# ONLINE TUTOR FINDING SYSTEM

***SOFTWARE ENGINEERING PROJECT REPORT***

**Keshav Mahavidyalaya**

**University of Delhi**

Submitted by- Bhavya Batra 5706

Muskan Kapoor 5775

Aatif Nisar 5734

**ACKNOWLEDGEMENT**

On the successful completion of our project ONLINE TUTOR FINDING SYSTEM, we would like to express our sincere gratitude to everyone who helped us in the completion of the project.

We are sincerely thankful to our project guide \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ for her interest, guidance and suggestions throughout the course of the project. We feel honoured and privileged to work under her. She shared her vast pool of knowledge with us that helped us steer through all the difficulties with ease. This project would not have been possible without her guidance.

**Bhavya Batra**

**Muskan Kapoor**

**Aatif Nisar**

**CERTIFICATE**

This is to certify that the project entitled ―ONLINE TUTOR FINDING SYSTEM submitted by Bhavya Batra ,Muskan Kapoor and Aatif Nisar has been carried out under my supervision. The project has been submitted as per the requirements in the fourth semester of BSc. (H) Computer

Science.

**Supervisor**

## ----------------------

**TABLE OF CONTENTS**

1. **Problem statement\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_6**
2. **Process model\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_7**
3. **Software Requirement Specification \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 9**

**3.1.1 Purpose\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_10**

**3.1.2 Scope \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_10**

**3.1.3 Overview \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11**

**3.2. Overall Description \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11**

**3.2.1 Product perspective \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11**

* + 1. **Product functions \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_12**

**3.3User characteristics \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_12**

**3.4 General Constraints \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_13**

**3Specific Requirements \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_13**

**3.5.1External Interface Requirement \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_13**

**3.5.1.1 User Interface\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_13**

**3.5.1.2 Hardware Interface \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_14**

**3.5.2 Software requirement \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_14**

**3.5.3 Performance Requirement \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 15**

* + 1. **Functional Requirement\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 15**

**3.5.5Other Requirements \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 15**

1. **Data Flow Diagram \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_16**

**4.1Level 0 DFD \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_16**

**4.2Level 1 DFD \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 17**

1. **Project Management\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_18**
2. **Scheduling \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_21**

**6.1Timeline chart \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 22**

1. **Design\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_23**
2. **Pseudo code \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_24**
3. **Testing \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_26**

**9.1Flow graph \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 28**

**9.2Cyclomatic complexity \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_30**

1. **Conclusion\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_32**
2. **Bibliography \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_33**

## 1.PROBLEM STATEMENT

Existing System

* Minor but important details about the teacher are missed out.
* Current working systems are not efficient and quick.

Proposed System

The proposed system eliminates the disadvantages of the existing system. The features of the “Online tutor finding system” system are as follows:

* Easy to use
* The user interface is attractive and easy to understand.
* The system provides quick access to the best teachers available.
* Secure payment gateway is provided to the users.
* The teachers’ details are verified and hence are authentic.
* Accurate information about the teachers is provided.
* Only verified teachers are displayed on the website.

## 2.PROCESS MODEL

The software process model used is LINEAR SEQUENTIAL MODEL/WATERFALL MODEL because of the following reasons:

* This is a small software which demands a systematic and sequential approach to software development.
* The project is dedicated to perform only the given set of tasks.
* Pre-requisite requirements of the project are clear.
* The structure of the software is pretty straight-forward with a simple hierarchy and without many complex transfers of control.

Characteristics of the Waterfall Model are:

* It is useful when requirements are well understood and stable.
* Suitable when work flows from communication to deployment in a linear fashion.
* It suggests a systematic and sequential approach to software development.

Communication

Planning

Modeling

Construction

Deployment

**3.SYSTEM**

# REQUIREMENT SPECIFICATIONS (SRS)

## 3.1INTRODUCTION

This is an online web based tuition platform where students get help in finding suitable teachers to study a particular subject. This application provides information regarding qualifications, timings for classes, fee structure and various courses available.

This system considers student’s requirements and lists suitable teachers for tuition purposes. This system also manages a teacher profile where teachers can submit their details.

### 3.1.1 PURPOSE

The objective of this project is as follows:-

* A student login page where they can find qualified teachers online according to their requirements.
* A teacher login page where they can submit their profile to find students to teach.
* Students can select course duration.
* Fees payment is done online.
* Class schedule list is generated.

### 3.1.2SCOPE

This system can be used by students to find teachers, pay fees and select course duration.

It can also be used by teachers to submit their profiles and get students to teach according to their capabilities.

### 3.1.3OVERVIEW

**Online tutor finding system** is a platform where both the users (teachers and students) can connect to fulfil their requirements. Services provided by this platform include online fee payment, class schedule generation and teacher search.

## 3.2OVERALL PERSPECTIVE

The system provides various services to the users. These include finding a suitable teacher, viewing class timings, selecting course duration and making payment.

### 3.2.1 PRODUCT METRICS

This system is aimed for students looking for teachers for tuition and for teachers looking for students to teach.

The proposed system is designed to eliminate the inconvenience faced by students to find home tutors suiting their requirements and also to eliminate teachers’ problems to find students for home tuition.

Online tutor finding system is an online platform for both the users to fulfil their requirements. It also has an online fee payment portal in which students can pay fees for as much classes as they require. This system also generates a class schedule based on timings suitable for both the users.

### 3.2.2PRODUCT FUNCTIONS

Online tutor finding system accepts student’s details like subject, standard, area and then provides a list of available teachers matching the requirements. This system also accepts teachers’ profile and puts them in database after verification.

## 3.3USER CHARACTERISTICS

Students just have to provide correct details .Students must provide a safe and calm environment for classes to take place. Teachers must have proof of qualifications like certificates. Teachers must also have their own vehicle to reach and come back to students’ home.

## 3.4GENERAL CONSTRAINTS

The working of this platform is completely dependent on a working internet connection at both server and client level.

## 3.5SPECIFIC REQUIREMENTS

**3.5.1EXTERNAL INTERFACE**

### REQUIREMENTS

#### 3.5.1.1USER INTERFACE

Client End Interfaces:-

1. Role identification page to differentiate users as students and teachers.
2. Different login pages for both the users to provide them their specific functionalities.
3. Registration page for new users.

#### 3.5.1.2 HARDWARE INTERFACE

At Client end a mobile device/computer is used.

#### 3.5.2 SOFTWARE REQUIREMENTS

Database- MYSQL is used as database as it easy to maintain and retrieve records by simple queries and is used at the server end. Development tools and Programming language- HTML is used to write the whole code and develop web pages with CSS, java script for styling work and PHP language for sever side scripting.

#### 3.5.3PERFORMANCE REQUIREMENTS

Performance of the platform works best with a good internet connection at the database site which feeds list of available teachers and must be backed up regularly to avoid any losses.

#### 3.5.4FUNCTIONAL REQUIREMENTS

Authentic details from teachers are required for verification. Students need to fill the requirements correctly for software to generate an appropriate list. A link with payment gateway is also required for online payment purposes.

**3.5.5 OTHER REQUIREMENTS**

Good knowledge of PHP, SQL, HTML, CSS & JavaScript

## 4.1 DFD LEVEL 0

Stude

ntdetails

Displayteachers

Studentrequirements

Displayclasstimings

Teacherdetails

Disp

layconfirmation

slipafterpayment

Teacherdocuments

**Tutorial Point**

**Confirmation**

**Slip**

**ClassTimings**

**Teacher**

**ListofTeachers**

**Student**

P a g e

### 4.2 DFD LEVEL 1

Page

17

Teac

herDocs

TeacherDetails

Student

Re

q.

StudentDetails

StoreinDatabase

**S**

**TUDENT**

**DATABASE**

FetchfromDatabase

**TEACHER**

**DATABASE**

ListofTeachers

DisplayClassTimingsfortheSelectedTeacher

DisplayPaymentReceipt

**USER**

**TEACHER**

**STUDENT**

**Get Teacher**

**Details**

**Get Student**

**Requirements**

**Get Student**

**Details**

**Get Teacher**

**Documents**

**Match Student**

**Req**

**& Teachers’**

**Details**

**Payment**

**Gateway**

**Selecting a**

**Teacher**

**List of Teachers**

**Confirmation Slip**

**Class Timings**

**4.3DATA DICTIONARY**

|  |  |
| --- | --- |
| USER | [TEACHER OR STUDENT] |
| LOGIN | [USER+USERNAME+EMAIL+PHONE NUMBER +PASSWORD] |
| USERNAME | [FIRST NAME + LAST NAME] |
| EMAIL | [\_\_\_\_\_\_@\_\_.\_] |
| PHONE NUMBER | [DIGIT+ DIGIT+DIGIT+ DIGIT+ DIGIT+ DIGIT+ DIGIT+ DIGIT+ DIGIT+DIGIT] |
| PASSWORD | [ALPHANUMERIC CHARACTERS] |
| SEARCH\_TEACHER | [TEACHER QUALIFICATIONS] |
| TEACHER QUALIFICATIONS | [DEGREE+ARE OF EXPERTISE] |
| DISPLAY | [LIST OF MATCHED TEACHERS] |
| BILL | [STUDENT NAME+ID+TEACHER NAME+CLASS TIMINGS+PHONE NUMBER+FINAL FEES] |
| PAYMENT METHOD | [CARD OR WALLET] |
| CARD DETAILS | [CARD NUMBER+EXPIRY+CVV] |
| WALLET DETAILS | [MOBILE NUMBER+WALLET NAME] |

## 5.PROJECT MANAGEMENT

### FUNCTION POINT ESTIMATION

|  |  |  |  |
| --- | --- | --- | --- |
| **Information Domain Value** | **Count** | **Weighing Factor** | **Weighing Count** |
| External Inputs (EIs) | 4 | 4 | 16 |
| External Outputs (EOs) | 3 | 5 | 15 |
| External Inquiries (EQs) | 1 | 4 | 4 |
| External Interface Files (EIFs) | 1 | 10 | 10 |
| Internal Logical Files (ILFs) | 2 | 7 | 14 |
| **COUNT TOTAL** | |  | 59 |

|  |  |  |
| --- | --- | --- |
| **S No.** | **Factor** | **Value** |
| 1 | Does the system require reliable backup and recovery? | 5 |
| 2 | Are specialized data communications required to transfer info to or from the application? | 2 |
| 3 | Are there distributed processing functions? | 4 |
| 4 | Is performance critical? | 0 |
| 5 | Will the system run in an existing, heavily utilized operational environment? | 2 |
| 6 | Does the system require on-line data entry? | 3 |
| 7 | Does the on-line data entry require the input transaction to be built over multiple screens or operations? | 3 |
| 8 | Are the ILFs updated online? | 5 |
| 9 | Are the inputs, outputs, files, or inquiries complex? | 2 |
| 10 | Is the internal processing complex? | 2 |
| 11 | Is the code designed to be reusable? | 3 |
| 12 | Are conversions and installations included in the design? | 0 |
| 13 | Is the system designed for multiple installations in different organizations? | 5 |
| 14 | Is the application design to facilitate change and for ease of use by the user? | 5 |
| **TOTAL** | | 41 |

**Value Adjustment Factors, ΣFi = 41**

**Unadjusted Function Point (UFP) = 24+15+4+10+14=59**

**Function Point (FP)** = UFP \* [0.65 + 0.01\*∑ (Fi)]

= 59 x (0.65 + 0.01 x 41) = **64.52**

**EFFORT**

Assuming that Average productivity for such a system is 5FP/pm i.e. 5 functional points per person month.

Effort= Total Functional Points/Average Productivity

=64.52 /5 = 13 pm i.e. 13 person month

**6. RISK TABLE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risks** | **Category** | **Probability** | **Impact** | **Mitigation** |
| **Lack of skill** | ST | 40% | 2 | External resources might help |
| **Quality not maintained** | DE | 60% | 3 | Take up steps to maintain quality at each stage of development. |
| **Building the wrong product** | CU | 30% | 1 | Early and continuous validation is critically important. You need to establish a clear vision and solid justification for the product. |
| **Size estimates may be low** | PS | 50% | 2 | Past experiences must be considered and a similar task on a smaller scale may be attempted |
| **Requirements not properly documented and understood** | CU | 50% | 1 | Regular interaction with the customer and getting the requirements verified before finishing them. |
| **Delivery deadline will be tightened** | BU | 40% | 2 | Review the progress from time to time and take appropriate steps to keep up with the schedule. |

1-      Catastrophic

2-      Critical

3-      Marginal

4-      Negligible

PD- Process Definition

ST – Staff size and experience

DE – Development Environment

CU- Customer Characteristics

BU- Business Impact

PS – Product Size

**RISK EXPOSURE : P\*C**

**RE=37.6**

**7.PROJECT SCHEDULING**

|  |  |
| --- | --- |
| **WORK TASKS** | **SCHEDULE** |
| 1. **IDENTIFY NEEDS AND**   **BENEFITS**   * 1. Meet with concerned members   2. Identify needs and project constraints   3. Establish problem statement   4. Milestone  1. **REQUIREMENT ANALYSIS**     1. Detailed discussion of the project    2. Creating Data flow Diagram    3. Data Dictionary    4. Milestone 2. **PROJECT MANAGEMENT**     1. Computing F.P. and Effort    2. Schedule table    3. Risk table    4. Timeline Chart    5. Milestone 3. **DESIGN ENGINEERING**     1. Architectural Design    2. Data Design    3. Pseudo Code    4. Milestone 4. **TESTING** | Week 1  Week 1  Week 2  Week 2  Week 3  Week 4  Week 5  Week 5  Week 5  Week 6  Week 7  Week 8  Week 8  Week 8  Week 9  Week 9  Week 10  Week 12 |

#### 

P a g e 22

**8.ARCHITECTURAL DESIGN**

**Online tutor**

**Finding system**

**Teacher**

**Student**

**Storestudent**

**details**

**Displayclass**

**timings**

**Displaylistof**

**applicable**

**teachers**

**Matchstudent**

**requirements**

**andteacher**

**Storeteacher**

**details**

**Getteacher**

**details**

**Getteacher**

**docs**

**Payment**

**Gateway**

**Selectateacher**

**Getstudent**

**details**

**Getstudent**

**requirements**

**Confirmation**

**slip**

P a g e 23

## 9.CODE

1. <?php
2. include 'database.php';
3. $fname=$\_POST["fname"];
4. $lname=$\_POST["lname"];
5. $uname=$\_POST["username"];
6. $pass=$\_POST["password"];
7. $repass=$\_POST["re\_password"];
8. $gender=$\_POST["gender"];
9. $mobile=$\_POST["mobile"];
10. if(empty($fname)||empty($lname)||empty($uname)||empty

($pass)||empty($repass)||empty($gender)||empty($mobile))

1. { echo"fill out all the fields!"; }
2. else {
3. $sql="SELECT username FROM users WHERE username='$uname'";
4. $result=$conn->query($sql);
5. $rowcnt=mysqli\_num\_rows($result);
6. if($rowcnt!=0)
7. { echo "username u entered already exist!"; }
8. elseif($pass!=$repass)
9. { echo "passwords didn't match!"; }
10. elseif(strlen($mobile)!=10 || is\_numeric($mobile)!=true)
11. { echo "incorrect mobile number!"; }
12. else
13. {
14. $sql="INSERT INTO users

(username,password,gender)

1. VALUES ('$uname','$pass','$gender')";
2. $result=$conn->query($sql);
3. echo "account created successfully"; 28. }
4. }
5. ?>

## 10.TESTING

Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use.

Two techniques of Software Testing are:

**White Box Testing:** White Box Testing is a software testing method in which the internal structure/ design/ implementation of the item being tested is known to the tester. It is mainly applicable to lower levels of testing. Programming knowledge and implementation knowledge are required for this.

**Black Box Testing:** Black Box Testing is a software testing method in which the internal structure/ design/ implementation of the item being tested is NOT known to the tester. It is mainly applicable to higher levels of testing. Programming knowledge and implementation knowledge are not required for this.

### Basis Path Testing

In software engineering, basis path testing, or structured testing, is a white box method for designing test cases. The method analyzes the control flow graph of a program to find a set of linearly independent paths of execution. The test cases are prepared so that each execution path will occur at least once.

### 10.1 PROGRAM FLOW GRAPH

Page

28

1

2

3

4

5

6

16

18

23

17

7

8

9

10

15

11

12

13

14

26

27

24

22

21

20

19

25

28

30

29

#### Independent Paths

Number of independent paths=5

Path 1:

1—2—3—4—5—6—7—8—9—10—11—30

Path 2:

1—2—3—4—5—6—7—8—9—10—12—13—14—15—16—17—29—

30

Path 3:

1—2—3—4—5—6—7—8—9—10—12—13—14—15—16—18—19—

29—30

Path 4:

1—2—3—4—5—6—7—8—9—10—12—13—14—15—16—18—20—

21—29—30

Path 5:

1—2—3—4—5—6—7—8—9—10—12—13—14—15—16—18—20—

22—23—24—25—26—27—28—29—30

### 10.2 CYCLOMATIC COMPLEXITY

**Cyclomatic complexity** is a software metric (measurement), used to indicate the complexity of a program. It is a quantitative measure of the number of linearly independent paths through a program's source code. It was developed by Thomas J. McCabe, Sr. in 1976.

Cyclomatic complexity is computed using the control flow graph of the program: the nodes of the graph correspond to indivisible groups of commands of a program, and a directed edge connects two nodes if the second command might be executed immediately after the first command.

One testing strategy, called basis path testing by McCabe who first proposed it, is to test each linearly independent path through the program; in this case, the number of test cases will equal the cyclomatic complexity of the program.

Definition:

C = E – N +2

Where,

C = cyclomatic complexity

E = the number of edges of the graph

N = the number of nodes of the graph

P= number of connected components in the graph

Calculating Cyclomatic Complexity

E = 33

N = 32

CYCLOMATIC COMPLEXITY = E-N+2

= 33-30+2 =5

CYCLOMATIC COMPLEXITY = Number of Predicate nodes +1

Predicate nodes in the graph: 10, 16, 18, 20

CYCLOMATIC COMPLEXITY = 4+1 =5

**Therefore, Number of Independent Paths = Cyclomatic**

**Complexity = 5**

## 11.CONCLUSION

Online Tutor Finding System is platform between teachers looking for jobs and students looking for subject specific teachers. It works as an intermediate. Waterfall model has been used in this project. All the requirements and constraints have been kept in mind while development of Online Tutor Finding System. The project works well with both of its users.

## 12.BIBLIOGRAPHY

The successful completion of this project has been achieved by the assistance from various resources which includes:

 Software Engineering — A Practitioner’s Approach, VI Edition by—Roger S. Pressman.

 An Integrated Approach to Software Engineering, III Edition by—

P. Jalote.