**CHAPTER 5**

**DESIGN**

**5.1 SYSTEM DESIGN**

System design is a reduction of an entire system by studying the various operations performed and their relationships within the system and the requirements of its success. One aspect of design is defining the boundaries of the system and determining whether or not the candidate system should consider other related system. System can be defined, as an orderly grouping of interdependent components can be simple or complex.

The idea of the systems has been most practical and necessary in computerizing the interrelationships and integration of operations, especially when using computers. Thus it’s a way of thinking organizations and their problems. An organization consists of several interrelated and interlocking components.

The most creative and challenging phase of the system life cycle is system design. The term design describes a final system and the process by which it is developed .It refers to the technical specifications that will be applied in implementing the candidate system .It also includes the construction of programs and program testing.

The first step in the system design is to determine how the output is to be produced and in what format. Samples of the output and the inputs are also presented .In the second step, input data and master files are to be designed to meet requirement of the proposed output .The processing phase’s system’s objectives and complete documentation.

Finally details related to justification of the system and an estimate of the impact of the candidate system on the user and organization are documented and evaluated by management as a step towards implementation. The final report prior to the implementation phase includes procedure flow chart, record lay outs, and a workable plan for implementing the KDMS system.

System design has two phases:

❖ Logical

❖ Physical

The logical design reviews the present physical system, prepares the input and output and also prepares a logical design walk- through .We have to deal with how to take entries required and whether and how to process the user data. Also we have to deal with how to present the data in an informative and appealing format .This design also involves the methodology to store, modify and retrieve data from the data base as per the requirement. Physical design maps out the details of the physical system, plans the system implementation, devices a test and implementation plan and new hardware and software. We have to decide how and where to store the input data and how to process it so as to present it to the user in an easy, informative and attractive manner .A major step in the design is the preparation of input and output report in a form acceptable to the user .In this a data entry operator can feed the relevant details asked by the system for a particular task as input.

**Module Description**

**Admin module**

Admin module is the important module which can access all modules in the integrated health club management system and handles staff management, user management, equipment management and billing. Admin can add new staff to system by entering their details such as name, block, address, email, phone number, date of birth and salary. Details of existing staff can be edited whenever necessary. By monitoring the existing staff, inefficient staff can be deleted. Admin can block the non-productive staff or unblock productive staff from the health club management system. When a registration request arrives to the system from a new user admin can either approve or reject those request. Invoices can be manually send or view by the admin and can also auto schedule invoices for the users. Using staff id, admin can view the attendance of the staff and the details of login time, logout time, total hours and salary. Admin can view attendance of each user and their class schedules from the admin panel. Peak time (rush time of the health club management system) and normal time of the system can viewed. Admin can send offers regarding discount/special programs and notifications to all or selected users in the health club management system. New equipment’s can be added with their name, type and building id and can view existing equipment or delete faulty/unused equipment’s inside health club and mark location of each equipment. Employee salary management can be executed like edit hourly wages and view average wages earned, wages earned for a time period. Admin can send notification to staff/members or schedule notification to staff/members. Status of each member can be viewed and admin can edit the recurring payments of member’s. Bills are calculated and viewed for the details like bill id, amount, discount etc.

**Virtual training and help desk using voicebot/chatbot**

**User bot**

* Using google voice to text convertor voice to text conversion is implemented.
* Using google text to voice convertor text to voice conversion is implemented.
* Using voicebot/chatbot new user registration is carried out through voice assisted/chat interface.
* User can ask doubts or suggestion to chatbot. With the help of dialog flow replay from the bot is generated.
* Using bot classes can be schedule or scheduled classes can be reschedule.
* Text chat and voice chat shifting option is available in the bot.
* Using voicebit/chatbot system can check availability of specific trainers and get their contact details.
* User can view payment history through bot and can pay fee online.

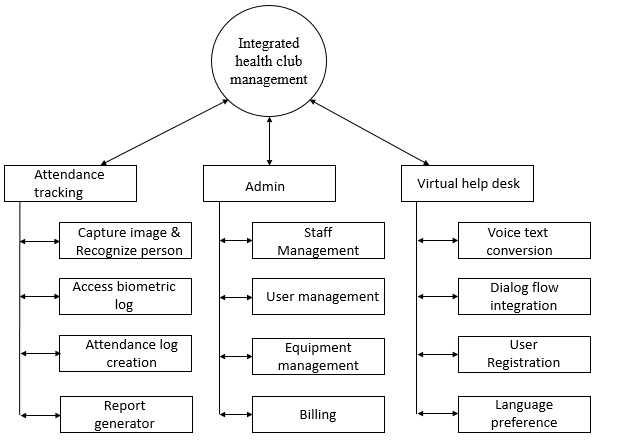
**Admin section for training**

* Admincan train the bot using the training interface.
* Admin can add static questions and dynamic using available api interface.
* Admin can set or change language preferences accordingly.
* Can allow users or block the users through chatbot.

**Attendance tracking module**

* Can fetch attendance from biometric modules and mark logs.
* Can fetch attendance using face recognition using openCV.
* Can generate auto billing based on attendance and can send report or invoices to the members

**System diagram**

****

**5.1.1 Data Flow Diagram**

A data flow diagram is a graphical technique that depicts information flow and transforms that are applied as data move from input to output. The DFD is used to represent increasing information flow and functional details. A level 0 DFD also called fundamental system model represents the entire software elements as a single bible with input and output indicated by incoming and outgoing arrows respectively.

The DFD is one of the most important tools used by the system analyst. It can be used for the visualization of data processing. DFD provides a graphical representation of the flow of data through an information system. DFD illustrates how data is processed by a system in terms of inputs and outputs. It is a picture of system processing and flow without excessive concern for details.

The DFD showing the top level of the system is called “Context Diagram”. It should be overview including basic inputs, processes and outputs. Then it is exploded in to more detailed lower level diagram that shows additional features of the system.

The purpose of DFD is to provide a semantic bridge between users and system developers. The diagrams are graphical, eliminating thousands of words, logical representations, modeling what system does; hierarchical, showing system at any level of details; and Jargon less, allowing user interaction and reviewing.

Data Flow Diagrams Notations

Process: 

Input/output: 

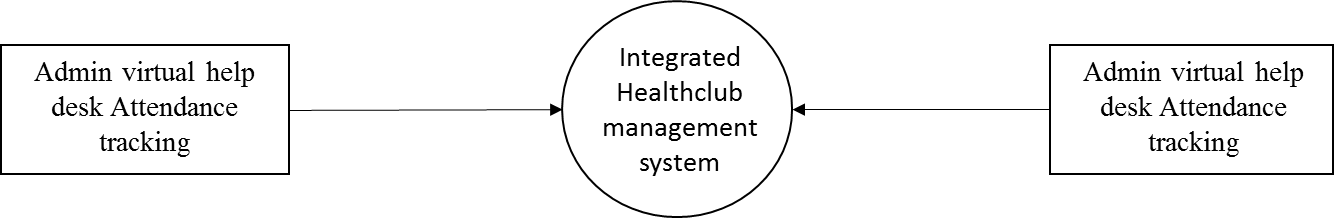
Flow of direction: 

Database/File:

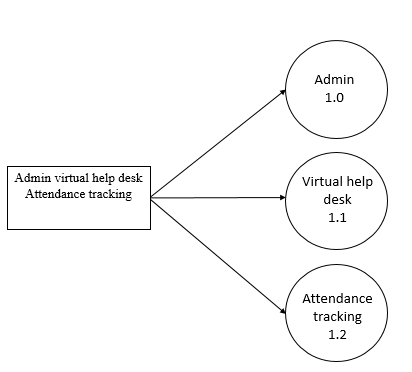


**5.1.2 Project DFD /UML**

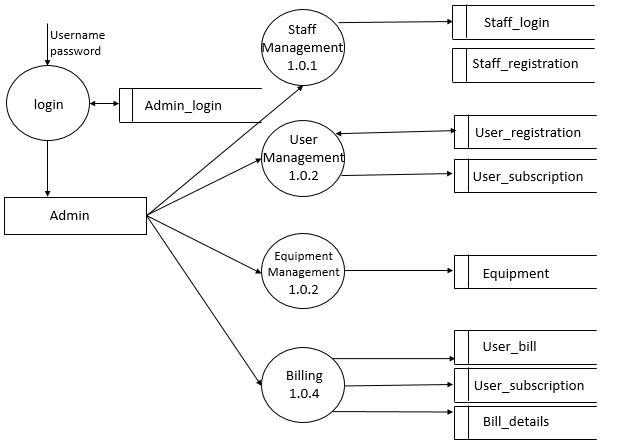
**Level 0 (context level)**

****

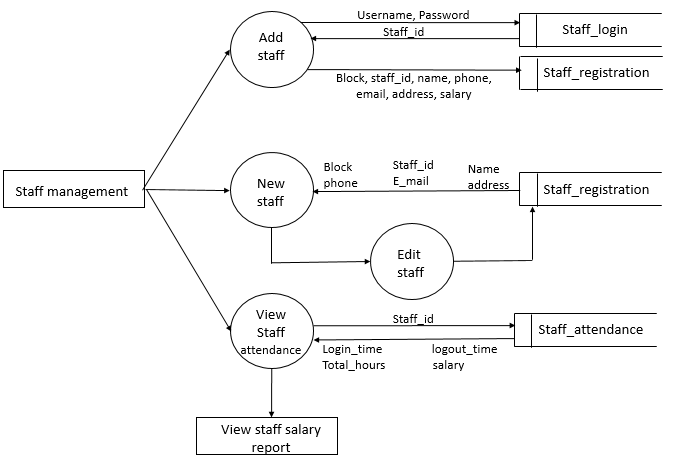
**Level 1**

****

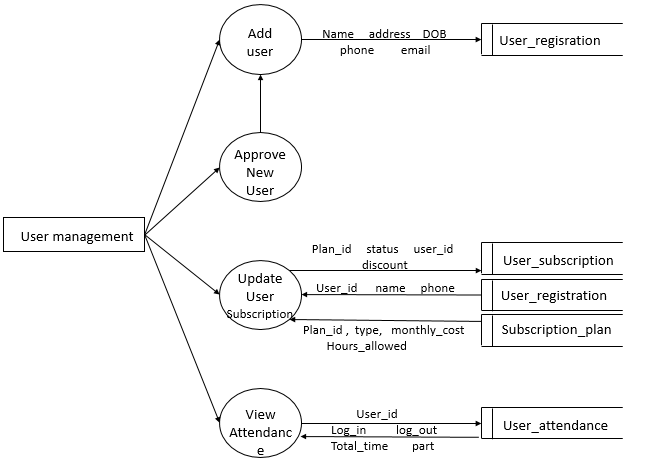
**Level 1.0 Admin**

****

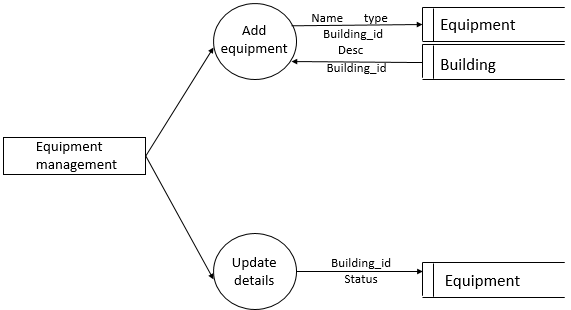
**Level 1.0.1**

****

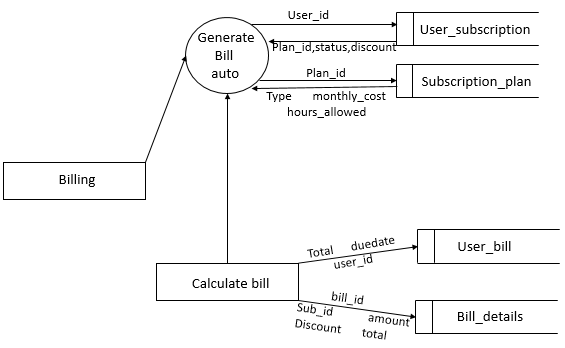
**Level 1.0.2**

****

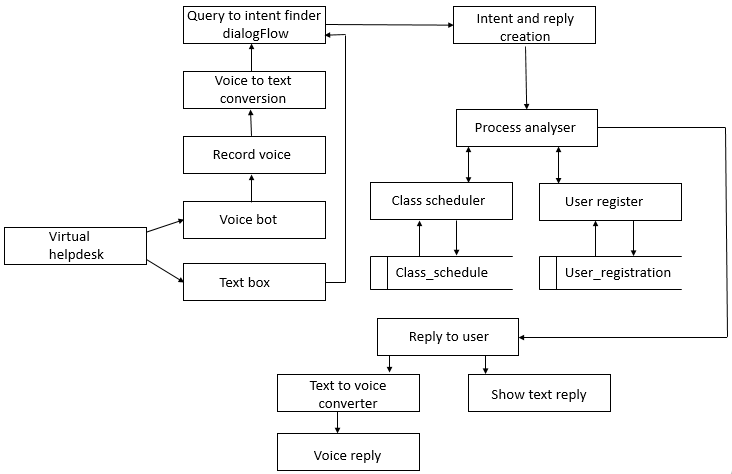
**Level 1.0.3**

****

**Level 1.0.4**

****

**Level 1.1**

****

**5.2 DATABASE DESIGN**

Table 5.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table name: Admin\_login primary key: Admin\_login\_id** | | | | |
| **Description : details of admin login** | | | | |
| **Field** | **Datatype** | **Size** | **Constraints** | **Description** |
| Admin\_login\_id | Integer | 11 | Primary key | Id of admin login |
| Username | Varchar | 15 |  | Username of admin |
| Password | Varchar | 15 |  | Password of admin |

Table 5.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table name:staff\_login primary key:Staff\_id** | | | | |
| **Description: details of staff login** | | | | |
| **Field** | **Datatype** | **Size** | **Constraints** | **Description** |
| Staff\_id | Integer | 11 | Primary key | Id of staff |
| Type | Varchar | 15 |  | Type of staff |
| Username | Varchar | 15 |  | Username of staff |
| password | Varchar | 15 |  | Password of staff |

Table 5.3

Table used: staff\_login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table name: staff\_registration primary key:Staff\_reg\_id** | | | | |
| **Description : details of staff egistration** | | | | |
| **Field** | **Datatype** | **Size** | **Constraints** | **Description** |
| Staff\_reg\_id | Integer | 11 | Primary key | Id of staff registration |
| Staff\_id | Integer | 11 | Foreign key | Id of staff |
| Block | Varchar | 10 |  | Block of staff |
| Name | Varchar | 30 |  | Name of staff |
| Phone | Varchar | 15 |  | Phone no of staff |
| Email | Varchar | 40 |  | Email id of staff |
| Address | Text |  |  | Address od the staff |
| Salary\_hour | Double |  |  | Salary hours of staff |

Table 5.4

Table used:staff\_login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table name:user\_registration primary key:User\_id** | | | | |
| **Description: details of user registration** | | | | |
| **Field** | **Datatype** | **Size** | **Constraints** | **Description** |
| User\_id | Integer | 11 | Primary key | Id of user |
| Name | Varchar | 15 |  | Name of the user |
| Address | Varchar | 30 |  | Address of user |
| Phone | Varchar | 15 |  | Phone no of user |
| Email | Varchar | 20 |  | Email of user |
| Dob | Date |  |  | Date of birth of user |

Table 5.5

Table used: staff\_login ,user\_registration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table name: class\_schedule primary key:Schedule\_id** | | | | |
| **Description: details of schedule** | | | | |
| **Field** | **Datatype** | **Size** | **Constraints** | **Description** |
| Schedule\_id | Integer | 11 | Primary key | Id of the schedule |
| User\_id | Integer | 11 | Foreign key | Id of the user |
| Staff\_id | Integer | 11 | Foreign key | Id of the staff |
| Start time | Datetime |  |  | Schedule’s start time |
| End time | Datetime |  |  | Schedule’s end time |
| Type | Varchar | 15 |  | Type of the schedule |

Table 5.6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table name: building primary key:building\_id** | | | | |
| **Description: details of building** | | | | |
| **Field** | **Datatype** | **Size** | **Constraints** | **Description** |
| Building\_id | Integer | 11 | Primary key | Id of the building |
| Name | Varchar | 30 |  | Name of the block |
| Number | Integer | 11 |  | Number of block |

Table 5.7

Table used: building

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table name: equipment primary key:Equipment\_id** | | | | |
| **Description: details of equipment** | | | | |
| **Field** | **Datatype** | **Size** | **Constraints** | **Description** |
| Equipment\_id | Integer | 11 | Primary key | Id of the equipment |
| Name | Varchar | 40 |  | Name of the equipment |
| Type | Varchar | 15 |  | Type of equipment |
| Building\_id | Integer | 11 | Foreign key | Id of building |
| Desc | Text |  |  | Descriptions |

Table 5.8

Table used: user\_registration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table name: user\_bill primary key:Bill\_id** | | | | |
| **Description : details of user bill** | | | | |
| **Field** | **Datatype** | **Size** | **Constraints** | **Description** |
| Bill\_id | Integer | 11 | Primary key | Id if the bill |
| User\_id | Integer | 11 | Foreign key | Id of the user |
| Total\_amount | Double |  |  | Total amount to be paid |
| Due\_date | Double |  |  | Due date of the payment |
| Paid\_amount | Double |  |  | Details of paid amount |

Table 5.9

Table used:building

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table name:subscription\_plan primary key:Plan\_id** | | | | |
| **Description: details of subscription plan** | | | | |
| **Field** | **Datatype** | **Size** | **Constraints** | **Description** |
| Plan\_id | Integer | 11 | Primary key | Id of plan |
| Type | Varchar | 15 |  | Type of plan |
| Building\_id | Integer | 11 | Foreign key | Id of building |
| Monthly\_cost | Double |  |  | Monthly cost plan |
| Hours\_allowed | Double |  |  | Allowed hours details |

Table 5.10

Table used:subscription\_plan,user\_registration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table name: user\_Subscription primary key:User\_subscription** | | | | |
| **Description : details of user subscription** | | | | |
| **Field** | **Datatype** | **Size** | **Constraints** | **Description** |
| User\_subscription\_id | Integer | 11 | Primary key | Id of user subscription |
| Plan\_id | Integer | 11 | Foreign key | Id of plan |
| Status | Varchar | 50 |  | Status subscription |
| Discount | Double |  |  | Discount available |
| User\_id | Integer | 11 | Foreign key | Id of user |

Table 5.11

Table used:user\_bill, user\_registration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table name :bill\_details primary key: Bill\_details\_id** | | | | |
| **Description: details of bill** | | | | |
| **Field** | **Datatype** | **Size** | **Constraints** | **Description** |
| Bill\_details\_id | Integer | 11 | Primary key | Id of bill details |
| Bill\_id | Integer | 11 | Foreign key | Id of bill |
| User\_sub\_id | Integer | 11 | Foreign key | Id of user |
| Amount | Double |  |  | Total amount |
| Discount | Double |  |  | Discount available |
| Total | Double |  |  | Total sum up |

Table 5.12

Table used: staff\_login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table name:staff\_attendance primary key:Staff\_attendance\_id** | | | | |
| **Description: details of staff attendance** | | | | |
| **Field** | **Datatype** | **Size** | **Constraints** | **Description** |
| Staff\_attendance\_id | Integer | 11 | Primary key | Id of staff attendance |
| Staff\_id | Integer | 11 | Foreign key | Id of staff |
| Log\_in time | Datetime |  |  | Login time of staff |
| Log\_out time | Datetime |  |  | Logout time of stff |
| Total hours | Double |  |  | Total hours worked |
| Salary | Double |  |  | Salary of staff |

Table 5.13

Table used :user\_registration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table name:user\_attendance primary key: User\_attendance** | | | | |
| **Description: details of user attendance** | | | | |
| **Field** | **Datatype** | **Size** | **Constraints** | **Description** |
| User\_attendance | Integer | 11 | Primary key | Details of user attendance |
| User\_id | Integer | 11 | Foreign key | Id of user |
| Log\_in | Datetime |  |  | Login details of user |
| Log\_in | Datetime |  |  | Logout details of user |
| Total\_time | Double |  |  | Total time user used |
| Part | Varchar | 15 |  | Part of user |

**5.2 Input Design**

Input designing is the basic theory to be considered during system study. The input media used in the system is the keyboard. Details are entered in the system through different data entry screens. The system is designed in a user-friendly manner. Appropriate error messages are displayed when a false data is entered. Design of the system is web-oriented and is highly interactive to the users. The user interface design is very important for any application. The interface design defines how the software communicates within itself, to system that interpreted with it and with human who use it. The interface design is very good. The user will fall into an interactive software application.

The input design is the process of converting the user-oriented description of inputs into a programmer-oriented specification. The objective of input design is to create an input layout that is easy to follow and prevents the user from committing errors. It covers all phases of input, right from the creation of initial databases to the actual data entry into the system. The input design is the link that ties the system into the world of its users. Hence, lays its importance in the design phase. The input design makes sure that while entering data, the end-users understand the format in which the data is to be entered so that it is accepted by the system, the data values that are mandatory for the system to function, the order in which transactions need to be processed etc.

The goal designing input data is to make the automation as easy and free from errors as possible. Input design, involves determining the record media, method of input, speed of capture and entry to the system. The main objectives that are guiding as in the input stages are:

• Controlling the amount of inputs

• Avoiding inordinate delay

• Controlling errors

• Avoiding extra steps

• Keeping the process simple

• To achieve highest level accuracy.

**5.3 Output Design**

Output refers to the results and information that are generated by the system. Here determine information to be present, decide layout and select output medium, arrange presentation of information in accepted format of column headings and pagination are specified. Output design plays a major role in providing the user with the required format. The major function of the output is to convey information and so its layout and design are careful considerations. Information must be carefully considered to the needs of the user. Standards for printed output suggest giving each output a name or title, providing a sample of the output layout, and specifying the procedure for providing the accuracy of the output data.

The output devices to consider depends on the compatibility of the devices with the system, response time requirement and printed quality required. 27 The design output form, attention is given to proper identification and wording, readability and use, composition and layout, order of data items and clarity of instructions. A well-designed form with clarity stated captions should be self-instructing. An organizations form must be centrally controlled for efficient handling. Computer output is the most important and direct information source to the user. Output design is a process that involves designing necessary outputs in the form of reports that should be given to the users according to the requirements. Efficient, intelligible output design should improve the systems relationship with the user and help in decision making. Since the reports are directing referred by the management for taking decisions and to draw conclusions they must be designed with almost care and the details in the reports must be simple, descriptive and clear to the user/ so while designing output the following things are to be considered.

* Determine what information to present.
* Arrange the presentation of information in an acceptable format.
* Decide how to distribute the output to intend receipts.
* Depending on the nature and future use of output required, they can be displayed on the monitor for immediate need and for obtaining the hardcopy.
* Efficient and intelligent output design should improve system relation with the user and help in decision making that is, this makes system user friendly to be displayed or printed as per the user’s choice.

**5.4 Program Design**

**i.** **Admin**

Step 1: Start

Step 2: Once logged in the admin has the privilege to staff management, user management, equipment management and billing.

Step 3: In staff management admin can add new staff, edit staff, view staff attendance and View staff salary report.

Step 4: In user management admin can add user, approve new user, update user subscription and view user attendance.

Step 5: In equipment management admin can add equipment, update equipment, location of equipment, delete equipment.

Step 6: In billing admin can calculate bill and auto generate bill.

Step 7: Stop

**ii. Virtual help desk**

Step 1: Start

Step 2: Virtual help desk can convert voice to text and vice versa.

Step 3: Virtual help desk can manage shifting option and availability of trainer.

Step 4: Virtual help desk can manage new user registration, schedule/reschedule of classes.

Step 5: Virtual help desk can view payment history.

Step 6: In virtual help desk, admin can train the bot, add static and dynamic question, can set language preferences, block/unblock users.

Step 7: Stop.

**iii. Attendance tracking**

Step 1: Start

Step 2: Attendance tracking can fetch attendance from biometric and mark log.

Step 3: Attendance tracking can also fetch attendance using face recognition using openCV.

Step 4: Can combine all attendance logs and automatically prepare attendance report.

Step 5: Attendance tracking can manage auto billing based on attendance.

Step 6: Attendance tracking can send report or invoices to the members.

Step 7: Stop

# CHAPTER 6

## **FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS**

**6.1 FUNCTIONAL REQUIREMENTS**

A functional requirement document defines the functionality of a system or one of its subsystems. It also depends upon the type of software, expected users and the type of system where the software is used. Functional user requirements may be high-level statements of what the system should do but functional system requirements should also describe clearly about the system services in detail. A function is nothing but inputs, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. Functional software requirements help you to capture the intended behavior of the system. This behavior may be expressed as functions, services or tasks or which system is required to perform. Functional

Requirements should include the following things:

* + Trainers should with his/her credentials provided by admin for doing any activity.
  + Each member should register to this system for auto identification.
  + If blocked users are trying to login, it should be denied.
  + Equipment’s of the health club should be listed and mark location of that equipment.
  + It should clearly define who will be allowed to create/modify/delete the data in the system
  + Face should be clearly visible (no mask) in the system for attendance marking.

**6.2 NON-FUNCTIONAL REQUIREMENTS**

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. Non-functional requirements are “system shall be requirement ". Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes”, “quality goals", "quality of service requirements" and "non-behavioural requirements. Some of the non-functional requirements are mentioned below i. Usability: The system shall have a clean interface with only needed features, clear terminology and tool tips wherever necessary. Warnings or alerts shall be specified in clear way. ii. Efficiency: The system shall respond to different searches being conducted like searching particular equipment, search free slot, etc. in a very fast way. iii. Portability: The system shall be independent of the specific technological platform used to implement it. iv. Reliability: Reliability defined as a measure of the time between failures occurring in a system (measure show frequently the system fails), so that the system shall operate without any failure for a particular period of time v. Availability: Availability measures the percentage of time the system is in its operational state so that the system shall be available for use 24 hours per day and 365days per year. iv. Security requirements: trainers should be authorized by admin

# CHAPTER 7

# TESTING

Software testing is critical element of software quality assurance and represents the ultimate review of specifications, design and code generation. System testing is the stage of implementation, it is aimed for ensuring that the system works accurately and efficiently before live operations commences. Testing is a purpose of executing a programme with intend of finding errors.

1. Preparing a test case that has high probability of finding undiscovered errors.
2. Testing to erase out all kinds of bugs from the program. Before going for testing, first we have to decide the type of test. For this impact system, unit testing is carried out. And the following things are taken to consideration. To ensure that information properly places in and out of the program.

**7.1 Testing Strategies**

An engineered product can be tested in one of these two ways. These testing strategies include:

* Black box testing
* White box testing

**White box testing**

White-box testing is a method of testing the application at the level of the source code. White-box testing (also known as clear box testing, glass box testing, and transparent box testing, and structural testing) is a method of testing [software](https://en.wikipedia.org/wiki/Software) that tests internal structures or workings of an application, as opposed to its functionality. In white-box testing an internal perspective of the system, as well as programming skills, are chooses inputs to exercise paths through the code and determine the expected outputs.

**Black box testing**

Black-box testing is a method of [software testing](https://en.wikipedia.org/wiki/Software_testing) that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: [unit,](https://en.wikipedia.org/wiki/Unit_test) [integration](https://en.wikipedia.org/wiki/Integration_testing)[, system](https://en.wikipedia.org/wiki/System_testing) and [acceptance.](https://en.wikipedia.org/wiki/Acceptance_test) It is sometimes referred to as specification-based testing.

**7.2 Unit Testing**

In this testing we test each module individually and integrate the overall system. Unit testing focuses verification efforts on the smaller unit of software design in the module. This is also known as module testing. The modules of the system are tested separately. The testing is carried out during programming stage itself. In these testing steps each module is found to work satisfactorily as regarding to the expected output given by the user. It is very easy to find errors and debug the system. In this project, after coding each module have to be individually tested to determine whether they are coded correctly so that they satisfy the requirements in the specifications and execute effectively as individual units was tested and run individually. . In this project unit testing can be applied to check each functionality of the health club Management system.

**7.3 Integration Testing**

Data can be lost across and interface; one module can have an adverse effect on other sub functions when combined, may not produce the desired major functions. Integrated testing is the systematic testing for constructing the uncovered errors within the interface. This testing was done with sample data. The need for integrated test is to find the overall system performance. According to this project, using integrated test plan prepared in the design phase of system developed as a guide, the integration test was carried out.

**7.4 Validation Testing**

Validation testing can be defined in many ways but a simple definition is the validation succeeds when software function in manner that can be reasonably accepted by the customer. After validation test have been conducted one of the two possible conditions exists: The function or performance characteristics confirmed to specification and accepted. A deviation from specification is uncovered and deficiency list is created. In this project, validation testing is done with all forms. All forms are validated completely. Name should contain only alphabets. In case of contact numbers and other numbers should contain only numeric values. The validation checking also includes boundary checking for variable fields. The “Integrated health club Management system” to validate the username and password of login page.

**7.5 System Testing**

Software testing is critical element of software quality assurance and represents the ultimate review of specifications, design and code generation. System testing is the stage of implementation, it is aimed for ensuring that the system works accurately and efficiently before live operations commences. Nothing is complete without testing, as it is vital success of the system.

Testing Objectives:

There are several rules that can serve as testing objectives, they are

* Testing is a process of executing a program with the intent of finding an error

• A good test case is one that has high probability of finding an undiscovered error.

* A successful test is one that uncovers an undiscovered errors.

A test case is a specification of the inputs, execution conditions, testing procedure, and expected results that define a single test to be executed to achieve a particular [software testing](https://en.wikipedia.org/wiki/Software_testing) objective, such as to exercise a particular program path or to verify compliance with a specific requirement. Test cases underlie testing that is methodical rather than haphazard. A battery of test cases can be built to produce the desired coverage of the software being tested. Formally defined test cases allow the same tests to be run repeatedly against successive versions of the software, allowing for used to design test cases. The tester effective and consistent [regression testing.](https://en.wikipedia.org/wiki/Regression_testing)