3 Level Authentication

Software Engineering Project

3 LEVEL AUTHENTICATION

SOFTWARE ENGINEERING PROJECT REPORT

B. Sc. (H) Computer Science

SUBMITTED BY:

Rallapalli Nagarjun	Sachin Kumar	Shashikant Patel	Yukti Kaushik
16035500000	15035500000	15035500000	16035500000

Keshav Mahavidyalaya

(University of Delhi)

ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them. We are highly indebted to **TutorialsDuniya.Com** for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would like to express our gratitude towards our teachers for their kind co-operation and encouragement which helped us in completion of this project.
A huge thanks and appreciations also go to the team members in developing the project and people who have willingly helped us out with their abilities.

CERTIFICATE

This is to certify that Software Engineering project report entitled "3 Level Authentication" is the work carried out by Rallapalli Nagarjun, Sachin Kumar, Shashikant Patel and Yukti Kaushik, student of BSc(H) Computer Science 4th Semester, Keshav Mahavidyalaya, University of Delhi under the supervision of TutorialsDuniya.com.

This report has not been submitted to any other organization/institution for the award of any other degree/diploma.

Project Supervisor

TutorialsDuniya.com

Teacher-In-Charge

TutorialsDuniya.com

Principal

TutorialsDuniya.com

ABSTRACT

The project is an authentication system that validates user for accessing the system only when they input correct password. The project involves three levels of user authentication. There are varieties of password systems available, many of which have failed due to bot attacks while few have sustained it but to a limit. In short, almost all the passwords available today can be broken to a limit. Hence this project is aimed to achieve the highest security in authenticating users.
It contains three authentication phases having three different kinds of password system. The password difficulty increases with each level. Users have to input correct password for successful login. Users would be given privilege to set passwords according to their wish. The project comprises of text password i.e. pass phrase, image based password and OTP password for the three levels respectively. This way there would be negligible chances of bot or anyone to crack passwords even if they have cracked the first level or second level, it would be impossible to crack the third one. Hence while creating the technology the emphasis was put on the use of innovative and nontraditional methods. Many users find the most widespread text-based password systems unfriendly, so in the case of three level password we tried creating a simple user interface and providing users with the best possible comfort in solving password.

TABLE OF CONTENTS

List of Figures

List of Tables

1. Software Requirement Specification

1.1. Introduction

1.1.1. Purpose

The purpose of product is to provide high level protection from thefts involving privacy of data. In this fast advancing world, almost all the security systems are becoming obsolete. So, something is needed to increase the level of security for our data, that's why this application is made.

1.1.1 Document Conventions

The font used for headings are Times with BOLD function and font used for content part in a paragraph is Arial with italics function. In the paragraph body, special lines have been quoted and highlighted with bold function.
Every heading is followed by its sub headings described by 1.1, 1.2.

1.1.3. Intended Audience and Reading Suggestions

This document is made for all the kind of readers who intend to get a brief knowledge about what this software is about.
The developers should directly jump to part 2, that is, Overall Description to part 5 of the document, that is, Other Non - functional Requirements.
The project managers need to read all the parts from part 1 to part 6.
The marketing staff should read part 5 Other Non – functional Requirements.
The users should read part 2, that is, Overall Description.
The testers should read part 2, 3 and 4.
The documentation writers should read part 2 to part 6.

1.1.4 Product Scope

The main objective is to provide high level of security so that users can rely on the storage that their data is secure. The more authentication methods get introduced in the near future, the more updates will be patched to the main software. **"The users can rely on our database for saving their sensitive data."** this is the prime objective of our software.

1.1.5 References

This SRS has been referred from SRS template provided by IEEE.

1.2 Overall Description

1.2.1 Product Perspective

Nowadays, many hackers hack into our system and can use our sensitive information for their purpose as mostly we are relying upon text-based password system where our username identifies us and password validates us. But this already existing technique has some weaknesses as more than one person can possess its knowledge at one time as the textual password can be easily known by using brute force method and hence our identity can land into trouble with people having venomous intent. Thus, along with text-based password two more techniques have been providednamely picture based password and OTP generated password.

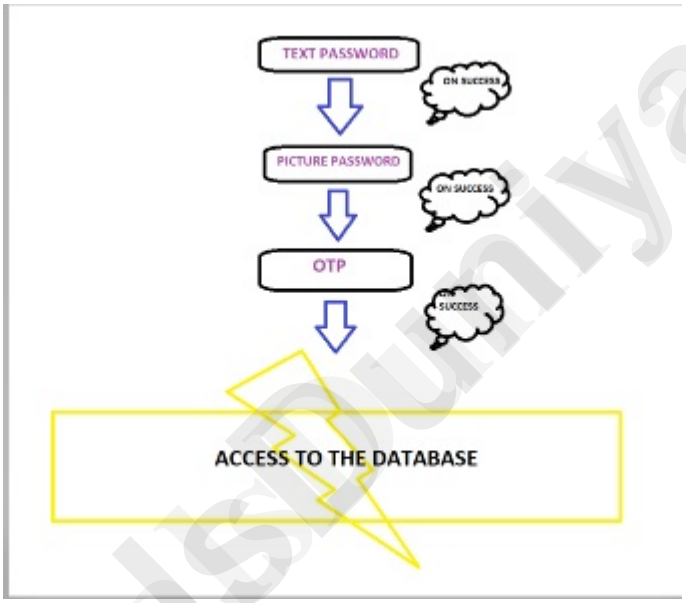

FIGURE 1: PROJECT OVERVIEW

1.2.2 Product Functions

- The user will be provided a sign-up page for the first time and sign in page after sign up for that particular user.
- When the user signs up, he will provide a mobile number/email on which an OTP system will be used.
- Also, the user will select a group of photos for which we will assign a picture password.\
- Once the user successfully logs in, he will have access to his own previously saved data or he can add more data to his online storage which is provided to each specific user (to a certain limit).

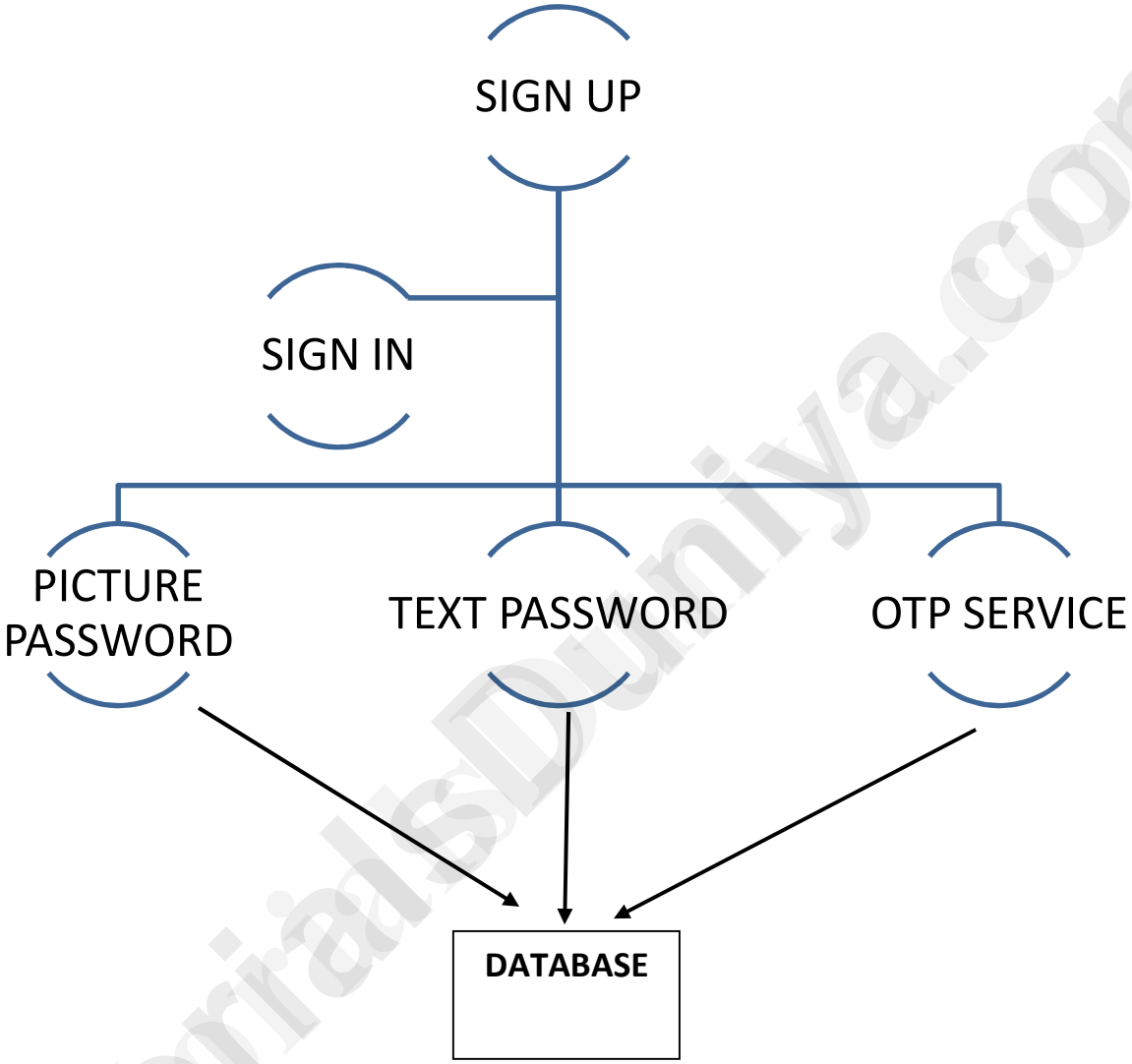

FIGURE 2: PROJECT IMPLEMENTATION

1.2.3 User Classes and Characteristics

This software can be used by any type of user. Everyone who wants to save their data from any attacks and maintain their privacy, can use this software.

1.2.4 Operating Environment

This software can operate on any hardware platform, as it is a web storage type of software.Any browser can be used to access this software. The browser should support JavaScript.

1.2.5 Design and Implementation Constraints

This system limits to one user at a time.

1.2.6 User Documentation

A video will be provided on how to sign up and initialize the user page for inputting the text password, picture password and enabling OTP service.
A tutorial will be provided on how to upload your files on the cloud storage assigned to you.
Also, the directives that would be followed in case of **"FORGOT PASSWORD"** situation will also be provided along with the user manual.

1.2.7 Assumptions and Dependencies

We are assuming that the OTP service provided is reliable and we can have no more than 1 person at a time accessing the storage.

1.3 External Interface Requirements

1.3.1 User Interfaces

The interface used in each webpage is similar to the Gmail login page.
The browser on which the user runs the webpage should be preferably 1024 x 768 px.
The errors are generally shown in the text fields and the errors that are due to database non connection or smtp server failure.

1.3.2 Hardware Interfaces

The interface uses PHP in backend for which a XAMPP server is being used.
And for sending OTP, the gmail SMTP server is being used.

The port for SMTP Gmail server is PORT 587 and for Apache server port 80 and MySQL port 3306. And PHPMyAdmin is the database used for MySQL queries.

1.3.3 Software Interfaces

reCAPTCHA API used for CAPTCHA.
PHPMyAdmin used for MySQL server.
Gmail SMTP server for sending mails.
PHPMailer and SMTP classes used for OTP functionality.
Refer to User Manual

1.3.4 Communications Interfaces

The port for SMTP Gmail server is PORT 587 and for Apache server port 80 and MySQL port 3306. And PHPMyAdmin is the database used for MySQL queries.

1.4 System Features

Refer to Brief Overview.

1.4.1 System Feature 1

LOGIN

1.4.1.1 Description and Priority

Priority: HIGH.

The user types in his username and password that he had used for signup.

1.4.1.2 Stimulus/Response Sequences

As the user hits the validate button in the login page, firstly the text field replies whether the inputted data is correct or not.

Then, user clicks the proceed button and proceeds to reCAPTCHA page.

1.4.1.3 Functional Requirements

REQ-1: The database should be connected.
REQ-2: The proceed button only works when correct data is validated

1.4.2 System Feature 2

SIGN UP

1.4.2.1 Description and Priority

Priority: HIGH.

The user types in his username and password.

1.4.2.2 Stimulus/Response Sequences

As the user hits the validate button in the signup page, firstly the text field replies whether the inputted data is correct or not.

Then, user clicks the proceed button and proceeds to reCAPTCHA page.

1.4.2.3 Functional Requirements

REQ-1: The database should be connected.
REQ-2: The proceed button only works when correct data is validated

1.4.3 System Feature 3

reCAPTCHA

1.4.3.1 Description and Priority

Priority: LOW.

The user selects the pattern and clicks proceed.

1.4.3.2 Stimulus/Response Sequences

As the user hits the proceed button in the reCAPTCHA page, the pattern is cross checked and if correct,

Then, user clicks the proceedbutton and proceeds to OTP page.

1.4.3.3 Functional Requirements

REQ-1: The internet should be connected
REQ-2: The proceed button only works when correct data is validated

1.4.4 System Feature 4

OTP

1.4.4.1 Description and Priority

Priority: HIGH.

The user types in his username and then the OTP received.

1.4.4.2 Stimulus/Response Sequences

As the user hits the validate UID button in the OTP page, firstly the text field replies whether the inputted data is correct or not.

Then the user receives the OTP on his email id and then hits validate button.

Then, user clicks the proceed button and proceeds to web page.

1.4.4.3 Functional Requirements

REQ-1: The internet should be connected.
REQ-2: The proceed button only works when correct data is validated
REQ-3: The database should be connected.

1.5 Other Nonfunctional Requirements

1.5.1 Performance Requirements

This software can be used by 5 users at a time with each user occupying 100 MB of space.

1.5.2 Safety Requirements

Safety of data is given prime importance in this software where users' data is stored and can only be accessed by his/her permission by authenticating through various phases provided. There will be no one accessing your storage other than yourself.
By providing OTP from the system only the intended user can achieve access onto his data allowing full safety to the data. Even if he/she by mistakenly shares the password of account then also full rights of retrieving the data remains with him/her as other users will not be able to gain access and will be blocked after several attempts.

1.5.3 Security Requirements

The user can jump to another webpage on the site by changing the URL to the location file.
The user can use SQL injection to break into the database for UIDs.
The proceed button in some pages work without validating causing chaos.

1.5.4 Software Quality Attributes

The software is correct, flexible, maintainable, portable and reusable.
But it is not robust because it is vulnerable to attacks and flaws.

This software is limited to localhost that means the system of the project invoking the files, therefore, it has very limited usage capability.

Appendix A: Analysis Models

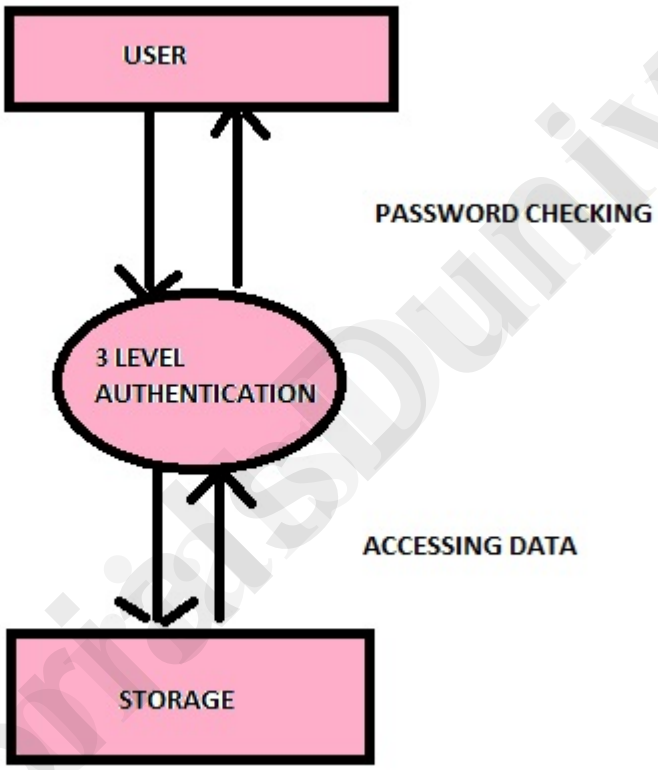

FIGURE 3: LEVEL 0 Data Flow Diagram

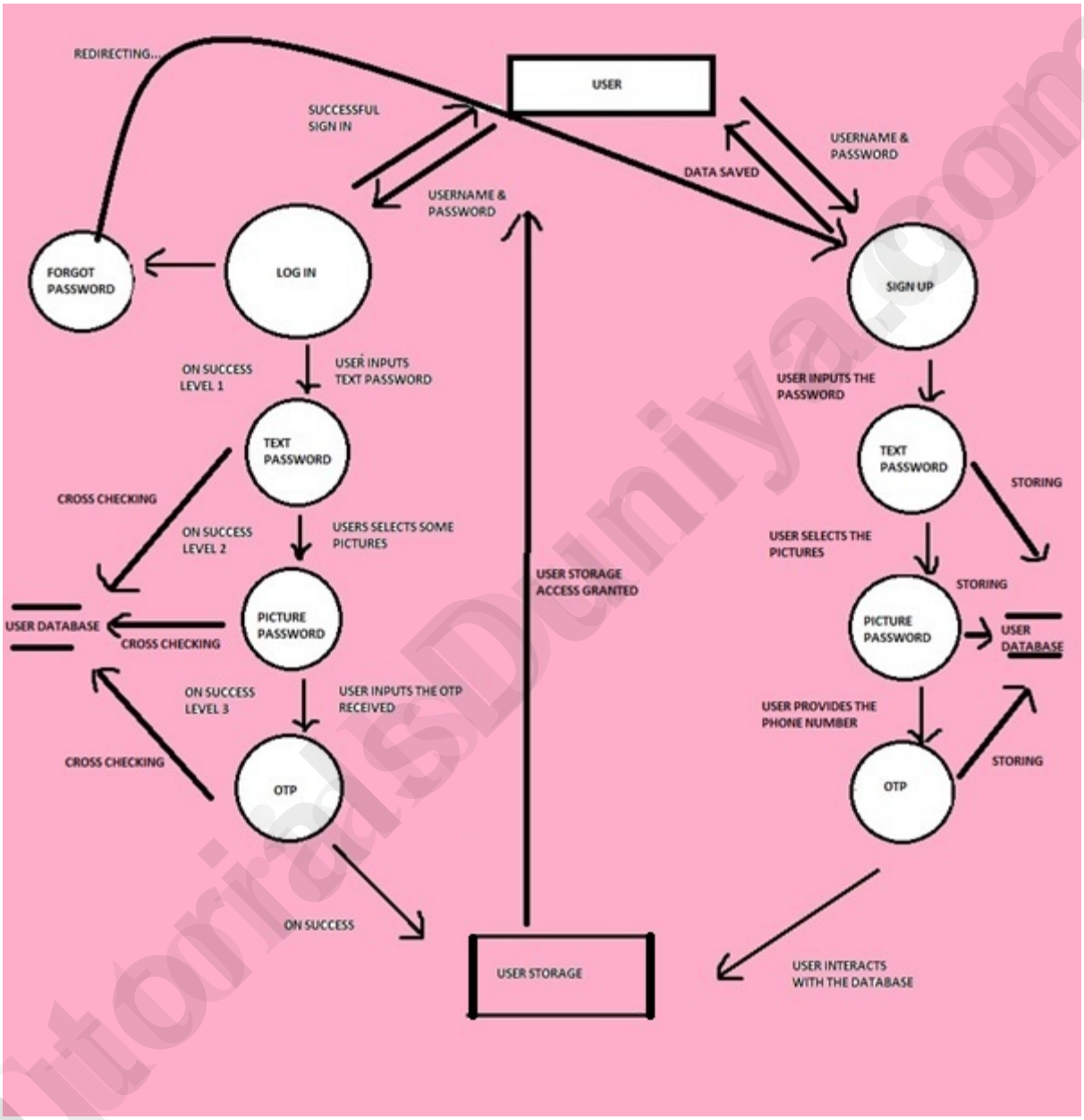

FIGURE 4: LEVEL 1 Data Flow Diagram

1.6 Data Dictionary

1.6.1 Log in <=> User:

I USER TO LOGIN DATA FLOW

(i)description:password

(ii)source:user

(iii)destination:login process (database = "uid")

(iv)type:user id (combination of alphanumeric and special characters)

II SUCCESSFUL LOGIN

(i)description:access to user storage

(ii)source: storage

(iii)destination: user

(iv)type:session started (on cross checking uid and password)

III TEXT PASSWORD

(i)description : user inputs text password

(ii)source: user

(iii)destination : PICTURE PASSWORD

(iv) type : alphanumeric password

IV PICTURE PASSWORD

(i) description : user selects some specific pictures

(ii) source : user

(iii) destination : OTP

(iv) type : picture password (picture size less than 1 MB each)

V OTP

(i) description: user inputs the OTP received by him/her

(ii) source : user

(iii) destination : OTP

(iv) type : One Time Password (generally numeric)

VI FORGOT PASSWORD

(i) description : user inputs the answer to the security question

(ii) source : user

(iii) destination : SIGN UP

(iv) type : alphanumeric

VII ON SUCCESS

(i)description : user gets access to his storage

(ii) source : database

(iii) destination : user

(iv) type : general media files

VIII CROSS CHECKING

(i)description:textpassword,otppassword,picture password are

cross checked by storing it in datastore for granting further access

(ii)source:processes = text password + picture password + otp

(iii)destination:datastore:otp+text+picture

(iv)type:alphanumeric(text password) + numeric(otp) + image(picture password)

1.6.2 SIGN UP

I FIRST SESSION

(i) description : user inputs username

(ii)source : user

(iii) destination :datastore

(iv) type : alphanumeric

II TEXT PASSWORD

(i)description : user inputs password

(ii) source : user

(iii) destination :datastore

(iv) type : alphanumeric

III PICTURE PASSWORD

(i)description : user selects pictures

(ii)source : user

(iii) destination :datastore

(iv) type : media files (size < 1 MB each)

IV OTP

(i) description : user inputs his mobile number

(ii) source : user

(iii) destination :datastore

(iv) type : numeric (=10 digit number)

V INTERACTING WITH USER STORAGE

(i) description: user interacts with his storage

(ii) source : user

(iii) destination : storage

(iv) type : media files

2. Size Estimation and Scheduling

2.1 Size Estimation

EQ TABLE			
Enquiry	weighing factor	weight	description
eq1	D	6	User database interaction
eq2	M	4	3rd party OTP service
	sum=	10	

TABLE 1: EXTERNAL ENQUIRIES

ILF TABLE			
Process	Weighing factor	weight	description
p1	D	15	user database
p2	D	15	login
p3	M	10	text password
p4	D	15	picture password
p5	S	7	OTP
p6	S	7	sign up
p7	M	10	forgot password
	sum=	79	

TABLE 2: INTERNAL LOGICAL FILES

EI TABLE

Input	Weighing factor	Weight	Description
i1	M	4	text password
i2	D	6	picture password
i3	S	3	OTP
i4	S	3	Username
i5	D	6	Storing in database
	sum=	22	

TABLE 3: EXTERNAL INPUT

EO TABLE

Output	Weighing factor	Weight	Description
o1	D	7	user storage access
o2	M	5	text password to picture password
o3	M	5	picture password to OTP
o4	D	7	Login
o5	S	4	signup
	sum=	28	

TABLE 4: EXTERNAL OUTPUT

EIF TABLE			
Process	weighing factor	weight	Description
p1	M	7	3rd Party OTP
p2	S	5	User Storage
	sum=	12	

TABLE 5: EXTERNAL INTERFACE FILES

CAF (Cumulative Adjustment Factor)= 0.65 + (0.01 * 14 *3) =1.07

Functional Unit	Number	Difficulty Level
EIF	2	12
ILF	7	79
EO	5	28
EI	5	22
EQ	2	10
	UFP=	151

TABLE 6: FUNCTION POINT ANALYSIS

FP (Function Point)= 1.07 * 151 = 161.57
LOC(Lines of code) = FP xLOC PER FP OF LANGUAGE = 161.57 x 67 = 10,825.19

EFFORT=1.4XL^0.93	**= 1.4 X (10,825.19)^0.93 = 7909.57**
DOCUMENTATION=	**30.4 x (10,825.19)^0.9 = 129,974.66**
DURATION =	**4.6 x (10,825.19)^0.26 = 51.488**

Assumptions taken during functional point analysis:

EI has taken input that is given to process/database

EO has given output that is the outcome of anyprocess/database

Processes are considered as ILF i.e. internal logical files

2.2 Project Scheduling

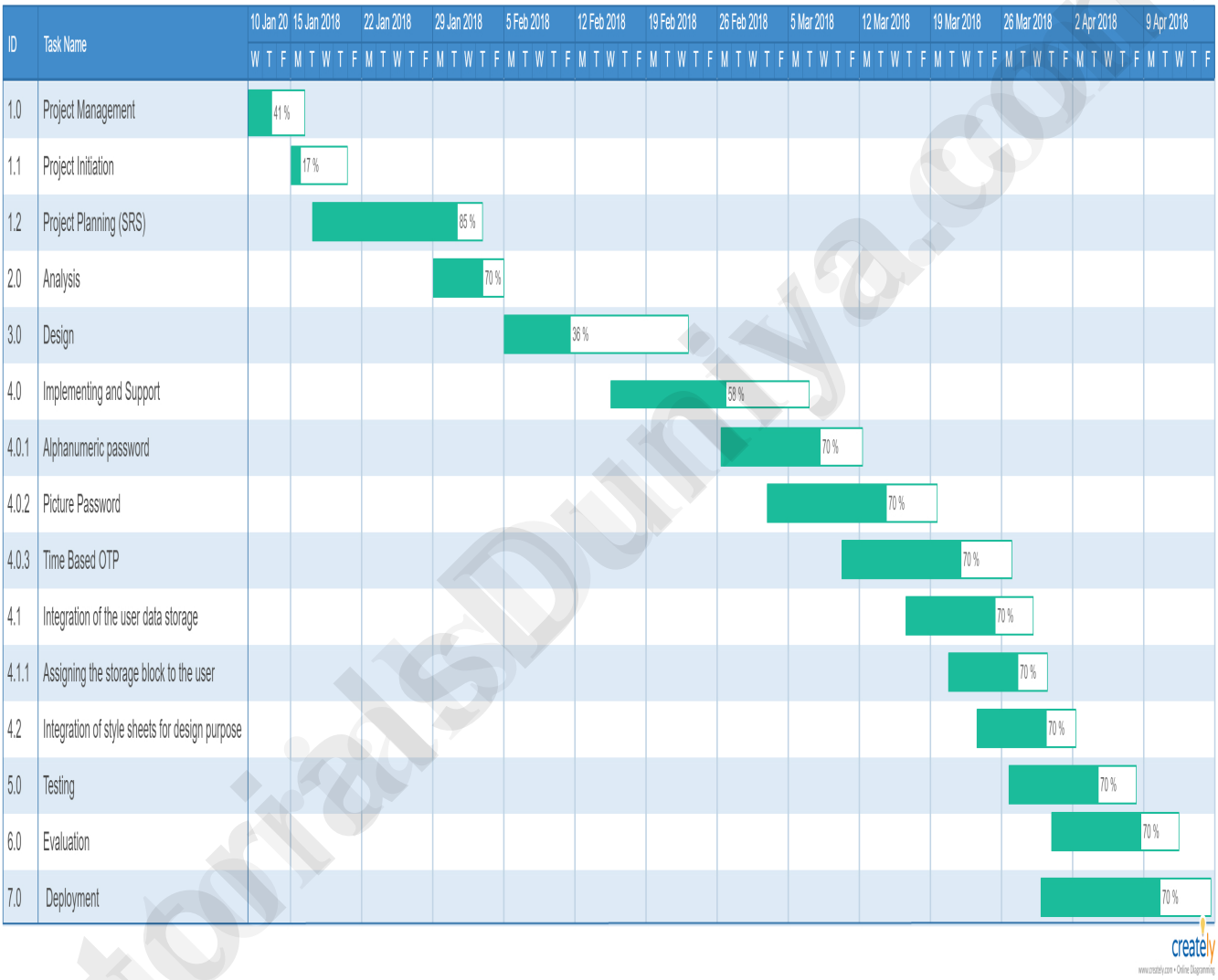

TABLE 7: Gantt chart

3. Architectural Design

Level 1

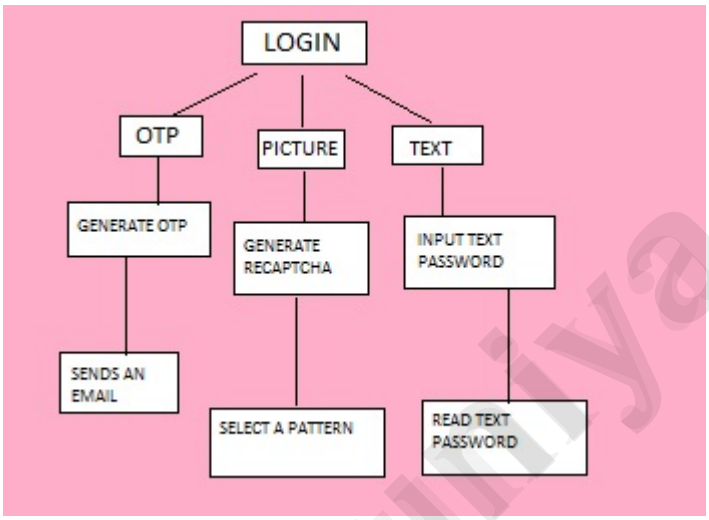

FIGURE 5: ARCHITECTURAL DESIGN LEVEL 1

Level 2

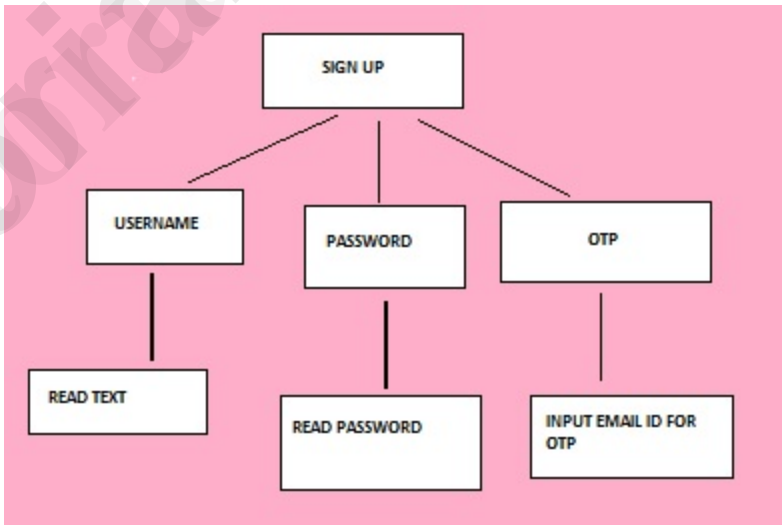

FIGURE 6: ARCHITECTURAL DESIGN LEVEL 2

4. Risk Analysis

4.1Risk Mitigation, Monitoring and Management Plan

Risk Mitigation, Monitoring and Management 1:

RISK INFORMATION SHEET			
Risk ID: R01	DATE: 10-02-2018	Probability: 80%	Impact: high
DESCRIPTION Lack of training on tools required to build the project			
REFINEMENT/CONTEXT: 1. Certain reusable components were developed by a third party with no knowledge of internal design standards. 2. Certain reusable components have been implemented in a language that is not supported on the target environment.			
MITIGATION/MONITORING: 1. Contact third party to determine conformance with design standards. 2. Check to determine if language support can be acquired.			
MANAGEMENT/CONTINGENCY PLAN/TRIGGER 1. Risk estimation computed to be ₹5000. Allocate this amount within project contingency cost. Develop revised schedule assuming that 18 additional components will have to be custom built; allocate staff accordingly. **Trigger:** Mitigation steps unproductive as of 20/02/2018			
CURRENT STATUS: **05/03/2018:** Mitigation steps initiated.			
Originator : Yukti Kaushik		**Assigned :** Rallapalli Nagarjun	

TABLE 8: RMMM 1

Risk Mitigation, Monitoring and Management 2:

RISK INFORMATION SHEET			
Risk ID: R02	DATE: 15-02-2018	Probability: 80%	Impact: Very high
DESCRIPTION: Staff lacks experience in skills required to complete the project.			
REFINEMENT/CONTEXT: 3. More work load on the experienced people. 4. Time exceeded due to inexperienced staff.			
MITIGATION/MONITORING: 3. Providing required knowledge regarding the project. 4. Bring in skilled staff.			
MANAGEMENT/CONTINGENCY PLAN/TRIGGER 2. Risk estimation computed to be ₹50000. Allocate this amount within project contingency cost. Develop revised schedule assuming that 18 additional components will have to be custom built; allocate staff accordingly. **Trigger:** Mitigation steps unproductive as of 23/02/2018			
CURRENT STATUS: **10/03/2018:** Mitigation steps initiated.			
Originator : Yukti Kaushik		**Assigned :** Rallapalli Nagarjun	

TABLE 9:RMMM 2

5. Implementation of Module

5.1 Coding

Otp.Php One Time Password in Project 3rd Module

```
<?php
// Import PHPMailer classes into the global namespace
// These must be at the top of your script, not inside a function
require 'Exception.php';


//Load composer's autoloader
require 'PHPMailerAutoload.php';
functionsendOTP($email,$otp){
   $mail = new PHPMailer(true);                        // Passing `true` enables exceptions
try {
   //Server settings
      $mail->SMTPDebug = 2;                              // Enable verbose debug output
      $mail->isSMTP();                                   // Set mailer to use SMTP
      $mail->Host = 'smtp.gmail.com';  // Specify main and backup SMTP servers
      $mail->SMTPAuth = true;                            // Enable SMTP authentication
      $mail->Username = 'rallapallinagarjun16@gmail.com';                // SMTP username
      $mail->Password = '16121998';                           // SMTP password
   // $mail->SMTPSecure = 'tls';                         // Enable TLS encryption, `ssl` also accepted
      $mail->Port = 25;                                  // TCP port to connect to
      // $message_body = "One Time Password for PHP login authentication is:<br/><br/>" . $otp;
   //Recipients
```

```
$mail->setFrom('rallapallinagarjun16@gmail.com', '3-LEVEL Authentication');
        $mail->addAddress($email);     // Add a recipient
    //Content
       // $mail->MsgHTML($message_body);
        $mail->isHTML(true);
     // Set email format to HTML
        $mail->Subject = 'OTP for login/signup';
        $mail->Body = "One Time Password for PHP login authentication is:<br/><br/>" . $otp;
        $result1 = $mail->send();
echo '<h1><center><b>Message has been sent</b></center></h1>';
return $result1;
    }
catch (Exception $e) {
echo 'Message could not be sent. Mailer Error: ', $mail->ErrorInfo;
return 0;
    }
}
?>
<?php
    $success=null;
            $error_message = null;
    $emailref = null;
            $conn = mysqli_connect("localhost","root","","mydb");
            if(isset($_POST["name1"]) && !empty($_POST["name1"])){
                        $result = mysqli_query($conn,"SELECT * FROM Baits WHERE email = '" .
$_POST['name1'] . "'");
```

```
                $count1 = mysqli_num_rows($result);
                if($count1>0){
                        //generate OTP
                        $otp = rand(100000,999999);
                        //Send OTP
                        $mail_status = sendOTP($_POST['name1'],$otp);
                        if($mail_status == 1){
                                $result = mysqli_query($conn,"INSERT INTO
otp_expiry(otp,is_expired,create_at) VALUES ('".$otp."',0,'" . date("Y-m-d H:i:s")."')");
                                //$current_id = mysqli_insert_id($conn);
                                // if(!empty($current_id)){
                                //      $success =1;
                                // }
                        }
else{
        $error_message = "error";
    }
                }
                else{
                        $error_message = "Email is not present in the database Either Sign Up or get the
hell out of here!";
                }
        }
        else if(isset($_POST["OTPrec"]) && !empty($_POST["OTPrec"])){
                $result2 = mysqli_query($conn,"SELECT * FROM otp_expiry WHERE otp =
'".$_POST["OTPrec"]."' AND is_expired<> 1");
```

```
$count = mysqli_num_rows($result2);
                    if($count>0){
                        $result2 = mysqli_query($conn,"UPDATEotp_expiry SET is_expired = 1
WHERE otp = '".$_POST["OTPrec"]."');
                        $error_message = "success";
header("Location:C:\xampp\htdocs\home2.php");
                    }
                    else{
                        $error_message = "Invalid OTP!";
                    }
            }

?>
<!DOCTYPE html>
<html>
<head>
        <title>OTP page</title>
        <link rel = "stylesheet" href = "OTP.css"/>
        <link href = "https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300" rel =
"stylesheet" type = "text/css"/>
        <link href = "https://fonts.googleapis.com/css?family=Open+Sans:300" rel = "stylesheet" type =
"text/css"/>
</head>
<body>
        <center>
<div class = "base">
```

```
<div id = "logo">
<imgsrc = "logo.png" width = "118" height = "138"/>
</div>
<div id = "info1">
        Input your email for receiving OTP
</div>
<div id = "info2">
        Enter OTP after submitting your email
</div>
<div id = "form1">
<form action = "OTP.php" method = "POST">
<div id = "mailbox">
<input placeholder = " Enter your email"
type = "email" name = "name1" style="width:270px; height:42px; border:solid 1px #c2c4c6; font-
size:16px;
padding-left:8px" autofocus  />
</div>
<div id = "OTPfield">
<input placeholder="Enter the OTP" name ="OTPrec" style="width:270px; height:42px; border:solid 1px
#c2c4c6; font-size:16px;
padding-left:8px"/>
</div>
<div id = "passusercheck" style = "width:270px; height:42px; border:solid 1px #c2c4c6; font-size:16px;
        padding-left:8px">
<?php
```

```
echo $error_message;
                ?>
</div>
<div>
<input type ="submit" id="button1" value ="Validate Email" />
</div>
<div>
<input type ="submit" id="button2" value ="Validate OTP" />
</div>
<div id = "jump1">

<button class = "button3" onclick="myFunction()" >
<span>
<a href ="#">Proceed</a>
</span>
</button>

</div>
</div>
<div id = "bottom">
<p>&copy; Designed by Yk<br></p>
</div>
</div>
    </center>
<script>
```

```
functionmyFunction(){
var x = document.getElementById("jump1");
var y = "<?php echo $error_message; ?>";
if(y!="success"){
if(x.style.display === "none"){
x.style.display ="block";
        }
else{
x.style.display = "none";
        }
      }
   }
</script>
</body>
</html>
```

5.2 Snapshots of Module29

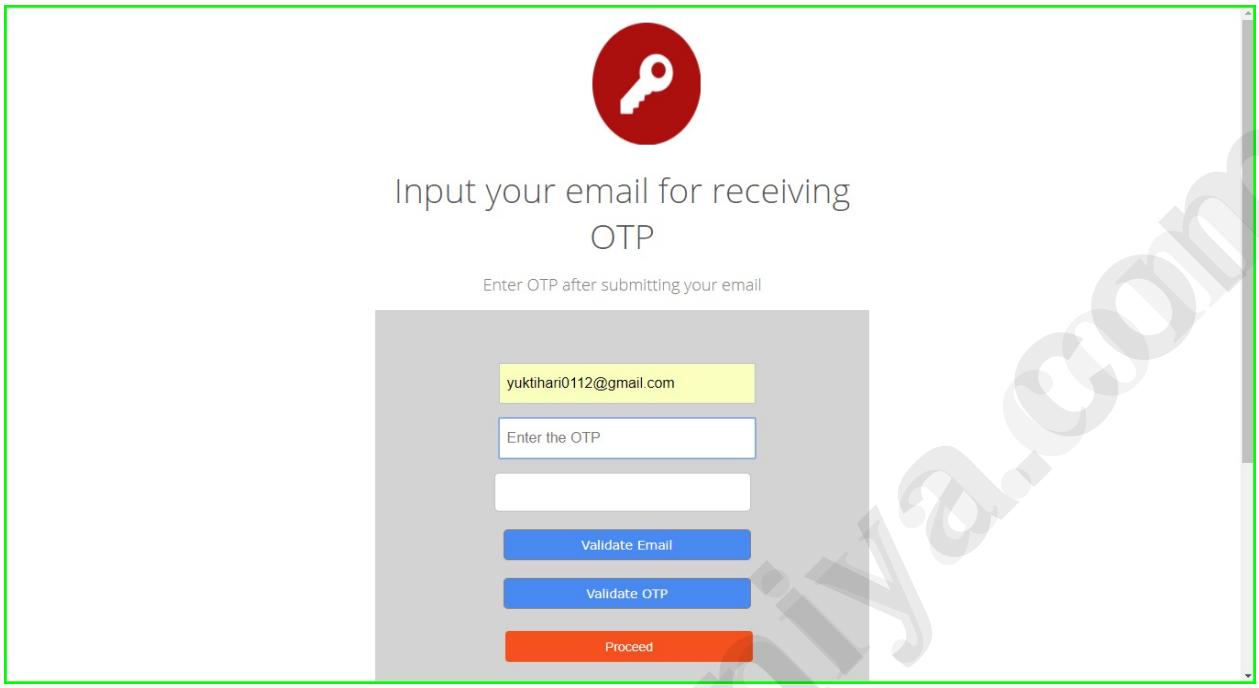

Figure 7: Snapshot 1

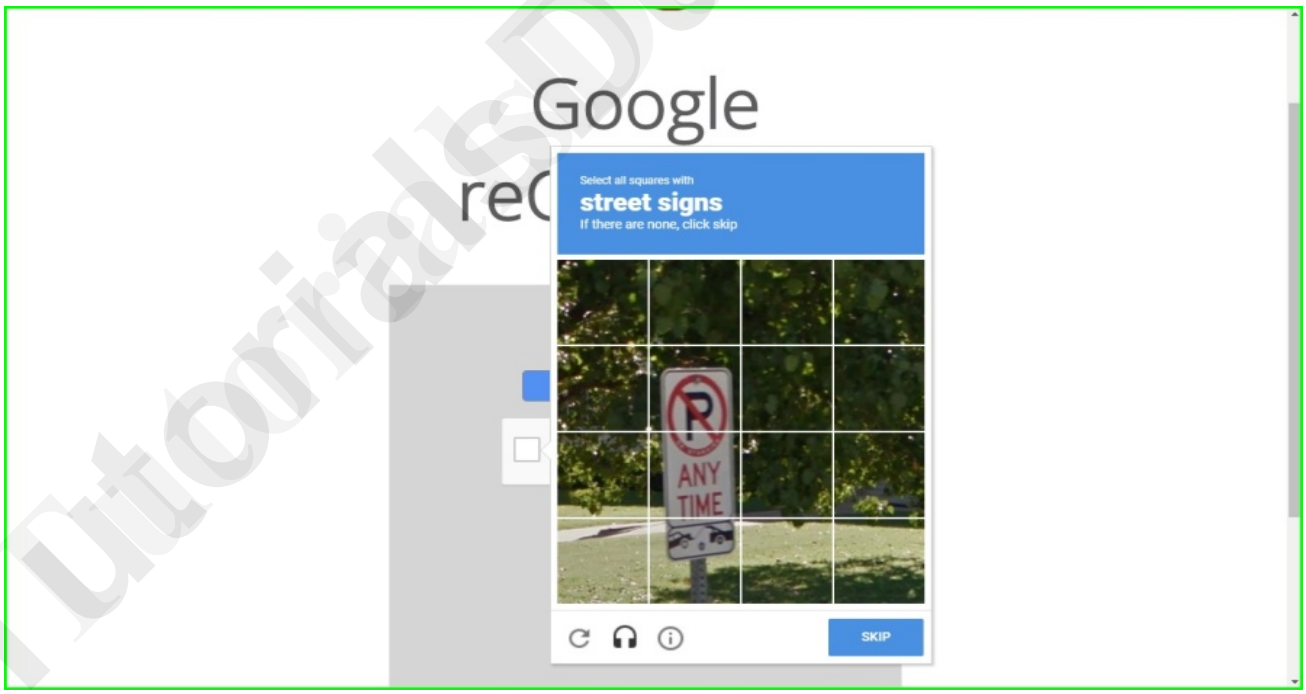

Figure 8: Snapshot 2

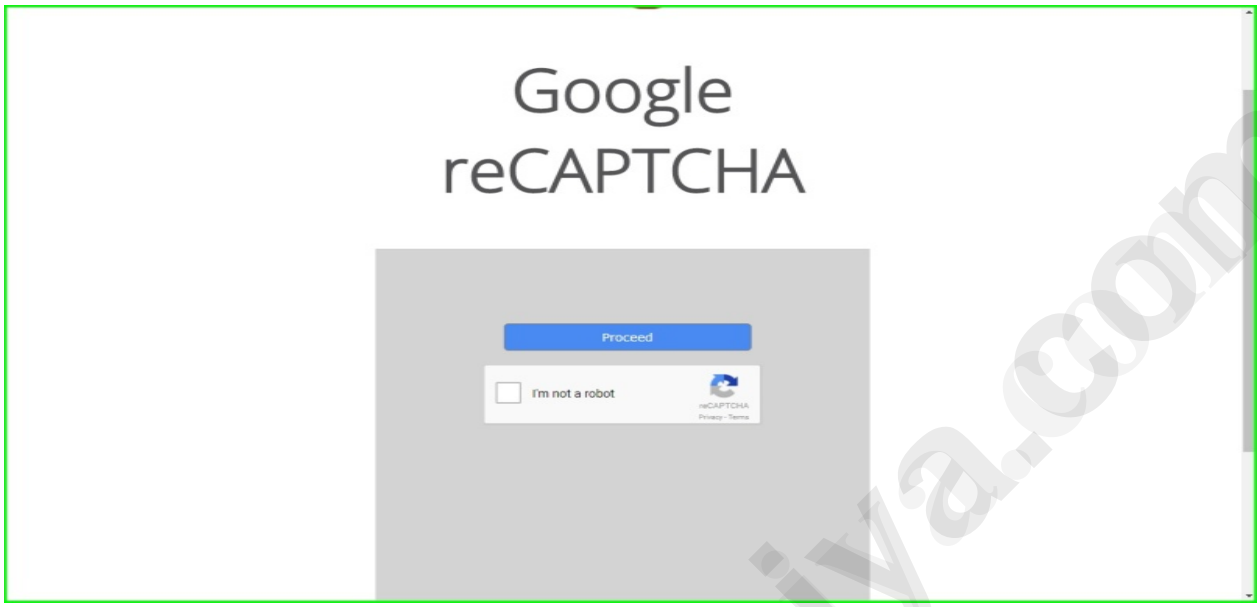

Figure 9: Snapshot 3

Figure 10: Snapshot 4

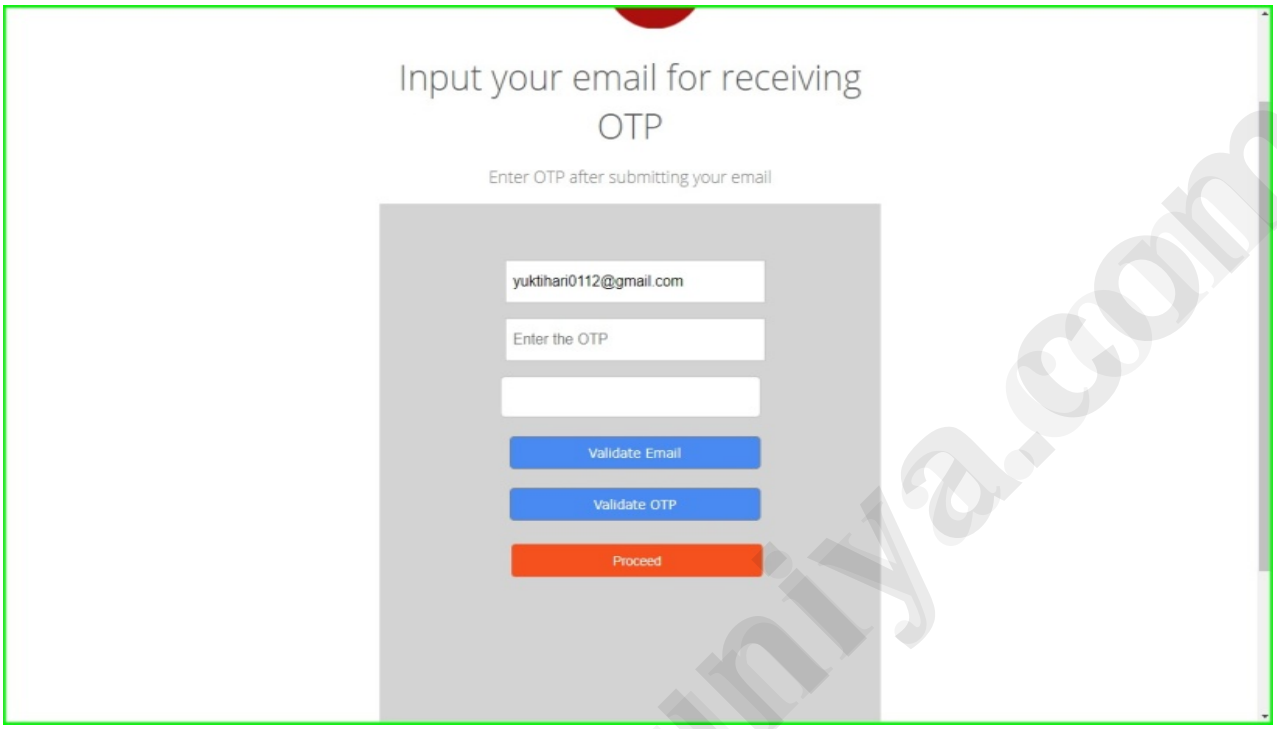

FIGURE 11: Snapshot 5

6. Testing

Basis Path Testing Of theRecaptcha Module

```php
<?php
        $captcha=null;
        if(isset($_POST['g-recaptcha-response'])){
          $captcha=$_POST['g-recaptcha-response'];
        }
        if(!$captcha){
          echo '<h2>Please check the captcha form.</h2>';
          exit;
        }
        $secretKey = "6LcLiUoUAAAAAN9LXAWHbxr4-YZ4m_6Wm8Uljzbb";
        $ip = $_SERVER['REMOTE_ADDR'];
        $response=file_get_contents("https://www.google.com/recaptcha/
            api/siteverify?secret=".$secretKey."&response=".$captcha."
            &remoteip=".$ip);
        $responseKeys = json_decode($response,true);
        if(intval($responseKeys["success"]) !== 1) {
          echo '<h2>You are a spammer ! </h2>';
        }
        else {
          header("Location: ..\PHPMailer\OTP.php");
          exit();
        }
?>
```

Figure 12: Code module

Program Graph of the above code:

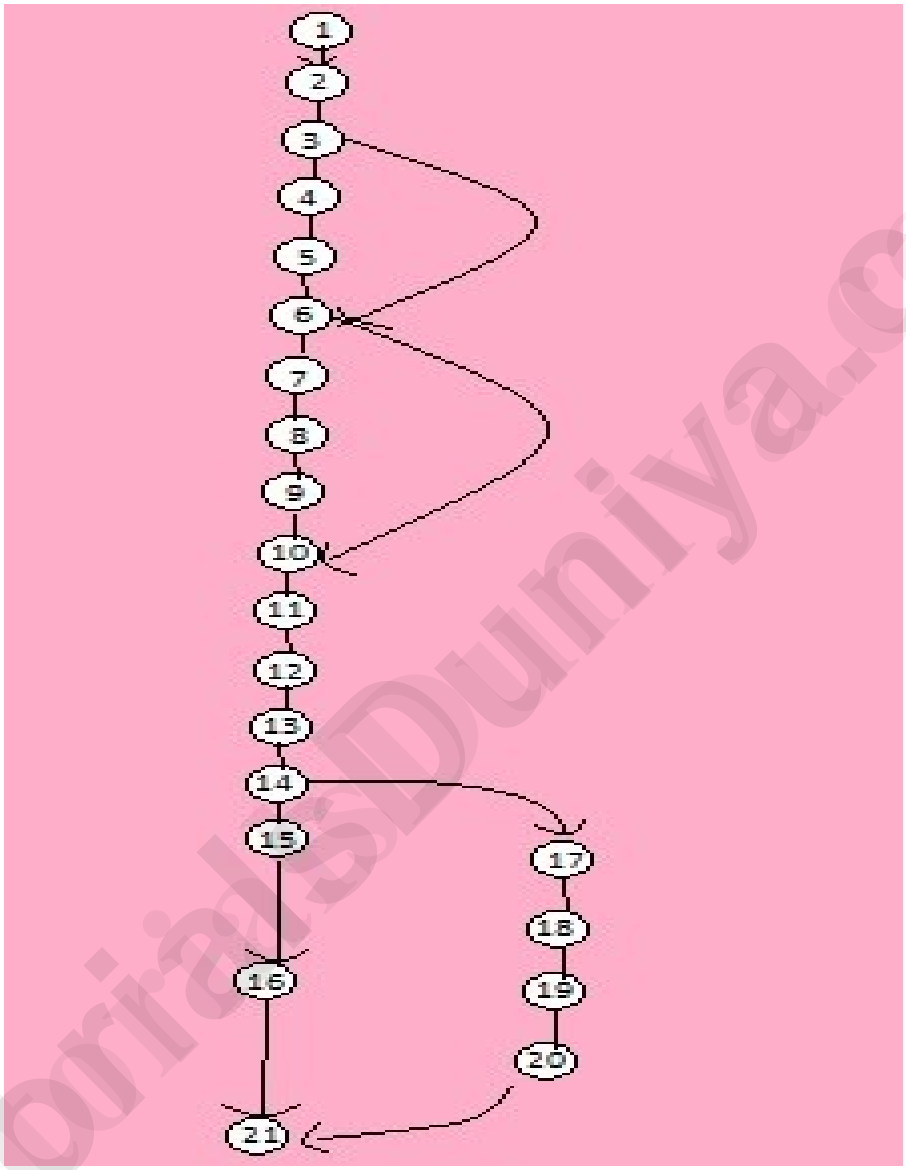

Figure 13: Program graph

Cyclomatic complexity of the given graph

VG = E – N + 2

VG = 23 – 21 +2

= 4

No. of regions = 3 + 1

= 4

No. of predicate nodes = 3

Cyclomatic complexity = Predicate nodes + 1 = 3+1 = 4

No. of independent nodes =

- 1-3 , (3,6) , (6,10) , 10-14 , 14-21
- 1-3, (3,6), 6-10, 10 -14, 14-15-16-21
- 1-6, 6-10, 10-14, 14-20, 20-21
- 1-14, 14-20 , 21

Which is equal to 4

No. of independent nodes = Cyclomatic complexity = 4

Test cases:

Path no.	Input	Expected Output	Received output
• 1-3 , (3,6) , (6,10) , 10-14 , 14-21	$_POST['g-recaptcha-response'] $captcha $responsekeys['success']	$captcha != null $captcha != null Another file is opened	$captcha != null $captcha != null Another file is opened
•1-3, (3,6), 6-10, 10 -14, 14-15-16-21	$_POST['g-recaptcha-response'] $captcha $responsekeys['success']	$captcha != null $captcha != null You are a spammer	$captcha != null $captcha != null You are a spammer

•1-6, 6-10, 10-14, 14-20, 20-21	$_POST[‘g-recaptcha-response’] $captcha $responsekeys[‘success’]	$captcha != null Please check the captcha form exit	35 $captcha != null Please check the captcha form exit
	$_POST[‘g-recaptcha-response’] $captcha $responsekeys[‘success’]	$captcha != null $captcha != null Another file is opened	$captcha != null $captcha != null Another file is opened

Table 10: Test case analysis

7.User Manual for the Project: 3 Level Authentication

The project's main objective is to provide three level authentication for the user to access a webpage.

The homepage is used for first level which is login page to login and proceed to the next authentication which is reCAPTCHA. The homepage has the link to the Sign Up page naming **Create Account**. For every page the user is directed to, he has to firstly validate his inputted data. Then, after successful validation, the user has to click the proceed button. If the user clicks the proceed button before the validation procedure, the **proceed** button will vanish and the user has to reload the page to again validate and proceed.

The validation procedure in the Sign Up procedure has the following possibilities:

1. Invalid User ID
2. The user has been taken
3. The passwords inputted by the user don't match
4. Fields are empty

The validation in the Login procedure has the following possibilities:

1. Invalid user credentials
2. Invalid Password

The page to which the login/signup page direct is the reCAPTCHA page. The user selects a pattern from the given images and then proceeds to the OTP page.

In the OTP page, the user has to firstly validate his/her email and then the user receives an email having 6 digit one time password and then the user inputs the OTP in the text field and then proceeds to the webpage.

When the email is validated, the CLIENT AND THE SERVER interaction is displayed behind the text fields.

If the username is used previously in the database, it can't be used again. Once OTP is generated and used, it can't be used again.

The project is working on localhost.

The SMTP protocol works on Port 25/Port 587.

Port 25 supports SSL/TLS encryption.

Some of the systems with enhanced security won't be able to send the mail as the system will block the unauthorized and unprotected access to that port.

Port 587 doesn't have TLS encryption so the mail server can work easily.

Mailing procedure is done by using SMTP.php and PHPMailer.php where a Gmail account is used to send the emails in the backend. That Gmail id should give access to the less secure apps feature in Gmail or else the mailer won't work.
Steps to be followed while signing up for a new account:-

1. Click on the link Create Account which will lead you to sign up page.
2. Enter your valid email id in the id box provided
3. Enter your password in the box provided
4. Validate your email
5. Then proceed further.

8. Conclusion

The delivered prototype of our software demonstrates how authentication can be achieved by using three level protection/authentication. With this we can conclude that our prototype can be used for security purposes by individuals and organizations in order to prevent their data from theft. Extending this model to other resources will definitely increase the security level at its best.

9. References

1. https://www.tutorialsduniya.com
2. Google reCAPTCHA API available at Google.com
3. github.com for PHPMailer and SMTP
4. php.net/manual
5. https://www.tutorialsduniya.com