FINAL REPORT FOR GROUP PROJECT
IIT 372-2

Group No: 05
Tuition Class Management System

Industrial Information Technology
Department of Computer Science and Technology
Uva Wellassa University
February 2019

Contents

Group Details

Group No: 05

Group Members

Name	Index Number	E-mail	Contact No
G.Arunprashath	UWU/IIT/15/006	arunprashath73@gmail.com	0752293921
H.M.P.P. Jayarathana	UWU/IIT/15/021	iit15021@std.uwu.ac.lk	0713608978
N.B.L.K. Perera	UWU/IIT/15/038	laskahitha1629@gmail.com	0714799841
W.K.M.R. Walpita	UWU/IIT/15/053	menuwanrangana077@gmail.com	0717039864
M.M.K.D. Gunasinhge	UWU/IIT/15/065	mmkalpadeshan@gmail.com	0712355434

Supervisor Details

Name	E-mail	Contact No
Sanjeewanie Senanayake	senanayakeshd@gmail.com	0719138132

Chapter 1: Introduction

1. Project Title   Tuition Class Management System

2. Project Description

In the modern-day society tuition classes have become one of the major way of giving education to the students in the current society. But building an interaction between children and parents is a difficult task. It tacks considerable time to do thinks manually so, to override this problem we are going to auto mate the manual system.

In the manual system following basic functions are provided,

- ❖ Student, parent, teacher details registration.
- ❖ Teacher's Monthly class fees obtain from students.
- ❖ Student's attendance.
- ❖ Student's marks.
- ❖ Information about students are passed to the parent every six month.

But this manual process arises a number of problems.

3. Problem in Brief
- ❖ A very poor communication between parents and children and most of the time parents are not aware of their children.
- ❖ It is taken the more time on manual data record system.

4. Proposed Solution
- ❖ Parents can identify the current location and sign in and out time from the class through an SMS for students not using the smartphone, also it can be viewed by the android application.
- ❖ Each and every student's details, basic details of parents, teachers details will be recorded in the database.
- ❖ Teachers can be rated by the student or parents.

6. Objectives
- ❖ Each student sign-in and sign-out is marked by the barcode reader.
- ❖ Parents can monitor their children's results, attendance and tuition fees through the android application.

7. Methodology
- ❖ we will meet every three days and discuss the problems and combines each individual work units.
- ❖ we are using the waterfall model.

Chapter 2: Functional/ Non-functional Requirements

1. Functional requirement

Desktop application
- ❖ Mark the Attendance by Barcode reader
- ❖ SMS function-> An SMS message is sent to the parent about student's sign-time, sign-out time.

Web application
- ❖ Keep record and update record of the details of Students, Parents, and Teachers.
- ❖ Track the students sign-in time and sign-out time and location.
- ❖ Search Attendance, Class fees, student marks, location, teacher's fees.
- ❖ Add and update class fees, student marks, teacher's fees.
- ❖ Add and Update papers by teachers.
- ❖ Students can view the papers.
- ❖ The Student can rate the teachers.
- ❖ Localization (English, Sinhala).
- ❖ Captcha Function on login and registration.

Android application
- ❖ View student signs in and signs out time.
- ❖ Get the tuition class location.

2. Non-functional requirements
- ❖ Usability (Color combination is valid for even color blind people, User interface responsive to any device.)
- ❖ User friendliness – adding icons and information messages.
- ❖ Increasing the Performance. (AJAX form submit use to data insert and search, convert high resolution images suitable quality)
- ❖ Security- prevent unauthorized access and SQL injection by using security functions.
- ❖ Barcode reader is not going to take more than one minute.
- ❖ Data adding and update is not reflecting with other accounts.
- ❖ Reliability.

Chapter 3: User levels, User Roles

1. User levels
- ❖ Web admin
- ❖ Teacher
- ❖ Parent or Student
- ❖ Public User

2. User Roles

Web Admin
- ❖ Web admin can handle every student, Parent and Teacher details including insert, delete and update.
- ❖ Web admin can change their profile.
- ❖ Change the system settings
- ❖ Add and update class fees, student marks, teacher's fees
- ❖ Search the details of the student, teacher, and class.
- ❖ Add or update the rating of teachers.

Teachers
- ❖ Teachers can only update their profile.
- ❖ Can view any student's details and student's results.
- ❖ Add and update papers.

Parents or Students
- ❖ They can only view the current student's details and restrict to access or view others profile.
- ❖ They can update their own profile.
- ❖ Rating the teachers.

Public User
- ❖ Search for the teachers.
- ❖ Create a new account.
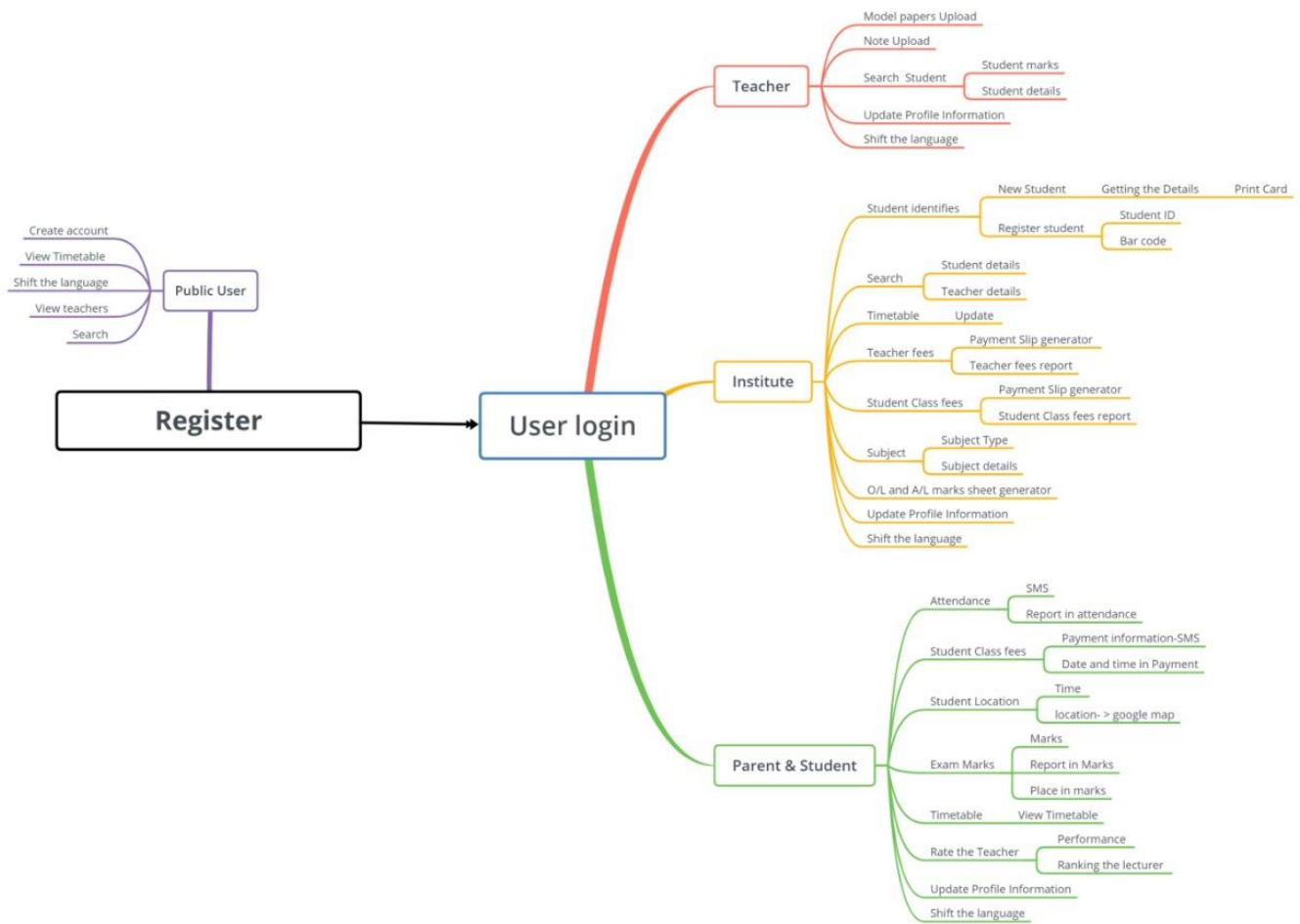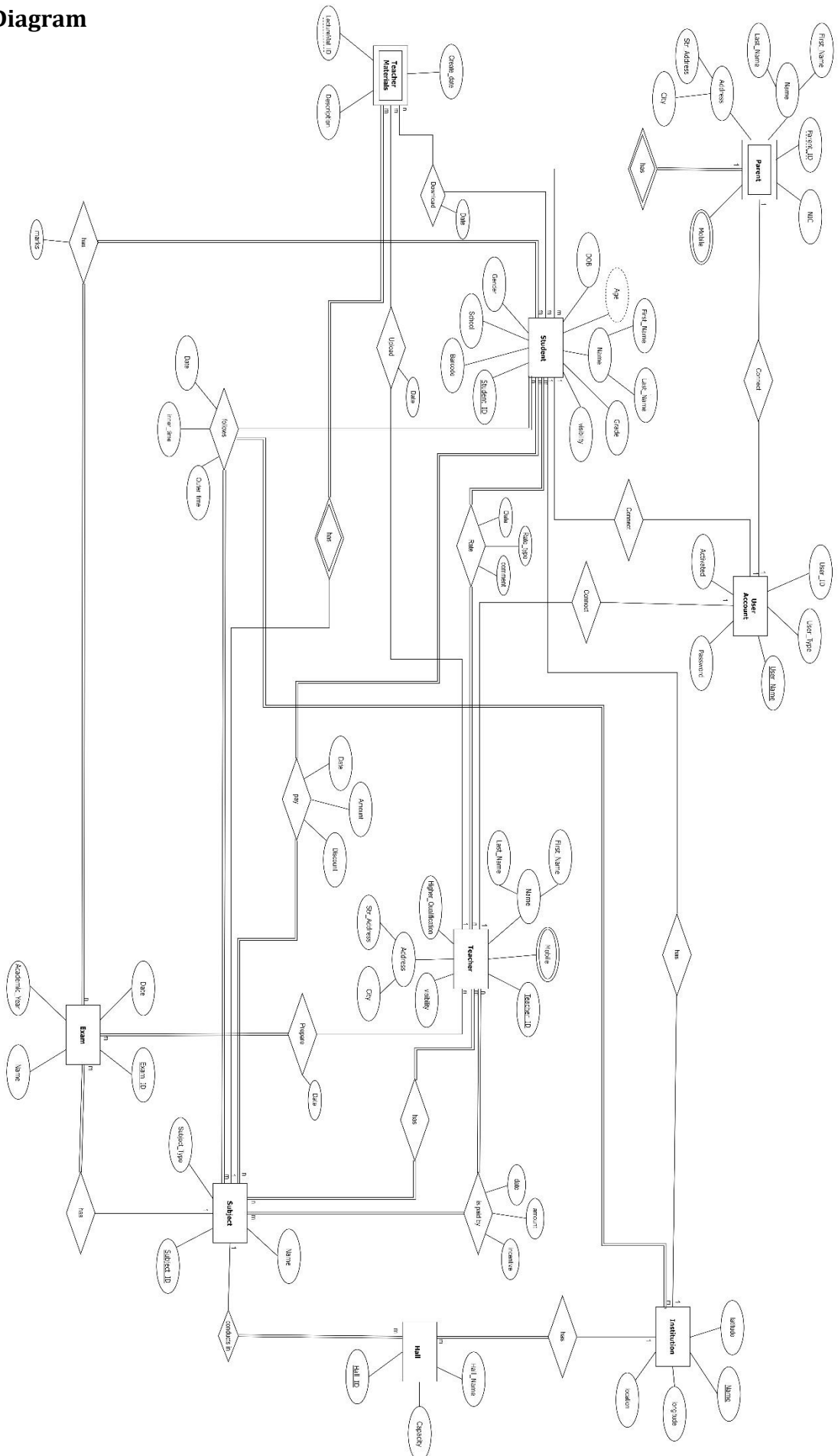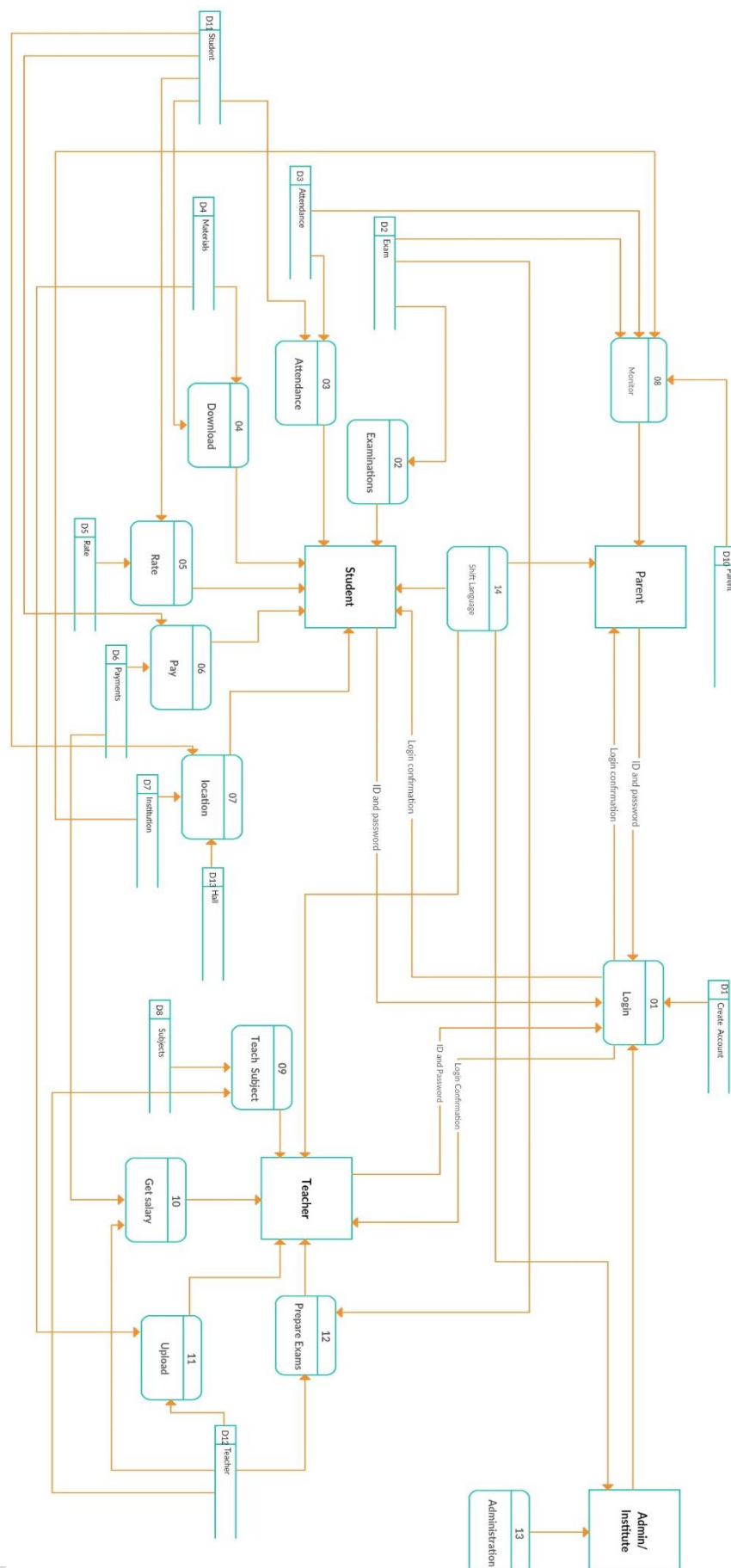- ❖ Shift the language (English or Sinhala)

SafyLite

Figure 1: System main functions, users and user roles.

SafyLite

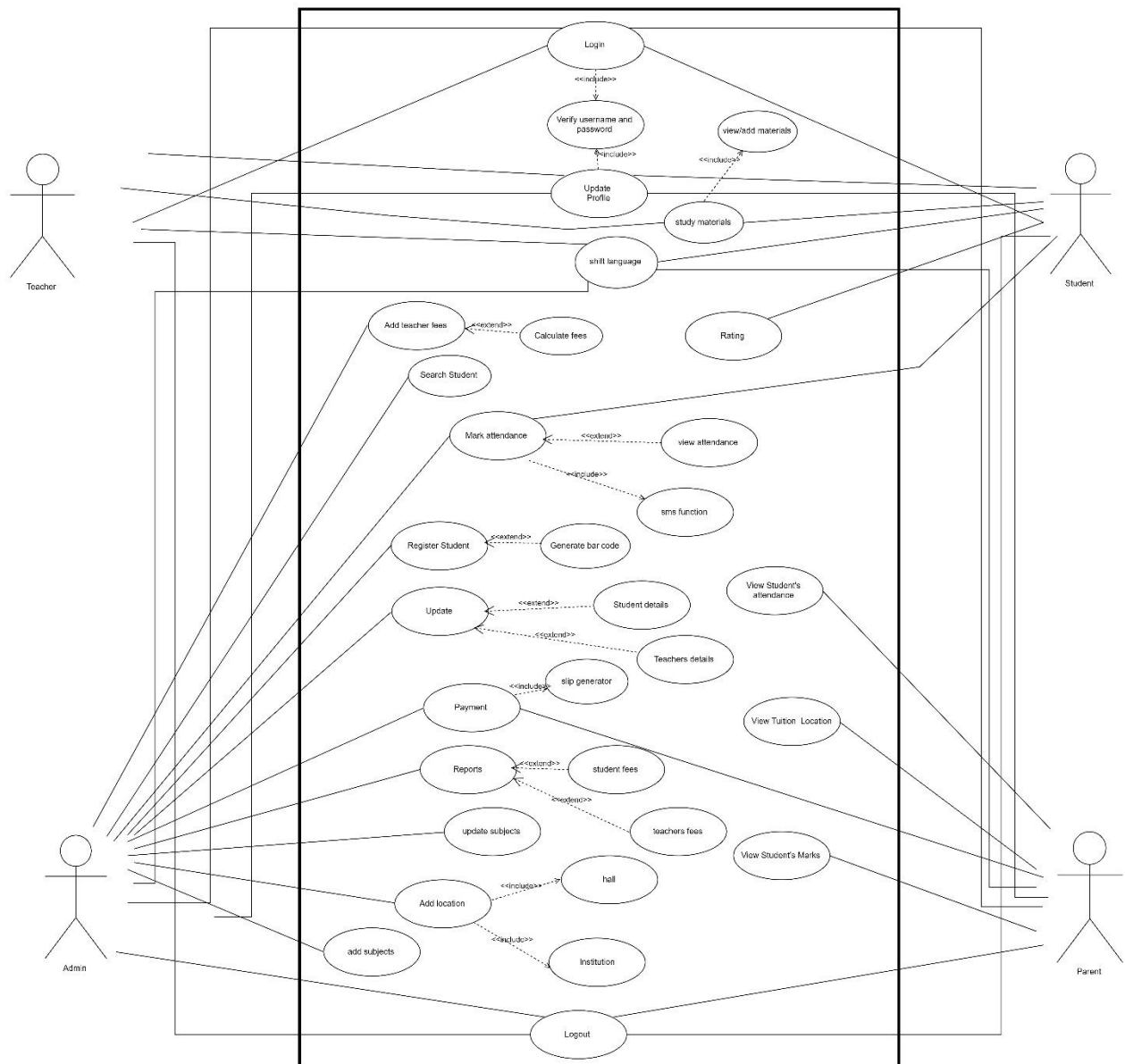# Chapter 4: System design diagram

1. ER Diagram
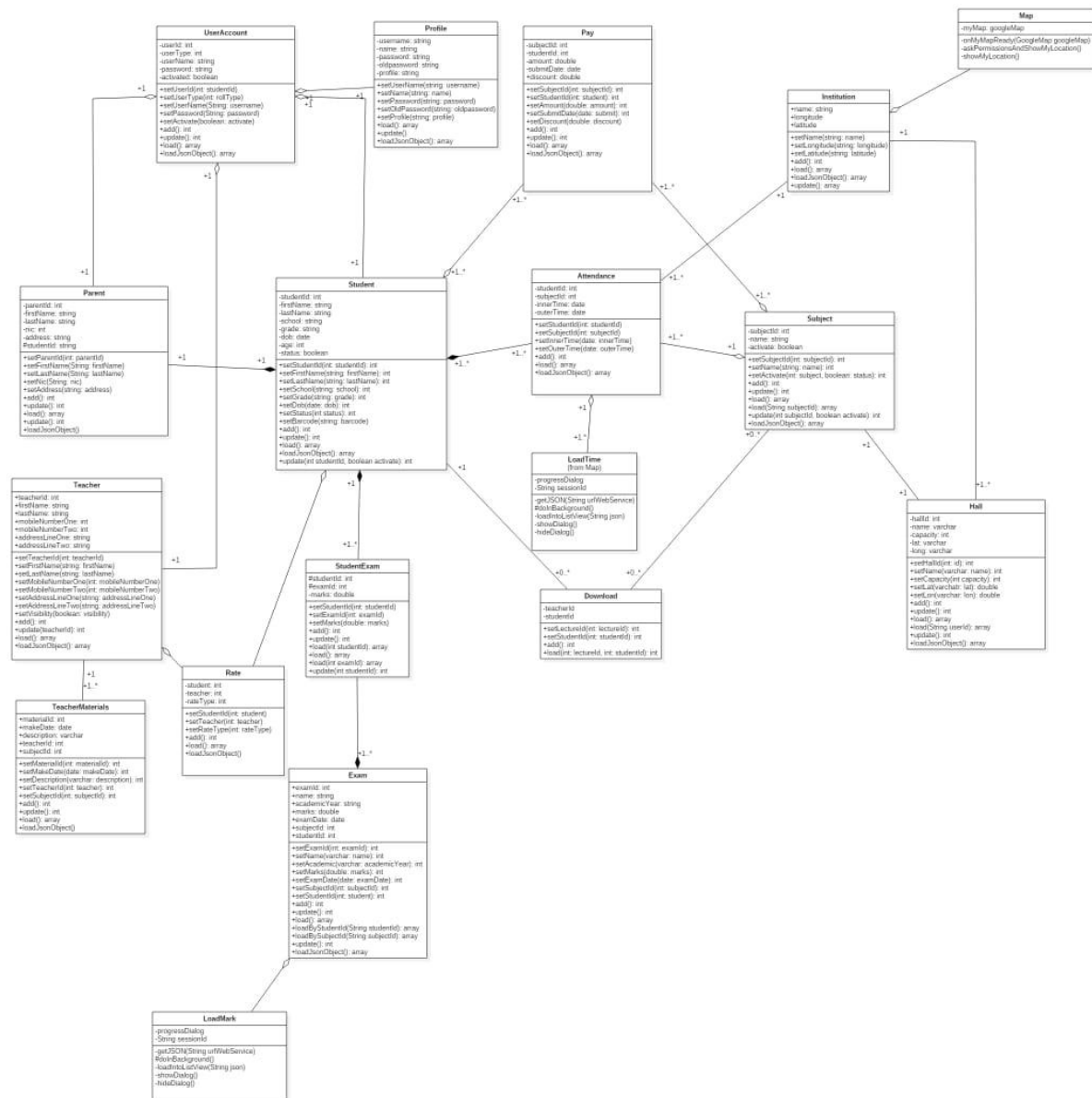
2.Data Flow Diagram

3.Use Case Diagram

4.Class Diagram

Chapter 5: Testing carried out

1.Unit Testing Level

We have used the codeIgniter framework to develop the project. That library is included the 'phpunit' library and we don't need external configuration.

Example: $this->load->library ('unit test'); (This code segment use to load php unit library)

White Box Testing

1.1

```
 public function Login($username,$password) {
        if (!isset($username) & !isset($password)) {
        return "Please enter Username and Password";
        } else {
        $this->Model_Login->setUsername($username);
        $this->Model_Login->setPassword($password);
        $this->Model_Schools->Login();

        return "Welcome ".$username;
        }
        }
        public function LoginTest() {
        echo 'Using php unit test';
        $test = $this->Login("pubudupraneeth@gmail.com", "12345678aA");
        $expected = "Welcome Praneeth";
        $testname = "Test Login";
        echo $this->unit->run($test, $expected, $testname);
        }
```

Using php unit test

Test Name	Test Login
Test Datatype	String
Expected Datatype	String
Result	Passed
File Name	C:\wamp\www\HANDME\application\controllers\Login.php
Line Number	53
Notes	

Attachment- 1

1.2

```
public function insertSubject($name,$date,$subjectname,$fees,$classType) {
        if (!isset($name) & !isset($date) & !isset($subjectname)& !isset($fees)&
!isset($classType)) {
        return "fails to insert";
        } else {
```

```
$this->Model_Subject->setName($name);
$this->Model_Subject->setDate($date);
$this->Model_Subject->setSubjectName($subjectname);
$this->Model_Subject->setFees($fees);
 $this->Model_Subject->setClassType($classType);
$this->Model_Subject->save();
return "Saved" }
}
public function addTest() {
echo 'Using php unit test';
$test = $this->insertSubject("Chemistry", "1/23/2019","1000",”AL”);
$expected = "Saved";
$testname = "Test Adding Subjects";
echo $this->unit->run($test, $expected, $testname);
}
```

Using php unit test

Test Subject	Test Subject
Test Datatype	String
Expected Datatype	String
Result	Passed
File Name	C:\wamp\www\HANDME\application\controllers\Subject.php
Line Number	53
Notes	

Attachment- 2

1.3

```
public function insertHall($Hallname,$SubjectName,$capacity) {
        if (!isset($Hallname) & !isset($SubjectName) & !isset($capacity)) {
        return "fails to insert";
        } else {
        $this->Model_Hall->setHallName($name);
        $this->Model_Hall->setSubjectName($SubjectName);
        $this->Model_Hall->setCapacity($subjectname);
        $this->Model_Hall->save();
        return "Saved";
        }
        }
        public function addTest() {
        echo 'Using php unit test';
        $test = $this->insertHall("A2", "Bio","75");
        $expected = "Saved";
        $testname = "Test Adding Halls";
        echo $this->unit->run($test, $expected, $testname);
        }
```

Using php unit test

Test Subject	Test Hall
Test Datatype	String
Expected Datatype	String
Result	Passed
File Name	C:\wamp\www\HANDME\application\controllers\Hall.php
Line Number	53
Notes	

Attachment- 3

1.4

```
public function insertLocation($lat,$long,$name) {
      if (!isset($lat) & !isset($lat) & !isset($lat)) {
      return "fails to insert";
      } else {
    $this->Model_Schools->setName($name);
      $this->Model_Schools->setLatitude($lat);
      $this->Model_Schools->setLongtitude($long);
      $this->Model_Schools->save();
      return "Saved";
      }
      }
      public function addTest() {
      echo 'Using php unit test';
      $test = $this->insertLocation("6.9934", "81.0550","Badulla");
      $expected = "Saved";
      $testname = "Test Adding Locations";
      echo $this->unit->run($test, $expected, $testname);
      }
```

Using php unit test

Test Name	Test Adding Locations
Test Datatype	String
Expected Datatype	String
Result	Passed
File Name	C:\wamp\www\HANDME\application\controllers\Location.php
Line Number	53
Notes	

Attachment- 4

2. Integration Testing Level

Our tuition management system Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing.

Black Box Testing

Positive Test Cases

01)

Project Name: Tuition management System	
PHP Web Application	
Test Case id: 01	**Test Designed by:** Praneeth
Test Priority: High	**Test Designed date:**1/23/2019
Module name: Login	**Test Executed By:**Praneeth
Test title: Verify Login Levels functions	**Test Executed date:**1/23/2019

Description: Verify Login Level functions for admin and student account

Pre-Conditions: Enable the All functions in admin and limited the functions in student account

Dependencies: Wampserver, Windows 10 OS

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Login to the Account Using admin		User should able to Login	User to navigate to	Pass	
2	Provide User name Password	Username:pubudupraneeth@gmail.com				
3	Provide Password	Password:12345678aA		login		
4	Click Login Button					

Post Conditions:
   User is validated with database and successful login into the user account page.

02)

Project Name: Tuition management System	
PHP Web Application	
Test Case id: 2	**Test Designed by:** Lakshitha
Test Priority: Medium	**Test Designed date:** 1/20/2019
Module name: Add Hall details	**Test Executed By:** Lakshitha
Test title: Verify add Hall details using access levels.	**Test Executed date:** 1/20/2019
Description: Test the Tuition management system Add Hall details functions for admin account	
Pre-Conditions: Add the Hall details use in admin account	
Dependencies: Wampserver, Windows 10 OS	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Click the add Button		User can be adding the new hall. show the message successfully added.	Adding new hall use in Admin.	pass	
2	Provide Hall name	Hall name: A2				
3	Provide Subject name	Subject name : Bio				
4	Provide Hall capacity	Hall capacity: 75				
5	Click Add Button					

Post Conditions:
    Admin can add hall details successfully.

03)

Project Name: Tuition management System	
PHP Web Application	
Test Case id: 3	**Test Designed by:** Lakshitha
Test Priority: High	**Test Designed date:**1/21/2019
Module name: Upload study materials	**Test Executed By:** Lakshitha
Test title: Verify add study materials using access levels.	**Test Executed date:**1/21/2019
Description: Verify add study materials functions for admin and teacher account	
Pre-Conditions: Add the study materials use in admin and teacher account	
Dependencies: Wampserver, Windows 10 OS	
Post Conditions: Admin and teacher can add study materials successfully.	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/ Fail)	Notes
1	Add study materials in admin and teacher account		User can be adding the study materials. show the message successfully added.	Adding study material	Pass	
2	Provide Document name	Document name: Accounting past paper				
3	Provide File path	File path: C:\Users\HP\Documents\Acc.pdf				
4	Provide Document description	Document description: Accounting mid paper-2015				
5	Click SaveButton					

04)

Project Name: Tuition management System						
PHP Web Application						
Test Case id: 4			**Test Designed by:** Arun			
Test Priority: High			**Test Designed date:**1/23/2019			
Module name: Subject Details			**Test Executed By:** Arun			
Test title: Verify Subject level functions			**Test Executed date:**1/23/2019			
Description: Verify Subject level functions (Add,Update,Delete)						
Post Condition : Validate the Subject details and add ,update,delete the records						
Pre - Condition: users have valid information the subject details Add ,Update,Delete						
Dependencies: Wampserver, Windows 10 OS						
Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/ Fail)	Notes
1	Click the Subject page		Users able see the Subject Page	Connect the correct page reference.	Pass	
2	Add Subject details ,Update Subject details ,delete subject details	TN:Kumar SN:Science Date:12/01/2019, fees:2000/ CT:A/L	User can see the Success Message	All new records Successfully.	Pass	
3	Update Subject details ,Update Subject details ,delete subject details	TN:Herath SN:Maths Date:13/02/2020, fees:2000/ CT:O/L	User can see Successfully added Message	Input details has Successfully added into Database.	Pass	
4	Delete the existing subject records	TN:Kumar SN:Science Date:12/01/2019 fees:2000/ CT:A/L	Show Successfully deleted message to users	Input details has Successfully deleted the existing record form the database.	Pass	

Negative Test Cases

01)

Project Name: Tuition management System	
PHP Web Application	
Test Case id: 5	**Test Designed by:** Lakshitha
Test Priority: High	**Test Designed date:**1/21/2019
Module name: Upload study materials	**Test Executed By:** Lakshitha
Test title: Verify add study materials using doesn't access Levels.	**Test Executed date:**1/21/2019
Description: Verify add study materials functions for student account	
Pre-Conditions: Add the study materials use in student account	
Dependencies: Wampserver, Windows 10 OS	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Try to add study materials in student account		Disable the adding function. download only	User can download study martials only.	Pass	

Post Conditions:
   Student can't add study materials. Student can download the student materials.

02)

Project Name: Tuition management System	
PHP Web Application	
Test Case id: 6	**Test Designed by:** Menuwan
Test Priority: Medium	**Test Designed date:** 1/20/2019
Module name: Add Hall details	**Test Executed By:** Menuwan
Test title: Verify add Hall details using doesn't access Levels.	**Test Executed date:** 1/20/2019
Description: Test the Tuition management system Add Hall details functions for student account or teacher account	
Pre-Conditions: Add the Hall details use in student account	
Dependencies: Wampserver, Windows 10 OS	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Click the add Button		Disable the adding function, Enable the view only.	User can view hall details only.	pass	
2						
3						
4						
5						

Post Conditions:
   Student can't add hall details. Student can download the hall details.

03)

Project Name: Tuition management System	
PHP Web Application	
Test Case id: 7	**Test Designed by:** Lakshitha
Test Priority: Middle	**Test Designed date:**1/21/2019
Module name: Add exam details	**Test Executed By:** Lakshitha
Test title: Verify add exam details using doesn't access Levels.	**Test Executed date:**1/21/2019
Description: Verify add exam details functions for student account	
Pre-Conditions: Add the exam details use in student account	
Dependencies: Wampserver, Windows 10 OS	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Try to add exam details in student account		Disable the adding function, view only.	User can view exam details only.	Pass	
2						

Post Conditions:
Student can't add exam details. Student can view the exam details.

04)

Project Name: Tuition management System	
PHP Web Application	
Test Case id: 8	**Test Designed by :** Kalpa
Test Priority: High	**Test Designed date:**1/23/2019
Module name: Subject level functions	**Test Executed By:** Kalpa
Test title: Verify Subject level functions	**Test Executed date:** 1/23/2019
Description: Verify Subject level functions (Add,Update,Delete)	
Post Condition : Validate the invalid Subject details and add ,update,delete the records	
Pre - Condition: users have invalid/Incorrect information to check negative cases the subject details (Add ,Update,Delete)	
Dependencies: Wampserver, Windows 10 OS	

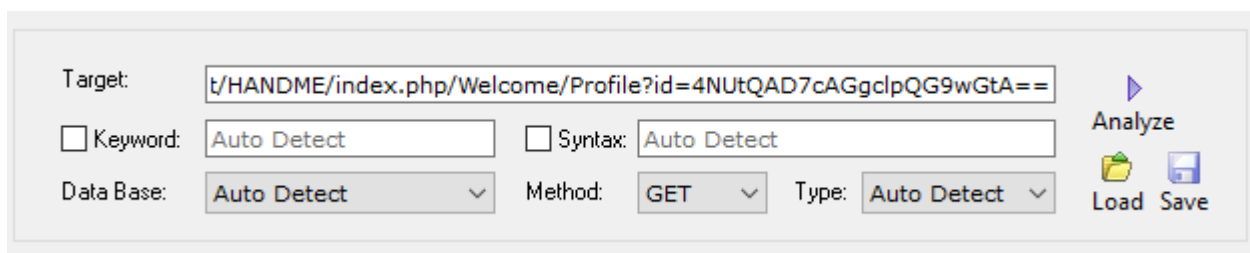| Step	Test Steps	Test Data	Expected Result	Actual Result	Status(Pass/Fail)	Notes
1	Click the Subject page		No event will occur, keep the home page.	Not link with the Subject page.	Fail	
2	Add Subject details ,Update Subject details ,delete subject details	TN:Lakshi SN:Civil Date:16/01/95 fees:9000 CT:Gr 6	User can see Error /Failure/pop up messages.	No records save in the database.	Fail	
3	Update Subject details ,Update Subject details ,delete subject details	TN:Pali SN:OS Date:16/01/2019 fees:7000 CT:Gr 7	User can see Error/failure/ Message.	Input details has not Successfully added into Database.	fail	
4	Delete the existing subject record	TN:Leon SN:Politic Date:10/01/2017 fees:5000 CT:Gr 9	User can see Error/failure/ Message.	Input details has not Successfully deleted the existing records form the database	Fail	

3. System Testing Level

Black Box Testing

System Test Case: Load Testing

Project Name: Tuition management System	
PHP Web Application	
Test Case id: 9	Test Designed by : Praneeth
Test Priority: High	Test Designed date:1/23/2019
Module name: Login	Test Executed By: Praneeth
Test title: Load Testing	Test Executed date: 1/23/2019

Description: Testing How much Capacity can hold the system

Pre Condition : Send One Hundred Student Searching Request in one Minutes using the 'Apache JMeter'

Dependencies: Wampserver, Windows 10 OS, Apache JMeter

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass /Fail)	Notes
1	Navigate to the Student Search page		Working fine after the Load Testing	Working fine	Pass	

Post Conditions: System can be hold more than 100 request within a minute.

SafyLite

System Test Case : Security Testing

Project Name: Tuition management System	
PHP Web Application	
Test Case id: 10	**Test Designed by :** Praneeth
Test Priority: High	**Test Designed date:** 1/23/2019
Module name: Student Account	**Test Executed By:** Praneeth
Test title: Security Testing	**Test Executed date:** 1/23/2019

Description: find vulnerability conditions in the User account page						
Post Condition : Determine is it possible to perform sql injections to the Student account page.						
Dependencies: Wampserver, Windows 10 OS, Havij v1.6						
Step	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**	**Notes**
1	Login as a Student	http://localhost/HANDME/index.php/Welcome/Profile?id=4NUtQAD7cAGgclpQG9wGtA==	Target Not Vulnerable	Target Not Vulnerable	Pass	
	Enter Username	Praneeth				
	Enter Password	12345678dA				

Post Conditions:
   After Login as a student some parameters pass to the account page. But using this parameters cannot perform SQL injections

Status: **I'm IDLE**   Clear Log

```
Finding columns count(MySQL,MsSQL 2005): 32
Error: 302 Found
Cannot find column count!
Target Not Vulnerable :(
It seems that input parameter is not effective! Check the following:
        Are you sure input parameter really exist?!
        Are you sure the input value '4NUtQAD7cAGgclpQG9wGtA==' is valid?
        Are you sure the 'GET' method is correct?
```

Attachment- 5

System Test Case: Usability Testing

Project Name: Tuition management System	
PHP Web Application	
Test Case id: 11	**Test Designed by:** Lakshitha
Test Priority: High	**Test Designed date:**1/23/2019
Module name: Exam function	**Test Executed By:** Lakshitha
Test title: Usability Testing	**Test Executed date:** 1/23/2019
Description: Testing most popular web browsers(Firefox,Chrome,Opera) support to the system and color matching in the System.	
Post Condition: Open the several web browsers and using color blind app to matching colors (RGBlind app).	
Dependencies: Wampserver, Windows 10 OS, RGBlind app, web usability plugin	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to the Exam details using several web browsers.		Working fine several web browsers.	Working fine.	Pass	

| 2 | Navigate to the Exam details using color blind app. | | Working fine the system Simulate color blindness. (Normal vision, Protanopia, Deuteranopia) | Working fine. | Pass | |

Post Conditions:
System can be work all web browsers correctly and colors also match in the system.

System Test Case : Performance Testing

Project Name: Tuition management System	
PHP Web Application	
Test Case id: 12	**Test Designed by :** Arun
Test Priority: High	**Test Designed date:** 1/23/2019
Module name: Student	**Test Executed By:** Arun
Test title: Performance Testing	**Test Executed date:** 1/23/2019
Description: Adding or Update data within a particular time period (milliseconds).	
Post Condition : System can be hold more than 100 request within a minute.	
Dependencies: Wampserver, Windows 10 OS, Web development tool bar	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Enter Subject ID,name Teachername , fees and date,Grade. And click add button	TN:Kumar SN:Civil Date:16/01/1995 fees:9000 CT:Gr 6	Not more than 10 second to complete saving process	201 Ms.	Pass	

Post Condition: I checked in the Performance in the wamp server, but the result can be varied among other servers.

Attachment- 6

4. UA (User Acceptance) Testing Level

Black Box Testing

1)

Project Name: Tuition management System	
PHP Web Application	
Test Case id: 13	**Test Designed by:**Praneeth
Test Priority: Medium	**Test Designed date:**1/23/2019
Module name: Rating Teachers	**Test Executed By:**Praneeth
Test title: Successfully Rating the Teachers	**Test Executed date:**1/23/2019
Description: As a student, I want to rate the teachers by their performance	
Pre-Conditions: Login as a student, search teachers and rating.	
Dependencies: Wampserver, Windows 10 OS, Firefox v25	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/ Fail)	Notes
1	Login as a Student					
2	Add username	Username:st 001				
3	Add Password	Password:12 345678dD				
4	Click Login Button		Showing successful Login message and Redirected to the Student account page	Redirected to the Student account page	Pass	
5	Navigate to Rating page					

6	Search Teacher name	Teacher name: Dr.Asiri Vanigasekara					
7	Select Performance of teacher	Select Confident:5					
		Interesting of Teaching Style:5					
		Punctuation of TeachingStyle:5					
		Satisfacations :5					
8	Click Set Rate Button		Increase the percentage	Percentage increased as 2%	Pass	Data was accumulating over the existence of teachers

Post Conditions:

User's Who admin can add location into the system successfully.
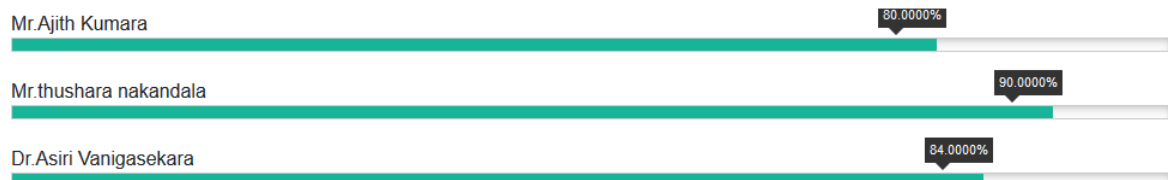
Attachment- 7

2)

Project Name: Tuition management System	
Desktop java Application	

Test Case id: 14	Test Designed by: Arun
Test Priority: Medium	**Test Designed date:**1/23/2019
Module name: Subject Details	**Test Executed By:** Arun
Test title: Successfully Add and View payment details.	**Test Executed date:**1/23/2019

Description: Login as a student, and I want to add and view my payment details
Post-Condition :
Pre-Conditions: I should have/know correct details about my subjects to input
Dependencies: Wampserver, Windows 10 OS, Firefox v25

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Login as a Student					
2	Add username	Username:st001				
3	Add Password	Password:12345678d				
4	Click Login Button		Showing successful Login message enter to the home page	Redirected to the Home page	Pass	
5	Click payment button		Open the payment window.	Open the jFrame for Payment	Pass	
6	Add teacher Name,	TN:Kumara SN:Science Date:16/01/2019	User should able to input data into the fields	the query take correct values from	Pass	

	Subject Name,date, Fees,and Class type	Fees: 5000 CT: A/L		the user input		
7	Click add button		Pop up message show as Successfully added.	Data saved in the database	Pass	
8	Click view button		Open the Payment view window	Retrieving the stored data via query from DB.	Pass	

3)

Project Name: Tuition management System	
PHP Web Application	
Test Case id: 15	Test Designed by: Lakshitha
Test Priority: High	Test Designed date: 1/22/2019
Module name: Hall add and update function	Test Executed By: Lakshitha
Test title: Successfully adding and updating the Hall details.	Test Executed date: 1/22/2019
Description: As an Admin, I want to add the hall details	
Pre-Conditions: Login as An Admin, add and update Hall details.	
Dependencies: Wampserver, Windows 10 OS, Firefox v25	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Not es
1	Login as an Admin					

2	Add/Update Hall name	Hall name: B1				
3	Add/Update Subject name	Subject name: Bio				
4	Add/Update Hall capacity	Hall capacity: 75				
5	Click Add/Update Button		Showing successful Add/Update message and Redirected to the Hall Details page	Redirected to the Hall details page	Pass	
6	Navigate to Hall details table					
7	Search Hall name	Hall name: B1				
8	Click Search Button		Find the Hall details B1	Find adding details.	Pass	

Post Conditions:

Admin can add new hall into the system successfully.

4)

Project Name: Tuition management System	
PHP Web Application	
Test Case id: 16	**Test Designed by:** Lakshitha
Test Priority: High	**Test Designed date:**1/22/2019
Module name: Study materials add function	**Test Executed By:** Lakshitha
Test title: Successfully adding the Study materials.	**Test Executed date:**1/22/2019
Description: As a teacher, I want to add the Study materials.	

Pre-Conditions: Login as a teacher, add and update Study materials.

Dependencies: Wampserver, Windows 10 OS, Firefox v25

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Login as an Teacher					
2	Add Document name	Document name: Accounting paper				
3	Add File path	File path; C:\Users\HP\Documents\Acc.pdf				
4	Add Document description	Document description: Accounting Paper 2014				
5	Click Add/Update Button		Showing successful Add message and Redirected to the study material page	Redirected to the study material page	Pass	
6	Navigate to study material table					
7	Search Document name	Hall name: Accounting paper				
8	Click Search Button		Find the study material Accounting paper	Find adding details.	Pass	

Post Conditions:
   Teacher can add new study material into the system successfully

Chapter 6: Individual Contribution

Members	Functions
UWU/IIT/15/006	**Desktop application –** ❖ Add, view, search teacher and student payment. **Web application -** ❖ Add, Update and Search Subject.
UWU/IIT/15/021	**Desktop application –** ❖ Print Card for new students, Print Slip after paid student's & teacher's payment. **Web application -** ❖ Rate the Teachers. ❖ Add location of institute. **Android application –** ❖ location, Attendance, view marks.
UWU/IIT/15/038	**Desktop application –** ❖ Scan Barcode of the students when enter to class and leave the class. ❖ Check student eligible or not eligible they attend to the class. **Web application -** ❖ Add, update and Search institute halls. ❖ Upload and download study material.
UWU/IIT/15/053	**Web application -** ❖ Add, Update and Search teacher details. ❖ Add, Update and Search Exam details. ❖ Add, Update Admin profile.
UWU/IIT/15/065	**Web application –** ❖ Search Student details. ❖ Localization.