**3. DESIGN AND DEVELOPMENT PROCESS**

**3.1 FUNDAMENTAL DESIGN CONCEPTS**

Although the degree of interest in each concept has varied over the year, each has stood the test of time. Each provides the software designer with a foundation from which more sophisticated design methods can be applied. Fundamental design concepts provide the necessary framework for “getting it right”.

During the design process the software requirements model is transformed into design models that describe the details of the data structures, system architecture, interface, and components. Each design product is reviewed for quality before moving to the next phase of software development.

**3.2 DESIGN NOTATIONS**

**3.2.1 DATA FLOW DIAGRAMS**

The data flow diagram is used for classifying system requirements to major transformation that will become programs in system design. This is starting point of the design phase that functionally decomposes the required specifications down to the lower level of details. It consists of a series of bubbles joined together by lines.

* + - Bubbles : Represent the data transformations.
    - Lines : Represents the logic flow of data.

Data can trigger events and can be processed to useful information. System analysis recognizes the central goal of data in organizations. This dataflow analysis tells a great deal about organization objectives are accomplished.

Dataflow analysis studies the use of data in each activity. It documents this finding in DFD’s. Dataflow analysis give the activities of a system from the viewpoint of data where it originates how they are used or hanged or where they go, including the stops along the way from their destination. The components of dataflow strategy span both requirements determination and system’s design. The first part is called dataflow analysis.

As the name suggests, user didn’t use the dataflow analysis tools exclusively for the analysis stage but also in the designing phase with documentation.

**Data Flow Diagram Symbols**

* **Entity**

* **Data Flow**
* **Process**
* **Data Source**

**DESCRIPTION**

**Process:** describes how input data is converted to output data.

**Data Store:** Describes the repositories of data in a system.

**Data Flow:** Describes the data flowing between process, Data stores and external entity.

**Sources:** An external entity causing the origin of data.

**Sink:** An external entity, which consumes the data.

**CONTEXT LEVEL DIAGRAM: (LEVEL 0)**

GENERATES VIEWS

ADMIN

STUDENT

REPORT TAKE TEST

**DATA FLOW DIAGRAM: (LEVEL 1)**

request

ADMIN

response ADMIN DETAILS

request STUDENT DETAIL

response

STUDENT

admin student

adds views

COMPANY PROFILE

adds views

COMPANY PATTERN

adds views

COMPANY QUESTION

prepares attends

MOCK TEST

generates views

REPORT

# 3.3 DESIGN PROCESS

## 3.3.1 TABLE DESIGN

**TABLE NAME:Tbl\_AdminDetails**

**ABBREVIATION: Admin**

**PRIMARY KEY: Var\_Admin\_UserID**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S.NO | ATTRIBUTE | DATATYPE | SIZE | KEY | DESCRIPTION | SAMPLE DATA |
| 1 | Var\_Admin\_UserID | Varchar | 30 | PK | User Name of the admin | admin |
| 2 | Var\_Admin\_Password | Varchar | 10 |  | Password of the admin | admin |

**TABLE NAME: Tbl\_StudentDetails**

**ABBREVIATION: Stud**

**PRIMARY KEY: Var\_Stud\_Username**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S.NO | ATTRIBUTE | DATATYPE | SIZE | KEY | DESCRIPTION | SAMPLEDATA |
| 1 | Var\_Stud\_Username | Varchar | 30 | PK | User Name of the student | 18LMCA09 |
| 2 | Var\_Stud\_Password | Varchar | 10 |  | Password of the student | Madhu18lmca09 |

**TABLE NAME:Tbl\_CompanyProfile**

**ABBREVIATION: CmpyProf**

**PRIMARY KEY: Num\_ CmpyProf \_Cid**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S.NO | ATTRIBUTE | DATATYPE | SIZE | KEY | DESCRIPTION | SAMPLEDATA |
| 1 | Num\_CmpyProf\_Cid | Number | 5 | PK | Company ID | 7869 |
| 2 | Var\_CmpyProf\_CName | Varchar | 40 |  | Company Name | Tata consultancy services |
| 3 | Var\_CmpyProf \_CDesc | Varchar | 100 |  | Company Description | TCS is an Indian multinational company headquartered in Mumbai. |

**TABLE NAME: Tbl\_CompanyPattern**

**ABBREVIATION: CmpnyPatn**

**FOREIGN KEY: Num\_CmpyProf\_Cid**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S.NO | ATTRIBUTE | DATATYPE | SIZE | KEY | DESCRIPTION | SAMPLEDATA |
| 1 | Num\_CmpyProf\_Cid | Number | 5 | FK | Company ID | 7869 |
| 2 | Var\_ CmpnyPatn \_R1 | Varchar | 15 |  | ROUND 1 | Aptitude |
| 3 | Var\_ CmpnyPatn \_R2 | Varchar | 15 |  | ROUND 2 | Technical |
| 4 | Var\_ CmpnyPatn \_R3 | Varchar | 15 |  | ROUND 3 | Group discussion |
| 5 | Var\_CmpnyPatn \_R4 | Varchar | 15 |  | ROUND 4 | HR |

**TABLE NAME:Tbl\_CompanyQuestion**

**ABBREVIATION: CmpnyQues**

**FOREIGN KEY: Num\_CmpyProf\_Cid**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S.NO | ATTRIBUTE | DATATYPE | SIZE | KEY | DESCRIPTION | SAMPLE  DATA |
| 1 | Num\_CmpyProf\_Cid | Number | 5 | FK | Company ID | 7869 |
| 2 | Var\_CmpnyQues\_Questions | Varchar | 75 |  | Questions | 500-200+10\*2? |
| 3 | Var\_CmpnyQues\_choice1 | Varchar | 100 |  | choice1 | 970 |
| 4 | Var\_CmpnyQues\_choice2 | Varchar | 100 |  | choice2 | 480 |
| 5 | Var\_CmpnyQues \_choice3 | Varchar | 100 |  | choice3 | 400 |
| 6 | Var\_CmpnyQues \_choice4 | Varchar | 100 |  | choice4 | 124 |
| 7 | Num\_CmpnyQues \_Coption | Number | 5 |  | Correct answer of the choice | 2 |
| 8 | Var\_CmpnyQues \_Canswer | Varchar | 100 |  | Correct answer of the question | 480 |

**TABLE NAME: Tbl\_MockTest**

**ABBREVIATION: MckTst**

**PRIMARY KEY: Num\_MckTst \_Tid**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S.NO | ATTRIBUTE | DATATYPE | SIZE | KEY | DESCRIPTION | SAMPLEDATA |
| 1 | Num\_ MckTst \_Tid | Number | 5 | PK | Test ID | 12456 |
| 2 | Var\_MckTst\_MQuestions | Varchar | 75 |  | Mock questions | 10+500-100? |
| 3 | Var\_MckTst\_Mchoice1 | Varchar | 100 |  | choice1 | 97 |
| 4 | Var\_MckTst\_Mchoice2 | Varchar | 100 |  | choice2 | 115 |
| 5 | Var\_MckTst\_Mchoice3 | Varchar | 100 |  | choice3 | 110 |
| 6 | Var\_MckTst\_Mchoice4 | Varchar | 100 |  | choice4 | 124 |
| 7 | Num\_MckTst\_Moption | Number | 5 |  | Correct answer of the choice | 3 |
| 8 | Var\_MckTst\_Manswer | Varchar | 100 |  | Correct answer of the question | 110 |

**TABLE NAME: Tbl\_Report**

**ABBREVIATION: Rpt**

**FOREIGN KEY: Num\_CmpyProf\_Cid**

**Num\_ MckTst \_Tid**

**Var\_Stud\_Username**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| S.NO | ATTRIBUTE | DATATYPE | SIZE | KEY | DESCRIPTION | | SAMPLE DATA |
| 1 | Num\_CmpyProf\_Cid | Number | 5 | FK | Company ID | 7869 | |
| 2 | Num\_ MckTst \_Tid | Number | 5 | FK | Test ID | 12456 | |
| 3 | Var\_Stud\_Username | Varchar | 30 | FK | User Name of the student | 18LMCA09 | |
| 4 | Num\_Rpt\_Totalmarks | Number | 2 |  | Total marks | 4 | |

# 3.3.2 INPUT DESIGN

Once the output requirement has been finalized, the next step is to find out what data need to be made available to the system to produce the desired outputs. The basic documents in which these data are available need to be identified. If necessary this documents may have to be revised or new documents may have to be introduced. Input is one of the most expensive phases of the operation of a computerized system and creates some times a major problem.

# 3.3.3 OUTPUT DESIGN

The output design presents the manipulated data to the end user. The output design acts as medium of communication to the user by providing the desired data that may be either a stored data fetched from the database or may be manipulated result displayed to the user for confirmation before it is stored into the database.

# DEVELOPMENT

Top-down approaches emphasize planning and a complete understanding of the system. It is inherent that no coding can begin until a sufficient level of detail has been reached in the design of at least some part of system. Programming, actually writing software code is just one part of the process, which is why people prefer to be called developers rather than merely programmers. Top-down programming is a programming style, the mainstay of traditional procedural languages, in which design begins by specifying complex task and then dividing them into separate modules. Eventually, the components are specific enough to be coded and the program is written.