**. If you like GeeksforGeeks and would like to contribute, you can also write an article using [contribute.geeksforgeeks.org](http://www.contribute.geeksforgeeks.org/) or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Introduction to project

**ONLINE HOSPITAL MANAGEMENT SYSTEM**

The Hospital Management System is designed for any hospital to replace their existing manual paper based system. The project Hospital Management system includes registration of patients, storing their details into the system, and also computerized billing in the pharmacy, and labs. The purpose of the project entitled as “HOSPITAL MANAGEMENT SYSTEM” is to computerize the Front Office Management of Hospital to develop software which is user friendly simple, fast, and cost – effective. The new system is to control the information of patients. Room availability, staff and operating room schedules and patient invoices. I have designed the given proposed system in the JSP to automate the process of day to day activities of Hospital like Room activities, Admission of New Patient, Discharge of Patient, Assign a Doctor, and finally compute the bill etc., All the control is under the administrator and the other members have the rights to just see the records not to change any transaction or entry.  These services are to be provided in an efficient, cost effective manner, with the goal of reducing the time and resources currently required for such tasks .  
  
**Proposed  hospital management system**  
  
“Hospital Management System” is developed to computerize the following functions that are performed by the system:

1. Using this hospital management system user can take online Appointments with the doctors.
2. Admin can register new patient through this system.
3. Admin can also check Discharge Detail of a patient form discharge report generated by the system.
4. Discharge of Patient – admin can discharge patient through the system and this information will display to other users like receptionist and accountant.
5. This system is also keep records of Patient’s Disease.
6. Admin can also check the report of admitted patient or admitted patient any time. Like total number of admitted patient in the given time in the hospital.
7. Admin can also check the availability of the doctors in the hospital.
8. Administrator can add new doctors in the system and can also check all the doctors list of the hospital.

SRS of Hospital Management System

# Table of Contents

Contents

[Table of Contents 2](#_Toc18107347)

[1. Introduction 3](#_Toc18107348)

[1.1 Purpose 3](#_Toc18107349)

[1.2 DocumentConventions 3](#_Toc18107350)

[1.3 Intended Audience and ReadingSuggestions 3](#_Toc18107351)

[1.4 Project Scope 3](#_Toc18107352)

[1.5 References 4](#_Toc18107353)

[2. OverallDescription 5](#_Toc18107354)

[2.1 ProductPerspective 5](#_Toc18107355)

[2.2 ProductFeatures 5](#_Toc18107356)

[2.3 User Classes and Characteristics 6](#_Toc18107357)

[2.4 Operating Environment 6](#_Toc18107358)

[2.5 Design and ImplementationConstraints 6](#_Toc18107359)

[2.6 UserDocumentation 7](#_Toc18107360)

[2.7 Assumptions andDependencies 7](#_Toc18107361)

[3. External Interface Requirements 8](#_Toc18107362)

[3.1 UserInterfaces 8](#_Toc18107363)

[3.2 HardwareInterfaces 8](#_Toc18107364)

[3.3 SoftwareInterfaces 9](#_Toc18107365)

[3.4 Communications Interfaces 9](#_Toc18107366)

[4. Other NonfunctionalRequirements 10](#_Toc18107367)

[4.1 PerformanceRequirements 10](#_Toc18107368)

[4.2 Safety Requirements 10](#_Toc18107369)

[4.3 Security Requirements 10](#_Toc18107370)

[4.4 Software Quality Attributes 11](#_Toc18107371)

[Appendix A: Glossary 11](#_Toc18107372)

# 

# Introduction

## Purpose

The purpose of the Hospital Management System (HMS) is to create a Free and Open Source Software (FOSS) Hospital Management System.

This project is being developed mainly for our client, Appalachian Health Network (AHN), but will be extensible enough to be adapted and customizable for deployment and integration into any hospital network.

The SHMS contains five distinct sub-systems:

* + - Radiology Information System(RIS)
    - Picture archiving and Communication System(PACS)
    - Image Acquisition Modality System(IAM)
    - Admit-Discharge-Transfer/Patient Registration System(ADT/PRS)
    - Hospital Management Systems(HIS)

## DocumentConventions

This document will use IEEE format. For clarity, acronyms and technical jargon, deemed uncommon by the author, will be annotated and included in the glossary. The format for headings is as followed:

Major headings are in **bold 12pt font,** and concurrent headings in **bold 11pt font**. Sections are in the format of x.y, where x and y are real, positive integers.

## Intended Audience and ReadingSuggestions

This Software Requirements Specification document is intended for software engineers, system testers and software designers in developing, testing, and producing the SHMS and for the project. It is suggested to read the sections sequentially, and to reference the appendices as one progresses, in order to clarify jargon terms and definitions*.*

## Project Scope

This SRS details the development of the Hospital Management System project and the five subsystems.

The scope of the RIS subsystem is to create a generic FOSS RIS program, which can be customized for deployment and integration into any hospital’s use; however, once developed, the RIS will be customized to the needs of our client, Allegheny Health Network.

The PACS scope is to create an archive and communication system that covers how medical pictures are gathered, stored, shared amongst medical professionals, and secured for confidentiality.

The ADT/PRS subsystem is to streamline the process of handling patient data. A patient’s information is store and share with the appropriate people automatically under this subsystem.

## References

WikipediaHISpage:<https://en.wikipedia.org/wiki/Hospital_information_system>AppalachianHealthNetwork: <https://www.ahn.org/>

# OverallDescription

## ProductPerspective

The Hospital Management System is an open source system comprising of five different subsystems. The five subsystems are as follow:

The FOSS RIS project is a separate program, which is a component of a larger FOSS Hospital Management System (HMS), similar to how Microsoft Word is a separate program inside Microsoft Office suite. In the FOSS HMS system, the FOSS RIS program performs all HIS operations. The RIS module uses the shared, global variables, enums, framework, and used to create the other FOSS HIS program components, just like with Microsoft Office. All data exclusive to the RIS module will be programmed in the RIS module.

The driving principle of this PACS is to automate and provide the infrastructure to digitally control the storage and transportation of images taken with compatible devices within a general hospital.

The ADT/PRS subsystem stores patient data, which other subsystems can access as required. This is accomplished by granting the other systems access to this subsystem’s patient database.

## ProductFeatures

The HMS has five subsystems and these subsystems shall perform the following features:

* + - The RIS subsystem shall include patient list management, radiology department workflowmanagement, request and document x-ray scanning, result entry, and reporting and printout/faxing and emailing of clinicalreports.
    - The PACS subsystem shall perform image importing/capturing, image encryption, local image storage, remote image retrieval, image compression, image display, and imageprocessing.
    - The ADT/PRS subsystem shall allow an administrator to enter patient information, such as name,age, etc. That information is then stored, and shared with other users as appropriate. It shall also alert the medical staff when a patient that requires different treatment is admitted, such as some with an infectiousdisease.

## User Classes and Characteristics

The entire FOSS HMS suite program has a set of users, each with different security privileges. These user types are head doctor/nurse, and doctor/nurse. The head doctor/nurse can control most of the system, can transfer data in/out of hospital networks and to other doctors/nurses who need the patient’s medical information from the patient database, and possesses read/write permissions on sending/receiving data to/from the database. This is the same for the ADT/PRS subsystem, where only the administrator can enter/edit data. All access and data transfers/receiving is logged, in order to maintain a level of transparency, in order to prevent abuse of the system, and in order to hunt down any unauthorized users or

hackers.

## Operating Environment

The FOSS HMS program runs on Windows 7, Windows 8, Windows 10 for 32-bit/x86 and 64-bit/x64 PC architectures. The software for the RIS subsystem will be written in C#, using Microsoft Visual Studio 2010. The program will be GUI-based (like with most modern Windows software).

The HIS subsystem will run off a Cloud-Based Platform. The Cloud-based server will utilize Oracle or SQL database running on the cloud. The operating system shall be a MS-Windows or UNIX. Integration to the server shall be done via a HTTPS, SFTP, or VPN to create, update, fetch, or delete data.

## Design and ImplementationConstraints

Items and issues that may limit the options available to the software developers are legal and ethical constraints with regard to HMS development and medical practices, and possible social and legal opposition by HIS corporations who loathe FOSS software. Moreover, parallel threads will need to take place in the larger HIS operation, which will require research in how to program and operate with several, parallel-running threads in the same application

Due to the large nature of the project, keeping track of the source code between the developer sub-teams will be difficult. Smaller bug fixes can be downloaded as hotfixes, also available for download from the website. Updates can be discovered by manually browsing our website, or by pulling down the help

tab, which has a “Check for Updates…” feature.

## UserDocumentation

The application will come with an “About” tab, which will allow users to access the offline and online HTML .hlp help manual. This manual will be updated with each new service pack. Other user documentation includes one user manual for lowest level users, one technical document describing the functionality of the sub- section in detail for use of technicians, one copy of documentation and link to current source for future contributors.

## Assumptions andDependencies

The developers, assume that we will have to “pave our own way” concerning programming the majority of the application, due to the mostly closed-source and secretive nature of major HMS software. For what we cannot find from open documentation and research, it is assumed that we will have to deduce how HIS standards and protocols work from observing external behaviors found in existing HIS software, and we will have to replicate the results using our own code and other FOSS applications and libraries.

It is assumed that the PACS will be used in a Hospital Environment by untechnical users. It is assumed that the infrastructure for capturing digital images in either .JPG, .GIF, .DICOM, etc will exist. It is assumed that the System will be networked, and capable of routing to an internet gateway.

# External Interface Requirements

## UserInterfaces

#### The user interface of the software will use standard Windows API and GUIs using C#. All five subsystems will utilize the Windows-style GUI. Due to the varying age group of users (from younger interns to middle-aged doctors), the GUI needs to adapt to the age group’s GUI and computing preference. The application has two (2) GUI styles:

* + - **Classic**
      * For olderusers
      * Similar to Microsoft OfficeXP
      * Uses the traditional GUI style from Microsoft XP and older,has
        + Tabs
        + Buttons
        + Dialogboxes
        + Listboxes
        + Etc
      * See the XP picturebelow.
    - **W7 Ribbon**
      * Uses the “ribbon” as seen in Microsoft Office2007
      * Suited for younger to middle-agedusers
      * Instead of primarily using menu dropdowns
        + Uses dynamictabs
        + See the W7 picturebelow

## HardwareInterfaces

Database interfacing will use standard TCP/IP protocols, but using FOSS libraries, for computers connected to the internet via LAN Ethernet cables. Due to security concerns and the ease of hackers cracking WiFi hotspots, WiFi internet support with the software is prohibited by the software. The software detects whether a LAN or WiFi connection is used, and will terminate the program if WiFi is detected to be in use for the internet connection.

The PACS subsystem shall utilize USB support for importing photos directly from the devices into the PACS. Meanwhile, the HIS subsystem shall enter patient information with a secure connection through an encrypted computer or tablet being used by the medical staff of the hospital. An encrypted computer that has a secure VPN is the preferred method of communicating to the database.

## SoftwareInterfaces

The FOSS RIS module will be able to interface with the bigger, FOSS HMS module, and will inherit the global, shared, structs, enums, functions, and variables, as well as use its own data pertinent and exclusive to itself. It will interface with standard Windows API and GUI. Data that will be shared between computers and instance of the software being run will be pushed and pulled from the patient database as needed/requested. This database can be configured and deployed for usage in case-specific usages for each hospital network that uses the software in its own LAN or intranet.

## Communications Interfaces

Communication interfaces will use TCP/IP for data transmission and SMTP/HTTP for generating emails of reports from the software. FTP can also be used in pushing generated document reports to a hospitals FTP server. All communication interfaces should have high baud data Tx and Rx rates ranging from Mbps to Gbps. FOSS TLS or higher encryption standards are a must-have, high priority requirement. All offline and online access will be monitored, for transparency purposes, and in order to reduce abuse and unauthorized access of the system.

# Other NonfunctionalRequirements

## PerformanceRequirements

The software should have **high** performance and **low** failure rates. The hardware and software should be able to transmit/receive data from databases with high baud rates, ranging from Mbps to Gbps. Machines should have **all** recent Windows updates installed, and have their security not compromised by viruses. Machines must have firewalls installed and active virus scanning software in usage. All database queries and data receiving/transmitting should be done using TLS or higher security transmission.

## Safety Requirements

In regards to the RIS subsystem and for the safety of the patients, all patients must wear lead vests for the body part being X-Rayed, and the X-Ray equipment must be regularly maintained, inspected, and used responsibly. The software shall have built in safeguards which shall terminate any imaging process should it produce an unsafe level of radiation. Should a user attempt to send a dosage of radiation which exceeds recommended levels the imaging session shall be terminated and any previously captured images stored.

All offline and online access will be monitored, for transparency purposes, and in order to reduce abuse and unauthorized access of the system. Most actions will require a yes/no confirmation before it will actually be performed.

## Security Requirements

All data receiving and transmissions should be done using FOSS TLS or higher encryption, in order to keep the patient’s private medical and social security information out of the wrong hands. In addition, all computers **must** have firewalls, and be operating on a **LAN** internet connection, **not** a WiFi connection. Moreover, all computers must have all recent Windows updates installed, and **must** have **solid** anti-virus software. Also, the user-permissions system mentioned in §2.3 will be implemented. Before any user can access the system, they shall be required to input a company username, an ID number, and a password. Each password shall be required to be between 8-12 characters in length and shall be required to contain at least one capital letter, one number, and one special character.

Passwords will need to be changed every half-year, with a unique password.

## Software Quality Attributes

Flexibility, reusability, robustness, and maintainability of the HMS system should be maximized, in order for clients to be able to deploy custom settings of the FOSS HMS to their individual hospital network needs.

# Appendix A: Glossary

##### 7Zip

* 1. Highly acclaimed and functional, multi-platform, FOSS compression file formatstandard
  2. Utilizes GNU LGPLLicense
  3. See <http://www.7-zip.org/>for moreinformation

##### Baudrate

* 1. Rate of transfer of data over the internet/network
  2. Measured in bit per second(bps)

##### Bit

* 1. BinaryDigit
  2. One zero(0) or one(1).

##### Database

* 1. Big memory address block which contains large set ofdata
  2. With subsets and fields that can search for by filter, read, andwritten

##### FOSS

* 1. **F**ree and **O**pen **S**ource**S**oftware
  2. Software that is freeware, and which has its source code available, for others to modify under the GNU SoftwareLicense
  3. See<https://www.gnu.org/copyleft/gpl.html>for GPLinformation.

##### HIS

* 1. **H**ospital **I**nformation**S**ystem
  2. Element of healthinformatics
  3. Focuses mainly on the administration needs ofhospitals
  4. In many implementations, a HIS is a comprehensive, integrated informationsystem designed to manage all the aspects of a hospital'soperation
     1. Medical
     2. Administrative
     3. Financial
     4. Legalissues
     5. Corresponding processing ofservices.
  5. (Paraphrase source:<https://en.wikipedia.org/wiki/Health_informatics>)

1. **HMS**
   1. **H**ospital **M**anagement**S**ystem

##### LAN

* 1. **L**ocal **A**rea**N**etwork
  2. Refers to computer connected to the internet/each other via physical EthernetCable

##### Ribbon

* 1. Refers to the Windows GUIelement that dynamically changes the items in a menubar

1. **SRS**
   1. **S**oftware **R**equirements**S**tatement
   2. Statement clarifying the what a software project is supposed to be engineered todo
   3. Specifies the limits, constraints, and big-picture, abstract plan of the softwareengineering

##### SQL

* 1. **Structured Query Language**

##### W7 RibbonStyle

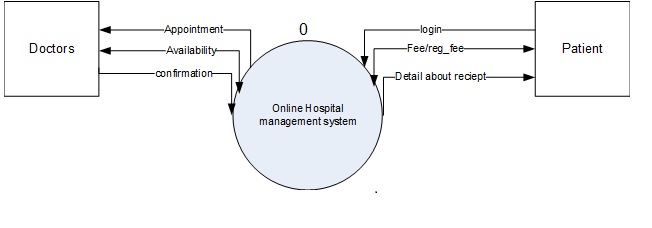
* 1. A GUI style in the HISsoftware which looks and acts like the ribbon in Microsoft Office2007

##### Windows API

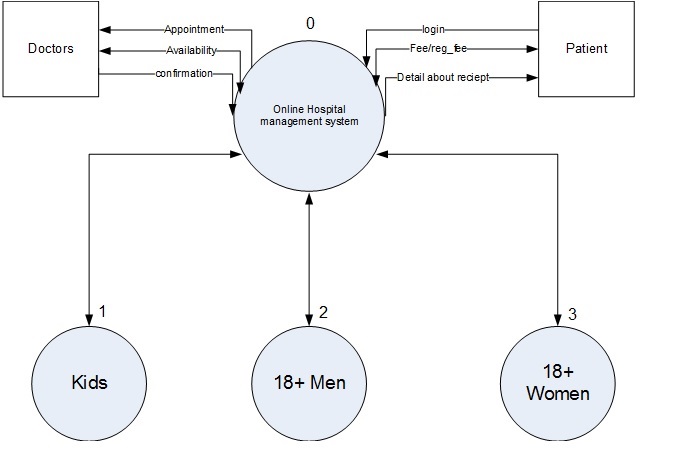
* 1. Windows **A**pplication **P**rogramming**I**nterface

**DFD**:-  Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation.

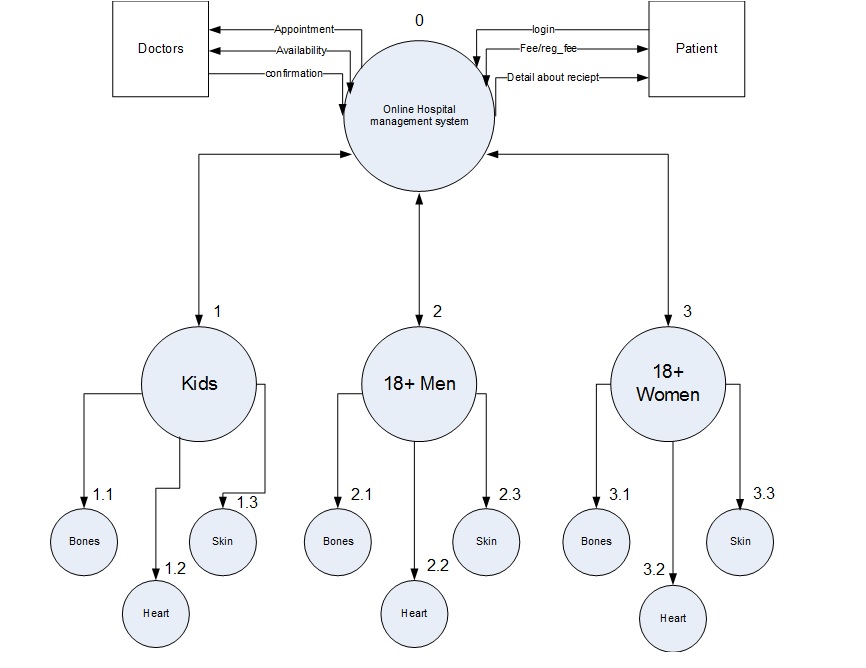
Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of the logical data flow.



0 LEVEL DFD



1 LEVEL DFD

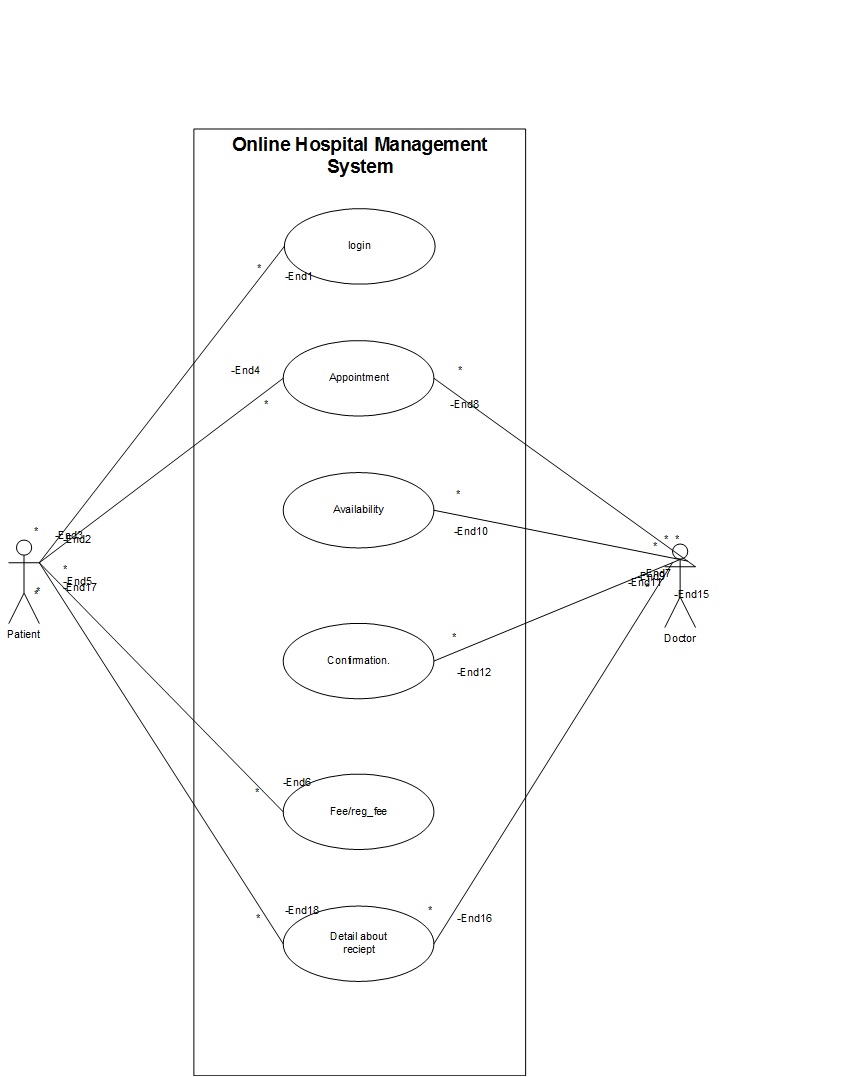


2 LEVEL DFD

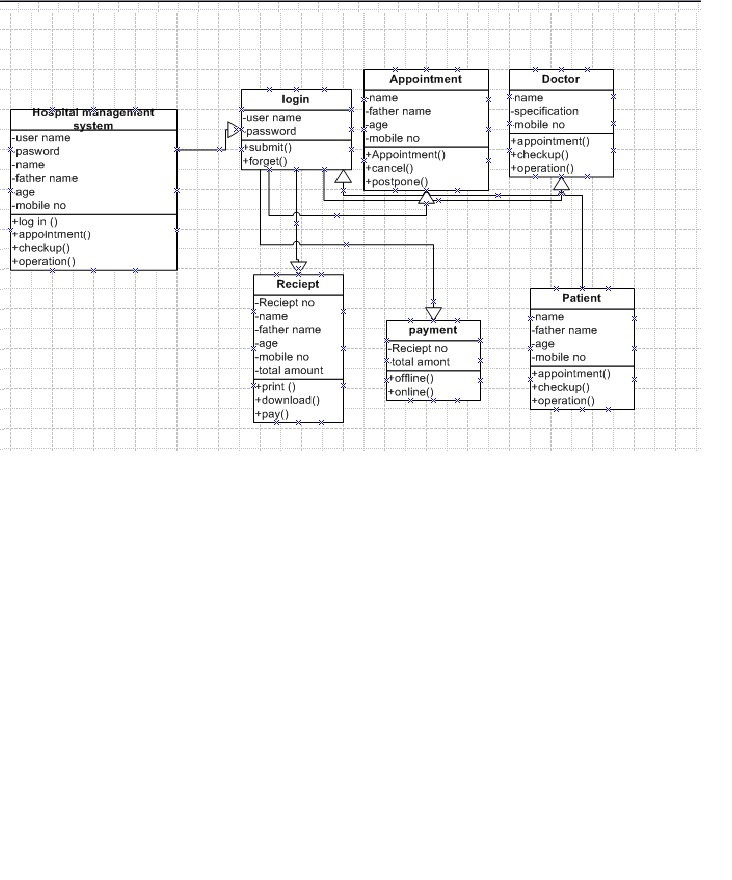
**UML**:- Unified Modeling Language (UML) is a general purpose modelling language. The main aim of UML is to define a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in other fields of engineering.

UML is not a programming language, it is rather a visual language. We use UML diagrams to portray the behavior and structure of a system. UML helps software engineers, businessmen and system architects with modelling, design and analysis. The Object Management Group (OMG) adopted Unified Modelling Language as a standard in 1997. Its been managed by OMG ever since. International Organization for Standardization (ISO) published UML as an approved standard in 2005. UML has been revised over the years and is reviewed periodically.

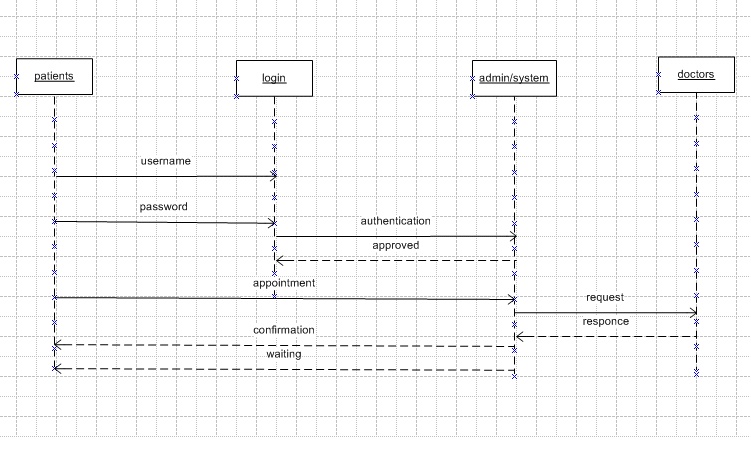
**USECASE DIAGRAM**: -A use case diagram is a dynamic or behavior diagram in UML**.**Use case diagrams model the functionality of a systemusing actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform



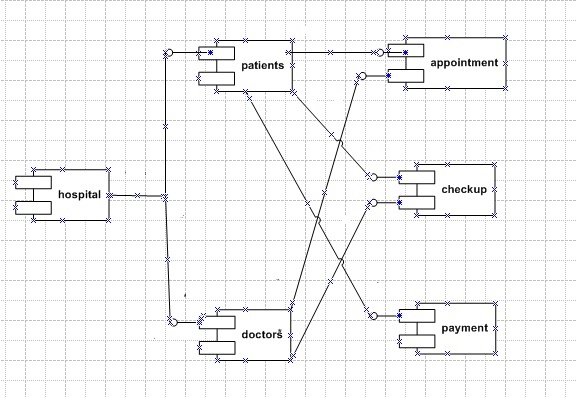
**CLASS DIAGRAM**: -In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structurediagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects



**SEQUENCE DIAGRAM**: -A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function.



**COMPONENT DIAGRAM**: - A component diagram, also known as a UMLcomponent diagram, describes the organization and wiring of the physical components in a system.Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.



### Unit Testing

Unit testing is the practice of testing small pieces of code, typically individual functions, alone and isolated. If your test uses some external resource, like the network or a database, it’s not a unit test.

Unit tests should be fairly simple to write. A unit tests should essentially just give the function that’s tested some inputs, and then check what the function outputs is correct. In practice this can vary, because if your code is poorly designed, writing unit tests can be difficult. Because of that, unit testing is the only testing method which also helps you write better code – Code that’s hard to unit test usually has poor design.

### Integration Testing

As the name suggests, in integration testing the idea is to test how parts of the system work together – the integration of the parts. Integration tests are similar to unit tests, but there’s one big difference: while unit tests are isolated from other components, integration tests are not. For example, a unit test for database access code would not talk to a real database, but an integration test would.

Integration testing is mainly useful for situations where unit testing is not enough. Sometimes you need to have tests to verify that two separate systems – like a database and your app – work together correctly, and that calls for an integration test. As a result, when validating integration test results, you could for example validate a database related test by querying the database to check the database state is correct.

