# CHAPTER 1

**INTRODUCTION**

Speech and textual information play a crucial role in communicating between humans. An article in “The New York Times” published that now-a-days the adults are spending more than 8 hours a day on screens of computers or mobiles. So the major communication between humans is conducted through web applications such as WhatsApp, Facebook, and Twitter etc. as a form of speech and textual conversation. In the present system, I have focused on designing a study application for students for communication to tutor through application namely Chabot in the educational domain. The proposed Chabot assists in answering questions provided by the students. To develop the system, I have employed an ensemble prediction method as linear regression in the presence of extracted features from our prepared dataset. In our current society most of us choose internet as the best teacher. They can find any information from internet easily even though some of them still demand a tutor or person to solve this doubts and query. It has also been the rise of online training centers and organization. Such like organization and sites are widely available today but in this system these have few limitations for serving the user’s doubts with a assistance for such organization few persons have to wait for all the time. It is not feasible because those users came different time and also request manual efforts. As a remedy we will proved a virtual trainer who isn`t an actual trainer it a program that can reply the queries with the help of given set of data. For collecting those data this program under goes training under the real trainers so they don’t want to seat all the time in front of those computer. It will help to reduce the manual effort. Virtual tutor is program that is working under machine learning technique.

**Overview of the Project**

The main purpose of this system is to maintain a better communication between the students and the Teachers. The system provides knowledge transfer between staff and students and at the same time a very fast communication between all the users. The objective of the project is that it would be beneficial to the students in preparing study materials and conducting exams. They can resolve the doubts about a particular subject by contacting with the tutors.

The main purpose of the project “Virtual tutor” is to make available tutors for the students to resolve the doubts about a particular subject by using the system without wait for all the time. The Virtual Tutor system provides instant replies and easy interaction

# CHAPTER 2

# SYSTEM SPECIFICATION

* 1. **HARDWARE SPECIFICATION**

PROCESSOR : INTEL CORE i5/8th

DISPLAY: 11” COLORMONITOR MEMORY SIZE: 1.99GB RAM

HARDDISK: 250GB KEYBOARD: 104 STANDARDS CLOCK SPEED : 2.24GHZ

**2.2 SOFTWARE SPECIFICATION**

PLATFORM: WINDOWS 10

FRONT END: PYTHON, DJANGO, VS CODE

BACKEND: SQL SERVER DOCUMENTATION TOOL:MS OFFICE 2016

**2.3 ABOUT DEVELOPING TOOLS**

**ABOUT FRONT END-PYTHON**

**Python**

Python is an interpreter, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding; make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python is dynamically typed and garbage-collected. It supports multiple programming Para-diagrams, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library. Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are optional. It has fewer syntactic exceptions and special cases than C or Pascal

.

**SQL Server**

SQL Server Management Studio (SSMS) is an integrated environment for managing any SQL infrastructure. Use SSMS to access, configure, manage, administer, and develop all components of SQL Server, Azure SQL Database, and Azure Synapse Analytics. SSMS provides a single comprehensive utility that combines a broad group of graphical tools with a number of rich script editors to provide access to SQL Server for developers and database administrators of all skill levels.

SQL Server Management Studio (SSMS) is a software application first launched with Microsoft SQL Server 2005 that is used for configuring, managing, and administering all components within Microsoft SQL Server. It's the successor to the Enterprise Manager in SQL 2000 or before. The tool includes both script editors and graphical tools which work with objects and features of the server.

A central feature of SSMS is the Object Explorer, which allows the user to browse, select, and act upon any of the objects within the server. It also shipped a separate Express edition that could be freely downloaded, however recent versions of SSMS are fully capable of connecting to and manage any SQL Server Express instance. Microsoft also incorporated backwards compatibility for older versions of SQL Server thus allowing a newer version of SSMS to connect to older versions of SQL Server instances. It also comes with Microsoft SQL Server Express 2012, or users can download it separately.

Starting from version 11, the application was based on the Visual Studio 2010 shell, using WPF for the user interface. Versions 18 and after are based on the Visual Studio 2017 Isolated Shell.

In June 2015, Microsoft announced their intention to release future versions of SSMS independently of SQL Server database engine release

**Microsoft Visual Studio 2010**

Microsoft Visual Studio 2010 Ultimate provides an integrated environment of tools and server infrastructure that simplifies the entire application development process. Deliver business results using productive, predictable, customizable processes and increase transparency and traceability throughout the lifecycle with detailed analytics. Microsoft Visual Studio 2010 Ultimate - efficient, powerful and smart tool that helps you easily create new solutions or enhancing existing applications unleash your creativity with powerful prototyping etc.

Develop applications based on C++, C# and .NET Framework and deploy them from a seamless environment to the known code repositories. Compile the code automatically and check for code errors with advanced debugging and code analysis. Customize the deployment factor as well as the platform support.

### Django

When you are ready to deploy your Django application to a remote server, you have various options for deployment.

### Simple deployment options

Before fully deploying your application, you can test the process, build configuration, and deployed behavior by using one of these interim techniques.

### Building and serving from disk

During development, you typically use the ng serve command to build, watch, and serve the application from local memory, using [webpack-dev-server.](https://webpack.js.org/guides/development/#webpack-dev-server) When you are ready to deploy, however, you must use the ng build command to build the app and deploy the build artifacts elsewhere.

Both ng build and ng serve clear the output folder before they build the project, but only the ng build command writes the generated build artifacts to the output folder.

The output folder is dist/project-name/ by default. To output to a different folder, change the outputPath in angular.json.

As you near the end of the development process, serving the contents of your output folder from a local web server can give you a better idea of how your application will behave when it is deployed to a remote server. You will need two terminals to get the live-reload experience.

* On the first terminal, run the [ng build command](https://angular.io/cli/build) in *watch* mode to compile the application to the dist folder.

content\_copyng build --watch

Like the ng serve command, this regenerates output files when source files change.

* On the second terminal, install a web server (such as [lite-server](https://github.com/johnpapa/lite-server)), and run it against the output folder. For example:

content\_copylite-server --baseDir="dist/project-name"

The server will automatically reload your browser when new files are output.

CHAPTER -3

PROBLEM DESCRIPTION

## 3.1 PROBLEM STATEMENT

The design phase is the second phase in the system development life cycle. In this phase computer, information system is designed in detail from the system specification generated during the study phase. The principle activities performed during the design phase are allocation of functions, identification of testing requirements, output screen and report design, input design and file design. In design phase, the analyst has the task of developing a detailed design of the system including layouts for all inputs, file and outputs.

In the design phase, the detailed design of the system selected in the study phase is accomplished and user-oriented performance specification is converted into a technical design specifications. The system design is the process of developing specification for a candidate system that meet the criteria established in system analysis. The principle activities performed during design phase include the allocation of function of between computer programs, equipment, manual operations, design of database used by the computer programs, specification of the requirement for input, processing and output and the definition of system and computer program test requirements.

There are many aspects to consider in the design of a piece of software. The importance of each should reflect the goals the software is trying to achieve. Some of these aspects are:

* + - EXTENSIBILITY- New capabilities can be added to the software without major changes to the underlying architecture.
    - ROBUSTNESS- The software is able to operate under stress or tolerate unpredictable for invalid Input.
    - RELIABILITY– The software is able to perform the required function under stated conditions for a specified period.
    - FAULT TOLERANCE– The software is resistant to and able to recover from component failure.
    - SECURITY– The software is able to withstand hostile acts and influences.
    - MAINTAINABILITY– The software can be restored to a specified condition within a specified period Of time.

COMPATIBILITY- The software is able to operate with the other products that are designed for interoperability with other product.

In this website, we use Data flow diagram, E-R diagram, Sequence diagram, and Schema Design. A DFD describes flow of the system. Sequence Diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. Schema diagram represents the elements of a system using abstract and graphic symbols: E- R diagram represents system by entity, attribute and its relationship.

3.2 PROBLREM SOLUTION

## 3.2 MODULE AND ITS DESCRIPTION

The system consists of following modules:

1. Administrator Module
2. Tutor Module
3. Student Module
4. **Admin**: Admin is the main user of the system .Admin manages all the approval of the tutor and the students and also has the login system with the user name and password and manages the complaints
5. **Tutor**: Tutor is the module of the virtual tutor .The tutor uploads the videos, notes regarding their particular subject and they train the bot. And they also have the privacy management
6. **Student**: Student is the end user of the system. They can see all their notes and videos regarding their subjects’ .Student can complaint regarding their subjects and they can chat with the bot to clear the doubts and queries .And it also has the privacy system that is login that access through password and username

CHAPTER 4

SYSTEM ANALYSIS

System Analysis is the process of gathering and interpreting facts, diagnosing the problems and using the information to recommended improvements on the system. System analysis is a problem solving activity that requires intensive communication system user and system developer. In system analysis the system is studied to the fundamental detail and analysed. The system analyst plays the role of an interrogator and dills deep into the working of the present system. The system is viewed as a whole and the inputs to the system are identified. The outputs from the organization are traced through various phases of the processing of inputs.

A detailed study of these processes must be made by the various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. The system is called existing system. The existing system is subjected to close study and the problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties. The solutions are given as proposals, which is the proposed system. The proposal is then compared to the existing system. The proposal is presented to the user for an endorsement. The proposal is viewed on user request and suitable changes are made. This loop then ends as soon as the user is satisfied with the proposal.

4.1 EXISTING SYSTEM

Now days the Companies need to spent more money and human resource to promote marketing of their products or items. Referrals have to invest their money first and sometimes the referred persons may not wish to buy the product. Companies get their income from newcomers; hence there are no actual sales. The strategy can produce tentative results. There are no rights set for the different users accessing each document. There is no option in the existing system to provide security for the data being transferred from one user to another.

DRAWBACKS:

Schedule stiffness: There’s no flexibility to create a personal calendar in order to combine studies and peHigher costs not only of money but also of time: as previously mentioned, commuting demands an economic expense. And if the classroom is far, time is also a factor to take into account. You have to decide whether the trip is worth it or not.

* Loss of individualization in a large group of students: it is hard for a teacher to have a customized study plan fit to the needs of each individual in the group.
* Less and more expensive didactic material available: traditional education is more limited when it comes to educational formats. Sometimes pictures, videos or forums are not accessible 24/7. This feature might limit the learning and hinder a good study plan.

## 4.2 FEASIBILITY STUDY

Feasibility is an important phase in the software development process it enables the developers 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 required for implementing it.

* + - The system can be implemented within the given cost and schedule constrains
    - The system contribute to the overall objectives

## 4.2.1 OPERATIONAL FEASIBILITY

Proposed systems are beneficial only if they can be turned into information systems. That is it will meet the organizations operating requirements and also checks that whether the system will work when it is developed and installed.

The web application SesPHR supports the operational feasibility to a great extends. The performance of this software is more accurate, more user friendly, effective, error free.

## 4.2.2 TECHNICAL FEASIBILITY

This is related to the technicality of the project. This evaluation determines whether the technology needed for the proposed system is available or not. It deals with hardware as well as software requirements. That is, type of hardware, software and the methods required for running the systems are analyzed. This involves financial consideration to accommodate technical enhancement. If the budget is a serious constraint, then the project is judged not feasible.

The web application SesPHR supports the technical feasibility to a great extends. That is, this software can be operated with the minimum technical support. It uses Python as front end, MySQL Server as database at windows platform and Mozilla Firefox and Google Chrome as browser. And also it provides accuracy, reliability, ease of access and data security.

## 4.2.3 ECONOMIC FEASIBILITY

## Economical analysis is most frequently used method for evaluating the effectiveness of a candidate system. If benefits outweigh costs, then the decision is made to design and implement the system. Otherwise further justifications or alternatives in the proposed system will have to be made if it is to have a chance of being approved. Economic analysis is the most frequently used method for evaluating the effectiveness of the proposed system. Considering the benefits and savings that are expected from the proposed system, the administrators decided to design and implement a new system

## 4.3 PROPOSED SYSTEM

This proposed system “Virtual Tutor” is an efficient way to solve the student’ doubt and query. During snowstorms and thunderstorms, colleges may cancel classes to avoid putting commuting students at risk of dangerous driving conditions. Rather than miss important class sessions, students in online courses can always "attend" by participating in discussion boards or chat sessions, turning in their work on time, and watching lectures or reading materials. Many students also find substantial savings on fuel costs with no commute for classes.

FEATURES

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations.

* More comfortable learning environment
* Easy to operate
* Lower cost
* Convenience and flexibility

More interaction and greater ability to concentrate

.

# CHAPTER - 5

# SYSTEM DESIGN

## 5.1 INFRASTRUCTURE DESIGN

The design phase is the second phase in the system development life cycle. In this phase computer, information system is designed in detail from the system specification generated during the study phase. The principle activities performed during the design phase are allocation of functions, identification of testing requirements, output screen and report design, input design and file design. In design phase, the analyst has the task of developing a detailed design of the system including layouts for all inputs, file and outputs.

In the design phase, the detailed design of the system selected in the study phase is accomplished and user-oriented performance specification is converted into a technical design specifications. The system design is the process of developing specification for a candidate system that meet the criteria established in system analysis. The principle activities performed during design phase include the allocation of function of between computer programs, equipment, manual operations, design of database used by the computer programs, specification of the requirement for input, processing and output and the definition of system and computer program test requirements.

There are many aspects to consider in the design of a piece of software. The importance of each should reflect the goals the software is trying to achieve. Some of these aspects are:

* + - EXTENSIBILITY- New capabilities can be added to the software without major changes to the underlying architecture.
    - ROBUSTNESS- The software is able to operate under stress or tolerate unpredictable for invalid Input.
    - RELIABILITY– The software is able to perform the required function under stated conditions for a specified period.
    - FAULT TOLERANCE– The software is resistant to and able to recover from component failure.
    - SECURITY– The software is able to withstand hostile acts and influences.
    - MAINTAINABILITY– The software can be restored to a specified condition within a specified period Of time.
    - COMPATIBILITY- The software is able to operate with the other products that are designed for interoperability with other product.

In this website, we use Data flow diagram, E-R diagram, Sequence diagram, and Schema Design. A DFD describes flow of the system. Sequence Diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. Schema diagram represents the elements of a system using abstract and graphic symbols: E- R diagram represents system by entity, attribute and its relationship.

## 5.2 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

Input design is the process of converting a user- oriented description of them input into a computer based system. This design is important to avoid errors in the data input process show the correct direction to the management for getting correct information from the computerized system. It is achieved by creating user friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides records viewing facilities.

When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate management are provided as when needed so that uses will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

Add: For adding all the specified details.

Search: For searching the specified details.

Edit: For editing the specified details.

Delete: For deleting the specified details

## Input Objectives

Controlling Amount of Input: Wherever user input is required, giving possible input values as default in that area reduces the amount of user keystrokes. Thus the user can pass on to next data without much typing. This makes the data entry much fast and error free. When the user has the format of input to be given, it will be very easy for the user to give input in the same format.

**Avoiding Delay**: A processing delay resulting from data entry operations is called a bottleneck. Such bottlenecks are made obsolete in this project by breaking up the amount of data to be entered in each form into different smaller and simpler forms.

**Avoiding Errors in Data:** The rate at which errors occur depends on the quantity of the data. As told in the above objective these errors are reduced by making the number of data to be entered in each form is reduced.

**Avoiding Extra Steps:** To fulfill any operation the user have no need to do complex steps, instead any operation can be done with simple easy to use steps.

## 5.3 OUTPUT DESIGN

Output design is one of the most important features of the information system. When the output is not of good quality, the users will be averse to use the newly designed system and may not use the system. There are many types of outputs, all of which can be either highly useful or can be critical to the users, depending on the manner and degree to which they are used.

Outputs from computer system are required primarily to communicate the results of processing to users. They are also used to provide a permanent hard copy of the results for later consultation.

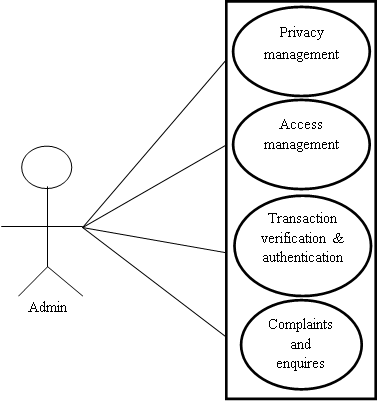
## Output Objectives

The output from an information system should accomplish one or more of the following objectives:

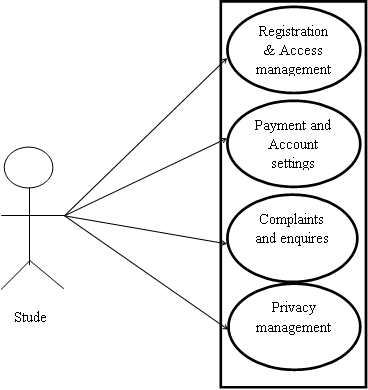
* Convey information about past activities, current status, or projections of the future.
* Signal important events, opportunities, problems or warnings
* Trigger an action
* Confirm an action

5.2 USECASE DIAGRAM

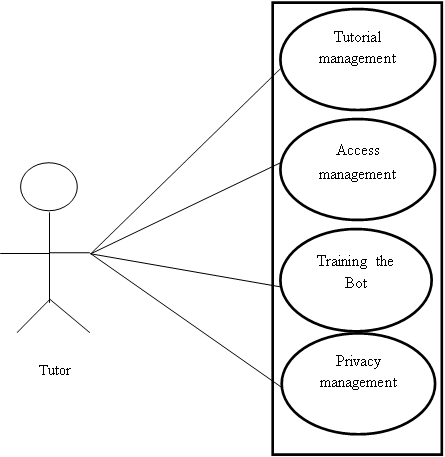
**Use case diagram for Admin**



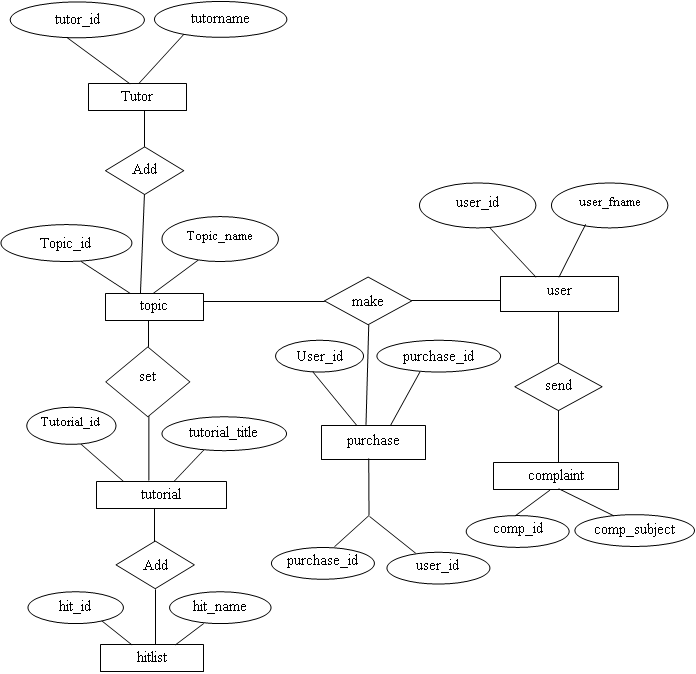
**Use case diagram for Student**



**Use case Diagram for Tutor**



ER DIAGRAM



**5.4 TABLE DESIGN**

**Table Name:** accesshist

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| ***access\_id*** | int(11) | Not Null | ***Access Id*** |
| topic\_id\_id | int(11) | Primary Key | Topic Id I |
| tutorial\_id | int(11) | Primary Key | Tutorial Id |
| user\_id | int(11) | Primary Key | User Id |
| date | varchar(25) | Not Null | Date |

**Table Name:** bank

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| ***bank\_id*** | int(11) | Not Null | ***Bank Id*** |
| bank\_name | varchar(100) | Not Null | Bank Name |
| bank\_holdername | varchar(100) | Not Null | Bank Holdername |
| bank\_cardno | varchar(100) | Not Null | Bank Cardno |
| bank\_cardtype | varchar(100) | Not Null | Bank Cardtype |
| bank\_cvv | varchar(100) | Not Null | Bank Cvv |
| bank\_expdate | varchar(100) | Not Null | Bank Expdate |
| bank\_balance | varchar(100) | Not Null | Bank Balance |

## Table Name: complaint

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| ***comp\_id*** | int(11) | Not Null | ***Comp Id*** |
| comp\_subject | varchar(200) | Not Null | Comp Subject |
| comp\_msg | varchar(250) | Not Null | Comp Msg |
| comp\_date | date | Not Null | Comp Date |
| comp\_replay | varchar(250) | Not Null | Comp Replay |
| comp\_user\_id | int(11) | Primary Key | Comp User Id |

## Table Name: examhist

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| ***exahist\_id*** | int(11) | Not Null | ***Exahist Id*** |
| term | varchar(100) | Not Null | Term |
| topic\_id\_id | int(11) | Primary Key | Topic Id Id |
| user\_id | int(11) | Primary Key | User Id |
| date | varchar(25) | Not Null | Date |
| correct | varchar(25) | Not Null | Correct |
| skiped | varchar(25) | Not Null | Skiped |
| wrong | varchar(25) | Not Null | Wrong |

## Table Name: hitlist

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| ***hit\_id*** | int(11) | Not Null | ***Hit Id*** |
| hit\_name | varchar(100) | Not Null | Hit Name |
| hit\_descr | varchar(4000) | Not Null | Hit Descr |
| tutorial\_id | int(11) | Primary Key | Tutorial Id |

## Table Name: login

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| ***log\_id*** | int(11) | Not Null | ***Log Id*** |
| username | varchar(100) | Not Null | Username |
| password | varchar(100) | Not Null | Password |
| role | varchar(100) | Not Null | Role |

## Table Name: purchase

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| ***purchase\_id*** | int(11) | Not Null | ***Purchase\_Id*** |
| purchase\_amt | varchar(100) | Not Null | Purchase Amt |
| purchase\_date | date | Not Null | Purchase Date |
| purchase\_status | varchar(100) | Not Null | Purchase Status |
| topic\_id\_id | int(11) | Primary Key | Topic Id Id |
| user\_id\_id | int(11) | Primary Key | User Id Id |

## Table Name: question

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| ***quest\_id*** | int(11) | Not Null | ***Quest Id*** |
| quest\_term | varchar(100) | Not Null | Quest Term |
| quest\_ques | varchar(250) | Not Null | Quest Ques |
| quest\_op1 | varchar(250) | Not Null | Quest Op1 |
| quest\_op2 | varchar(250) | Not Null | Quest Op2 |
| quest\_op3 | varchar(250) | Not Null | Quest Op3 |
| quest\_op4 | varchar(250) | Not Null | Quest Op4 |
| quest\_ans | varchar(250) | Not Null | Quest Ans |
| topic\_id\_id | int(11) | Primary Key | Topic Id Id |

## Table Name: topic

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| ***topic\_id*** | int(11) | Not Null | ***Topic Id*** |
| topic\_name | varchar(100) | Not Null | Topic Name |
| topic\_details | varchar(150) | Not Null | Topic Details |
| tutor\_id\_id | int(11) | Primary Key | Tutor Id Id |
| topic\_cover | varchar(1000) | Not Null | Topic Cover |
| topic\_package | varchar(100) | Not Null | Topic Package |

## Table Name: tutorial

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| ***tutorial\_id*** | int(11) | Not Null | ***Tutorial Id*** |
| tutorial\_title | varchar(100) | Not Null | Tutorial Title |
| tutorial\_description | varchar(150) | Not Null | Tutorial Description |
| tutorial\_cover | varchar(1000) | Not Null | Tutorial Cover |
| tutorial\_video | varchar(1000) | Not Null | Tutorial Video |
| topic\_id\_id | int(11) | Primary Key | Topic Id Id |
| tutorial\_order | varchar(100) | Not Null | Tutorial Order |

## Table Name: tutors

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| ***tutor\_id*** | int(11) | Not Null | ***Tutor Id*** |
| tutor\_name | varchar(100) | Not Null | Tutor Name |
| tutor\_gender | varchar(100) | Not Null | Tutor Gender |
| tutor\_address | varchar(500) | Not Null | Tutor Address |
| tutor\_email | varchar(200) | Not Null | Tutor Email |
| tutor\_phone | varchar(100) | Not Null | Tutor Phone |
| tutor\_qualification | varchar(200) | Not Null | Tutor Qualification |
| tutor\_designation | varchar(100) | Not Null | Tutor Designation |
| tutor\_photo | varchar(1000) | Not Null | Tutor Photo |
| tutor\_status | varchar(50) | Not Null | Tutor Status |
| tutor\_logid\_id | int(11) | Primary Key | Tutor Logid Id |

## Table Name: user

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| ***user\_id*** | int(11) | Not Null | ***User Id*** |
| user\_fname | varchar(100) | Not Null | User Fname |
| user\_lname | varchar(100) | Not Null | User Lname |
| user\_paddress | varchar(100) | Not Null | User Paddress |
| user\_caddress | varchar(100) | Not Null | User Caddress |
| user\_email | varchar(100) | Not Null | User Email |
| user\_phone | varchar(200) | Not Null | User Phone |
| user\_status | varchar(50) | Not Null | User Status |
| user\_logid\_id | int(11) | Primary Key | User Logid\_Id |

5.5 DATA FLOW DIAGRAM(DFD)

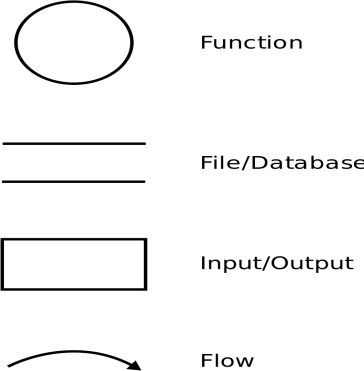
Data Flow Diagram (DFD) representing a system at any level of detail with a graphic network of symbols showing data flows, data stores, data processes, and data sources. The purpose of DFD is to provide a semantic bridge between users and system developers. The diagram is the basis of structured system analysis.

A level 0 DFD, also called a fundamental system model or a context model represents the entire software elements as a single bubble with input and output indicated by incoming and outgoing arrows respectively. Additional process and information flow parts are represented in the next level i.e., Level 1 DFD. Each of the processes represented at Level 1 are sub functions of overall system depicted in the context model. Any processes, which are complex in Level 1, will be further represented into sub functions in the next level, i.e., in level 2.

Data flow diagrams illustrate how data is processed by a system in terms of inputs, and outputs. Represent major components or functions with Circles. Actions for input by a user or a system go in Rectangular Boxes. Databases are represented by Parallel lines enclosing a phrase corner.

# DFD Components

DFD consists of processes, flows, warehouses, and terminators. There are several ways to view these DFD components



### LEVEL 0

Caption:-LEVEL 0 DFD used to graphically represent the flow of data in VIRTUAL TUTOR.

Admin

Tutor

Student

Admin

Tutor

Student

FIGURE 1: DFD DIAGRAM

### LEVEL-1 DFD FOR MODULE ADMIN

Caption:-LEVEL 1 DFD used to graphically represent the flow of data in Admin Module.

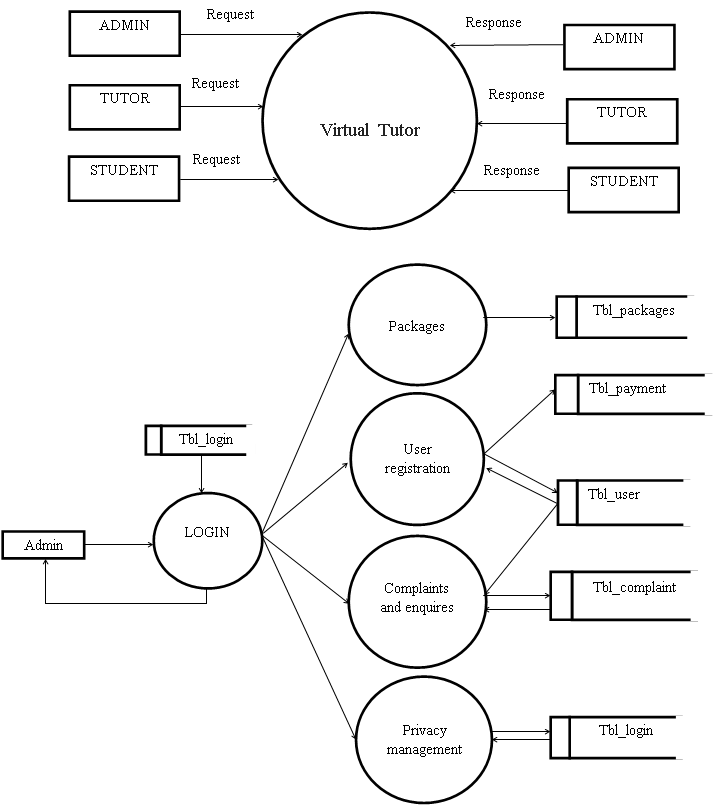


FIGURE 2: DFD ADMIN DIAGRAM

### DFD FOR MODULE TUTOR

Caption: - DFD used to graphically represent the flow of data in Tutor Module.

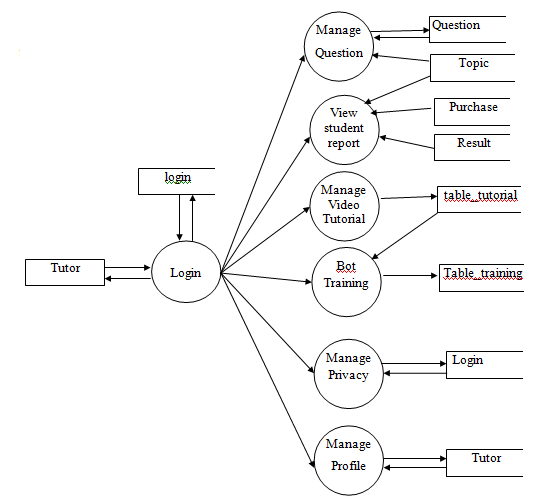
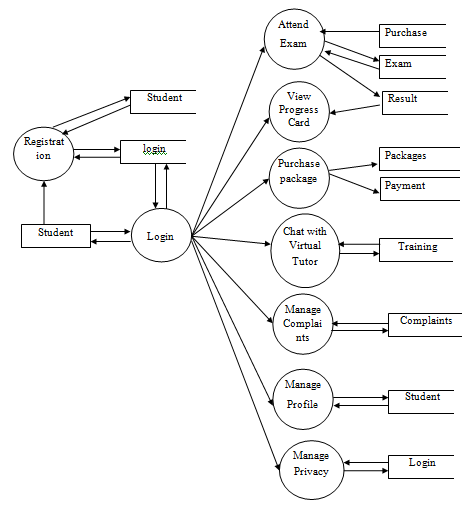


FIGURE 3 : DFD TUTOR DIAGRAM

### DFD FOR MODULE STUDENT

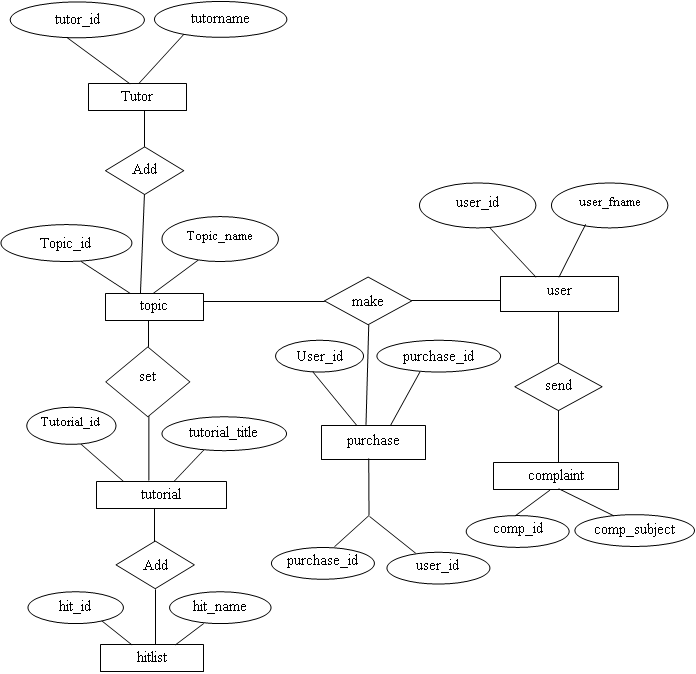
Caption: - DFD used to graphically represent the flow of data in Student Module.



5.8 E-R DIAGRAM

### 3.5.1 ENTITY RELATIONSHIP DAIGRAM FOR VIRTUAL TUTOR

Caption:-E-R Diagram is used for shows the relationship of entity sets stored in database of VIRTUAL TUTOR.



**CHAPTER 6**

**SYSTEM TESTING**

**SOFTWARE TESTING**

Software testing accounts for the largest percentage of technical effort in the software process. The objective of the software testing is to uncover errors. To fulfill this objective, a series of test steps unit, integration, validation and system tests are planned and executed.

.

**6.1 TESTING METHODS**

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

**7.1 UNIT TESTING**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**7.2 Integration Testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**7.1.1 Black Box testing**

Knowing the specified function that a product has been designed to perform, test can be conducted that demonstrates each function that is fully operational, at the same time searching for errors in each function. Black Box testing focuses on functional requirement of the software. Black Box testing attempts to find out errors in the following categories:

* Incorrect or missing functions
* Interface errors
* Errors in data structures or external database access.
* Performance and errors
* Initialization and termination errors

**7.1.2 White Box Testing**

Knowing the internal working of a product test can be conducted to ensure that all gears mesh‖ that is internal operation performs according to specification and all internal components have been adequately exercised. Using white box testing methods, the software engineer can derive test cases that

* Guarantee that all independent paths within a module have been exercised at least once basis path testing.
* Exercise all logical decisions on their true and false sides- Condition testing.
* Execute all loops at their boundaries and within their operation bounds- Loop testing.
* Exercise internal data structures to assure their validity-data flow testing.

**6.2 TEST CASES**

A test case is a document that describes an input, action or event and an expected response, to determine if a feature of an application is working correctly. A test case should contain particulars such as test case identifier, test case name, objective, test conditions, input data requirements steps, and expected results.

**Test case**

Project Title : Searching Trajectories by Regions of Interest(TSR)

Software Tool : JAVA

Test objective : To check whether the entered User name and Password are valid or invalid.

Test data : Username = admin and password = admin

**CHAPTER 7**

**SYSTEM IMPLEMENTATION**

System implementation is the final phase that is putting the utility into action. Implementation is the state in the project where theoretical design turned into working system. The most crucial stage is achieving a new successful system and giving confidence in new system that it will work efficiently and effectively. The system is implemented only after thorough checking is done and it is found working according to the specifications.

Implementation is the state in the project where theoretical design turned into working system. The implementation stage is a system project in its own right. It involves careful planning, design, investigation of the current system and constraints on implementation, design of methods to achieve change over and evolution method. Once the planning has been completed the major effort is to ensure that the programs in the system are working properly. At the same time concentrate on training user staff.

**7.2 EQUIPMENT ACQUISITION**

Equipment vendors can provide specifications for equipment installation. They usually work with projects equipment installation team is planning for adequate space, power and light, and a suitable environment. After a suitable site has been completed, the computer equipment can be installed

**7.3 TRAINING**

The purpose of training is to ensure that all the personnel who are to be associated with the computer based system possesses necessary knowledge skills.

**7.4 EVALUTION**

Many teachers would like to give better, more detailed feedback to students on their writing assignments but realize that there are not enough hours in the day to do so..

**7.5 OPERATIONAL EVALUTION**

The importance of the indicators presented ensures the success of the operational phase of the management system. Therefore, once the operation of the system is guaranteed,

**7.6 ORGANIZATIONAL IMPACT**

Identification and measurement of benefits to the organiszation in such areas financial concerns,operational effiency and competitive impact.

**7.7 USER MANAGEMENT ASSESMENT**

Evaluation of attitude of senior and user managers within the organization,as well as end users.

**7.8 DEVELOPMENT PERFORMANCE**

Evaluation of the development process as overall development time and effort,conformance to budgets and standards and other project management citeria.Includes assessment of development methods and tools.

**7.9 DOCUMENTATION**

After the testing and implementation was completed,the whole system was documented and presented in a readable manner.This was to ensure that when any corrections,manipulations or updating are performed in future,the users would face no problem in performing those changes.Documentation of the sourcecode,the tables that tables that were used to construct the base for system,the framework which bound the programs.

**CHAPTER 8**

**SOFTWARE MAINTENANCE**

Maintenance is the most costly process in the development of software. It is necessary to eliminate the errors in the system during its working life and to tune the system to tune the system to any variation in its working environment. The key software maintenance issues are both managerial and technical. The key management issues are: alignment with customer priorities, staffing which organization does maintenance, estimating costs. Key technical issues are: limited understanding, impact analysis, testing, and maintainability measurement.

Best and worst practices in software maintenance because maintenance of aging legacy software is very labor intensive it is quite important to explore the best and most cost effective methods available for dealing with the millions of applications that currently exists.

There are three different types of software maintenance

**8.1 CORRECTIVE MAINTENANCE**

This is concerned with fixing reported errors in the software. Coding errors are relatively cheap to correct; design errors are more expensive as they may involve the rewriting of several program components.

**8.2 ADAPTIVE MAINTENANCE**

Changing the software to some new environment such as a different platform or to execute in a different operating system, the software functionality does not radically change.

**8.3 PERFECTIVE MAINTENANCE**

This involves implementing new functional or non-functional system requirements. These are generated by software customers as their organization or business changes

**8.4PREVENTIVE MAINTENANCE**

Modification of a software product after delivery to detect and correct latent faults in the software product before they become effective faults.

**CHAPTER 9**

**CONCLUSION**

This page is meant for the technology conclusions of the project report. In the former pages, all details about the development of software have been explained. Virtual Tutor is developed as a web application using Python as front end and MY SQL as back end. Every effort has been made to make the system as user friendly as possible. All the activities provide a feeling like an easy walk over to the user who is interacting with the system. Trial run of the system has been made and gave good results.

The new system has overcome most of the limitations of the existing system and works according to the design specification given. The developed systems dispense the problem and meet the needs of by providing reliable and comprehensive information. All the requirements projected by the user have been met by the system.

All the modules are tested separately and put together to from the main system.

The key features of the system are:

* User friendly
* Ease in handling and implementation
* Adaptable to future enhancements

**CHAPTER 10**

**FUTURE ENHANCEMENT**

The proposed system is developed and tested with some amount of sample data, which satisfy all the requirements. through the system the students can effectively manage their time, learn the materials, and complete assignments on their own schedules.

The system is flexible and if any changes can be made with out much difficulty. Further improvements and extensions can be made in the system to make overall work easier.

**CHAPTER 11**

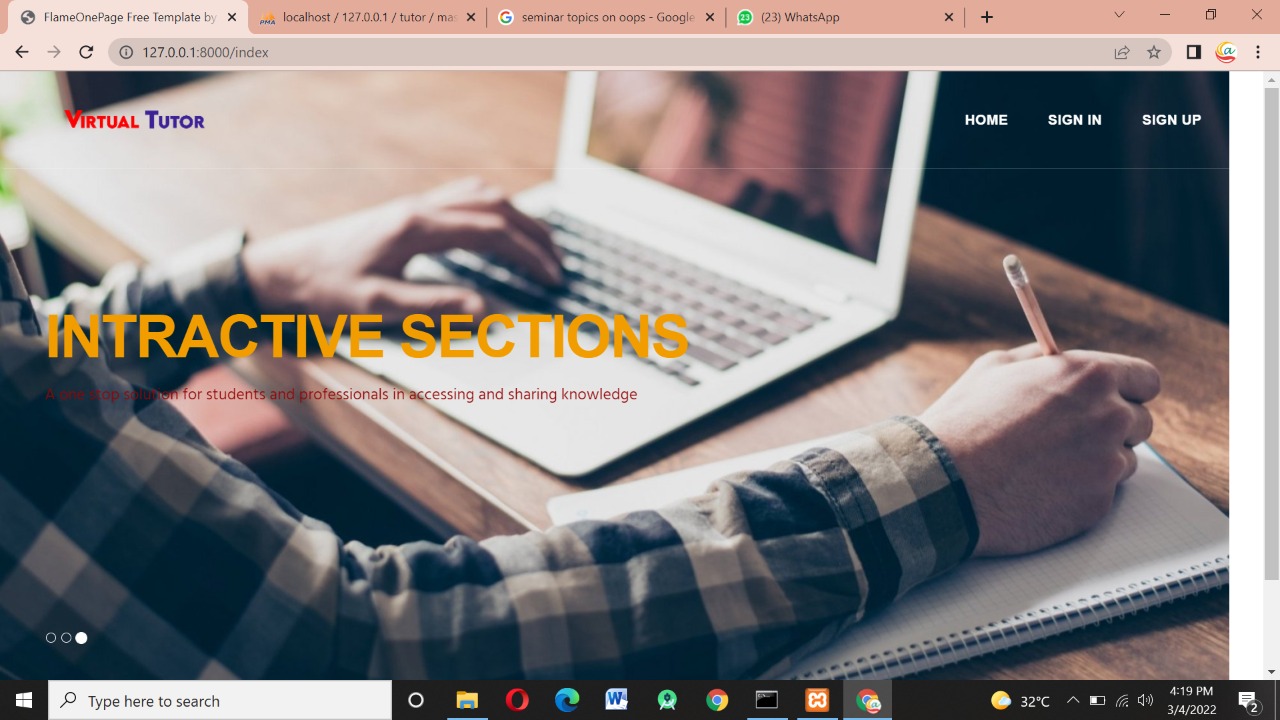
**APPENDIX**

**11.1 SCREEN SHOTS**

## 5.2 SCREENSHOTS

### SCREENSHOT 1:- Home screen

Home screen on the Virtual Tutor Website



CHAPTER 12

REFERENCES

Pankaj Jalote. An integrated Approach to Software Engineering, Third Edition, Narosa Publications. (A case study approach whereby a project is developed by the course of the book , illustrating different activities of Software Development)

**WEBSITES**

* [www.mysql.com](http://www.mysql.com/)
* [www.w3schools.com](http://www.w3schools.com/)
* [www.google.com](http://www.google.com/)
* [www.wikipedia.com](http://www.wikipedia.com/)