**Group Documents**

**DEVELOPMENT TEAM:**

* **BOB NKONJELA**
* **CHITALU ILUNGA**
* **CHIKWE MULUWE**
* **GLORIA MUTEMI**

**IDENTIFYING ENTITIES**

**Initiation entities**

* Course
* Course code
* Faculty
* Email
* users
* Administrator

**Final entities**

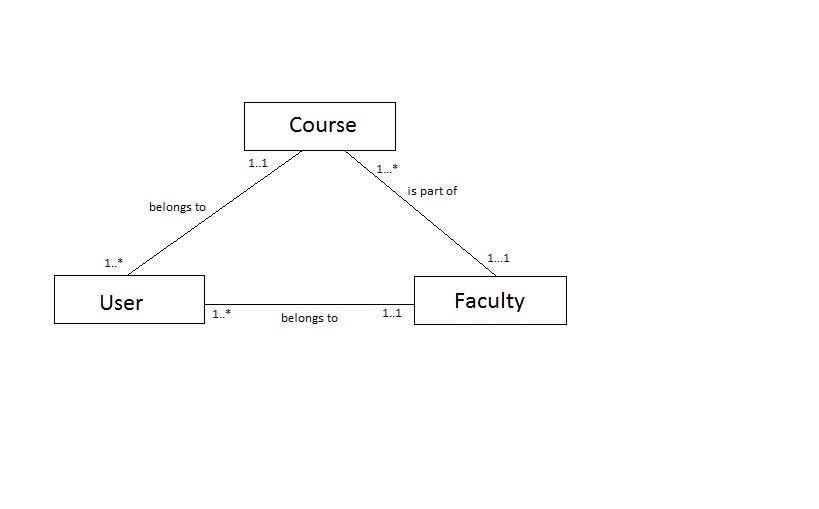
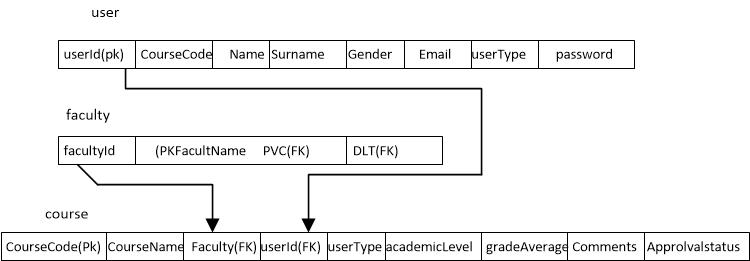
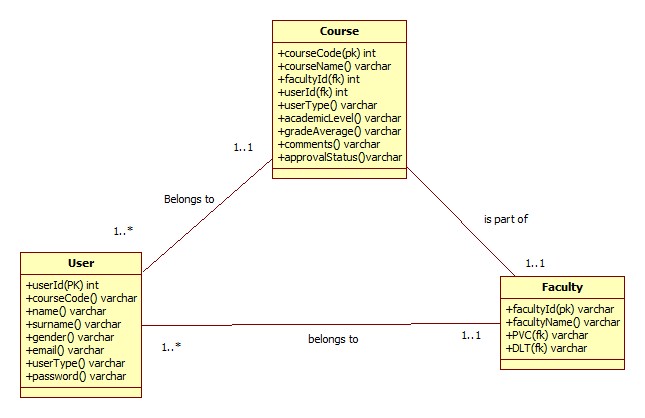
* User
* Faculty
* Course

**IDENTIFYING RELATIONSHIPS**

|  |  |  |
| --- | --- | --- |
| **Entity** | **Relationship** | **Entity** |
| User | Belongs to | Course |
| User | Lectures | Course |
| User | Leads | Course |
| Course | Belongs to | Faculty |
| User | Belongs to | faculty |
| User | Directs | faculty |

**Multiplicity**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity** | **Multiplicity** | **Relationship** | **Multiplicity** | **Entity** |
| user | 1 | Belongs to | 0...\* | Course |
| User | 1 | Lectures | 0...\* | Course |
| User | 1 | Leads | 1...1 | Course |
| course | 1..\* | Belongs to | 1 | Faculty |
| User | 1 | Belongs to | 1 | Faculty |
| User | 1 | Directs | 1..\* | Faculty |



1st Meeting Minutes

6th march 2016

Time:11:30 hrs -12:30

Attendees: Chitalu Ilunga, Chikwe Mulilwe, Bob Nkonjela, Gloria D Mutemi

Agenda : group orientation

* Got acquainted with each other
* Exchanged numbers
* Build team sprint
* Agreed on the frequency on minutes

**2nd Meeting Minutes**

10th march 2016

Time: 11:30- 12:30

Attendees: Chitalu Ilunga, Chikwe Mulilwe, Bob Nkonjela, Gloria D Mutemi

Agenda coming up with a project plan and scheduling

* Timeframe:
* Plan was drawn and refined
* Chitalu was assigned to finalise the plan and email copies crew
* Next meeting scheduled for the 12th march 2016
* Meeting closed

**3rd Meeting Minutes**

12th march 2016

Time: 11:30-12:30

Attendees : Chitalu Ilunga, Chikwe Muliliwe, Bob Nkonjela, Gloria D Mutemi

Agenda: Database, analysis, design and implementation

* Reviewing minutes from previous meeting
* Brainstorming of the system’s specifications.
* Assumptions and justifications made.
* Lists of entities where identified.
* Entities identified.
* Designing of the conceptual, relational models and implementation..
* Came to the end of the meeting, next to be scheduled for 15th march.

Goals: Complete analysis, Conceptual model, Logical schema, Entity relationship diagram

Process; Normalisation, Chens notation.

End of meeting.

**4th Meeting Minutes**

15th march 2016

Time: 11:30-12:30

Agenda database enhancement

* Reviewing minutes from previous meeting.
* Entity relationships reviewed and approved.
* Implementation of the database was done.
* Next meeting scheduled for

Goal: to build a database

Process: MySQL

End of meeting.

**5th Meeting Minutes**

11th April 2016

Time: 11:30- 12:30

Agenda: Web Design interfaces

* Reviewing of last meeting
* We discussed and agreed on how the interface should look like

**6th Meeting Minutes**

12th April 2016

Time 11:30- 12:30

Attendees: Chitalu Ilunga, Chikwe Muliliwe, Bob Nkonjela, Gloria D Mutemi

Agenda: Log notifications

**7th Meeting Minutes**

13th April 2016

Time: 11:30- 12:30

Attendees: Chitalu Ilunga, Chikwe Muliliwe, Bob Nkonjela, Gloria D Mutemi

Agenda: Email notification

* Reviewing minutes from the last meeting
* Brainstorming user specifications for email notifications
* Discussed on how to implement the emailing system
* Whether to send to the CMR from the server rather than create an email to email application
* Came up with an email notification strategy once accounts were created.
* End of meeting

**8th sprint**

14th April 2016

Time: 11:30- 12:30

Attendees: Chitalu Ilunga, Chikwe Muliliwe, Bob Nkonjela, Gloria D Mutemi

Agenda: Testing

* Creating Test Plan.
* Test log who to carry out what test and what document result.
* Test strategies

**Assumptions**

A number of assumptions were made during the requirements analysis phase, below are the following

* All users of the state university system will log in the system by using their login credentials.
* There will be 3 faculties with a guest account.
* Viewing of reports is done on the course monitoring reports.
* Two types of staff will have access to the system eg. Staff members and students.
* It was assumed that an administrator could also be a lecture and perform all duties of a lecture, but a lecture cannot perform the functions of an administrator
* It has been assumed that when course leader submits a course monitoring report , the system emails a copy to the cm for approval
* Once cmr has been approved by course moderator, a copy of the course monitoring report is also sent to the PVC and DLT
* The Course monitoring report is stored on a database
* The name of the university using the course monitoring reports web based application is called Lsk State University
* It was assumed when the DLT response is submitted a copy of the CMR is sent to the CL and CM of the course, and a copy is also sent to the PVC and DLT of the Faculty.
* All CMRs are commented on by the DLT within 14days
* Each course monitoring report belongs to one course.
* Users that don’t belong to a course have their own course code.
* There will be 3 faculty members with a guest account.
* Overdue CMRs will be classified as expired.
* CMRs will not have their own table. They will be integrated with the course table.

**1st Sprint**

Process- Normalisation and Chen’s notation

|  |  |
| --- | --- |
| **SPRINT No. 2** | |
| **Sprint Goal** | to analyze ,design and implement a database |
| **Sprint Back log** | System analysis, database design conceptual, logical modeling, database implementation and insertion and retrieval. |
| **Sprint Review** | All backlogs requirements were met, normalization was applied to data anomalies |
| **Sprint Retrospective** |  |

**2nd Sprint**

|  |  |
| --- | --- |
| **SPRINT No. 2** | |
| **Sprint Goal** | Designing and implementing database for the eSupervision system that will meet user needs of the system. |
| **Sprint Back log** | **Conceptual Modelling Design**   * Identifications of entities * Identifications of relationships among entities * Identifications of attributes of entities * Determination of attribute domains * Specifying the constraints on the entities and relationships * Creating an entity relationship diagram   **Logical Design**   * Mapping of the entity relationship diagram into a relational schema * Validation of relations using normalization * Check integrity constraint   **Physical Data Model**   * Creation of tables for the database * Design of general contraints.   **Database insertion and Retrieval** |
| **Sprint Review** | The database was designed and implemented according to the sprint goals |
| **Sprint Retrospective** | The sprint was well executed by the team members though the scrum team members had some different divergent views on creating an entity relationship diagram, but in the end we reached a consensus and implemented the ERD |

**3rd Sprint**

**Web Design**

|  |  |
| --- | --- |
| **SPRINT No. 3** | |
| **Sprint Goal** | **Designing of the User Interface** |
| **Sprint Back log** | * Design a secure role based login form and homepage * Create web page containing a form for faculty * Create web page showing A guest account for each Faculty can be used to view approved CMRs and to see statistical reports and exception reports. * Create web page for All CMRs be commented on by the DLT for the Faculty within 14 days * Create web page containing a form for bulk allocation of students to their supervisor and second marker * Create personal dashboards for each student * Create web page containing a form for messaging * Create web page containing a form for uploading documents * Creating web page containing a form for creating events * Design a complete Site Map for the web site |
| **Sprint Review** | Some of the goals were not met by the Scrum team due to time constraints |
| **Sprint Retrospective** | Creating the dashboards for all users was a challenge however the users where able to login to the system and perform other functionalities of the system. |

**4th Sprint**

Validation

Input validation results

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint Goal | To validate user input |  |  |
| Sprint Backlog | User validation |  |  |
| Sprint Review | Security functionality |  |  |
| Sprint retrospectives | Backlog requirements were mate |  |  |

**5th sprint**

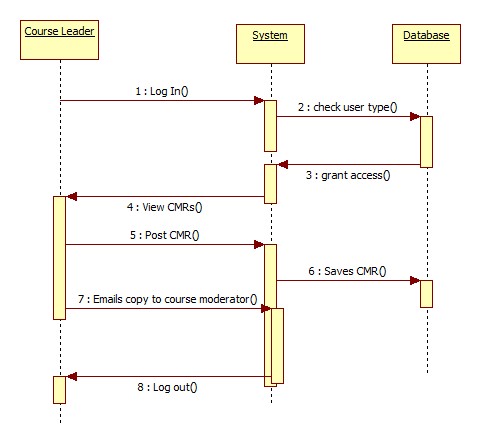
Php programming and all its features

Functional system

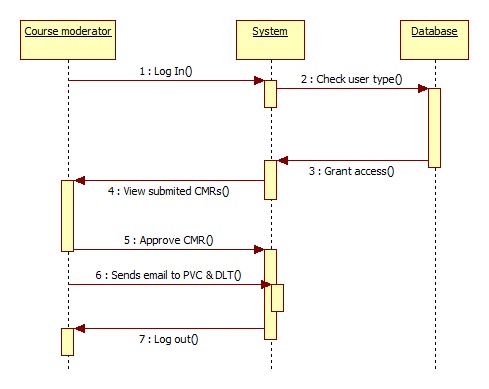
|  |  |
| --- | --- |
| Sprint goal | Implement role based user functionality |
| Sprint backlog |  |
| Sprint review |  |
| Sprint retrospective |  |

Sequence Diagrams

CL sequence diagram

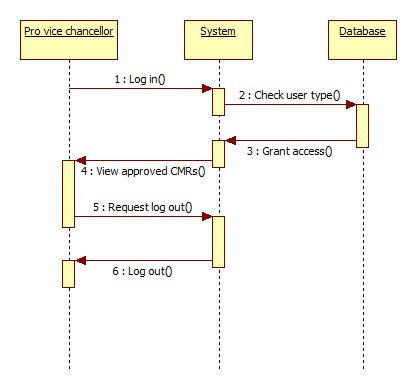


CM diagram

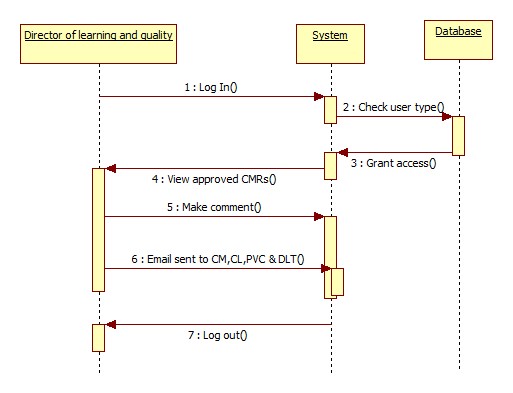


PVC diagram

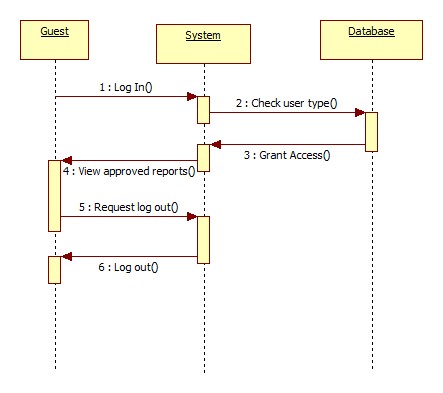
\



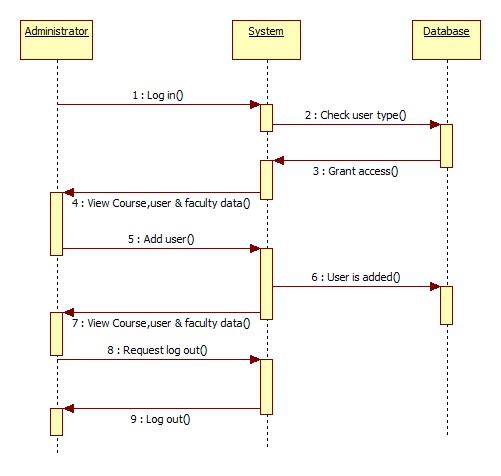
DLT diagram



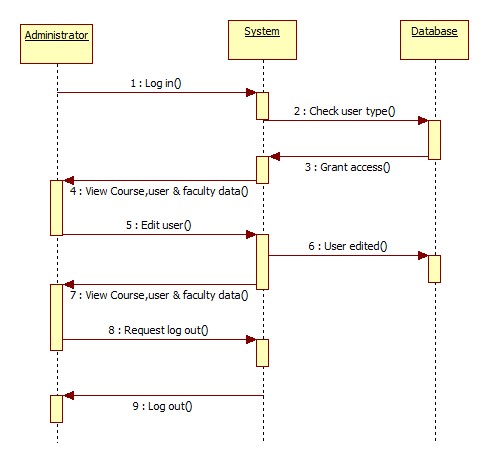
Guest diagram



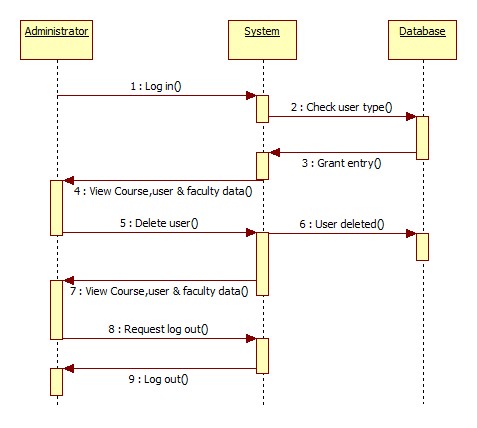
Admin add user diagram



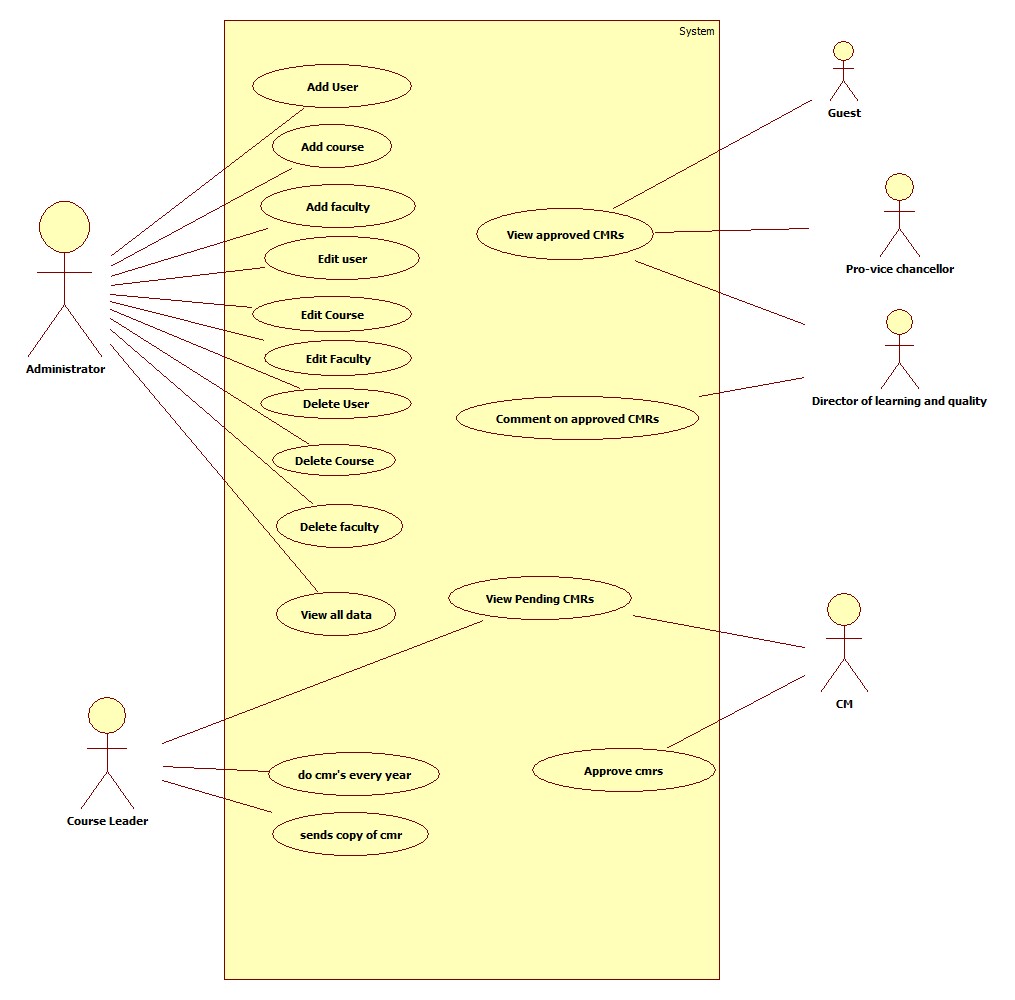
Admin edit user



Admin delete user



# Use Case Diagram:



# 

# Test Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Black Box Test 1 | | Test Element: index.php | Designed by: Chitalu Wedex Ilunga | |
| Data source: Database: ‘University’ -> Table:’users’ | | Objective: Test Administrator login functionality. | Tester: Chitalu Wedex Ilunga | |
| Test Case | Description | Tasks | Expected Result | Actual Result |
| 1.1 | Test Administrator login functionality | * Enter Administrator username and password. * Click login button. | Be securely Redirected to Administrator page. (**adminHome.php**) | Securely redirected to administrator page. |

Fig 1.1 the above results are viewable in the ‘Administrator functions’ screencast available in the group repository.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Black Box Test 2 | | Test Element: adminHome.php | Designed by: Chitalu Wedex Ilunga | |
| Data source: user input | | Objective: Test Administrator user creation functionality. | Tester: Chitalu Wedex Ilunga | |
| Test Case | Description | Tasks | Expected Result | Actual Result |
| 1.2 | Test Administrator user creation functionality. | * Click add user button. * Select user course. * Type name and surname. * Select user gender. * Type user email address. * Select user type. * Assign user password. * Click add user button. | Have user added to the user list and database with information entered | User added to the user list and database with information entered |

Fig 1.2 the test results are viewable in the ‘Administrator functions’ screencast available in the group repository.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Black Box Test 3 | | Test Element: adminHome.php | Designed by: Chitalu Wedex Ilunga | |
| Data source: user input | | Objective: Test Administrator course creation functionality. | Tester: Chitalu Wedex Ilunga | |
| Test Case | Description | Tasks | Expected Result | Actual Result |
| 1.3 | Test Administrator user creation functionality. | * Click add course button. * Type course code. * Type course name. * Select faculty. * Type course user ID. * Select user type. * Assign user password. * Click add course button. | Have course added to the course list and database with information entered. | Course added to the course list and database with information entered. |

Fig 1.3 the test results are viewable in the ‘Administrator functions’ screencast available in the group repository.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Black Box Test 4 | | Test Element: adminHome.php | Designed by: Chitalu Wedex Ilunga | |
| Data source: user input | | Objective: Test Administrator faculty creation functionality. | Tester: Chitalu Wedex Ilunga | |
| Test Case | Description | Tasks | Expected Result | Actual Result |
| 1.4 | Test Administrator faculty creation functionality. | * Click add faculty button. * Type faculty ID. * Type faculty name. * Type Pro Vice-Chancellor ID. * Type Director of Learning and Quality ID. * Click add faculty button. | Have a faculty added to the faculty list and database with information entered. | A faculty added to the faculty list and database with information entered. |

Fig 1.4 the test results are viewable in the ‘Administrator functions’ screencast available in the group repository.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Black Box Test 5 | | Test Element: adminHome.php | Designed by: Chitalu Wedex Ilunga | |
| Data source: Database: ‘University’ -> Table:’users’ and user input. | | Objective: Test Administrator user information edit functionality. | Tester: Chitalu Wedex Ilunga | |
| Test Case | Description | Tasks | Expected Result | Actual Result |
| 1.5 | Test Administrator user information edit functionality. | * Click edit user link. * Type new name and surname. * Type new user email address. * Select new user type. * Assign new user password. * Click edit button. | Have updated user information added to database and user list. | Updated user information added to database and user list. |

Fig 1.5 the test results are viewable in the ‘Administrator functions’ screencast available in the group repository.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Black Box Test 6 | | Test Element: adminHome.php | Designed by: Chitalu Wedex Ilunga | |
| Data source: Database: ‘University’ -> Table:’course’ and user input. | | Objective: Test Administrator course information edit functionality. | Tester: Chitalu Wedex Ilunga | |
| Test Case | Description | Tasks | Expected Result | Actual Result |
| 1.6 | Test Administrator course information edit functionality. | * Click edit course link. * Type new course name. * Select new course faculty. * Type user ID. * Click edit button. | Have updated course information added to database and course list. | Updated course information added to database and course list. |

Fig. 1.6 the test results are viewable in the ‘Administrator functions’ screencast available in the group repository.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Black Box Test 7 | | Test Element: adminHome.php | Designed by: Chitalu Wedex Ilunga | |
| Data source: Database: ‘University’ -> Table:’faculty’ and user input. | | Objective: Test Administrator faculty information edit functionality. | Tester: Chitalu Wedex Ilunga | |
| Test Case | Description | Tasks | Expected Result | Actual Result |
| 1.7 | Test Administrator faculty information edit functionality. | * Click edit faculty link. * Type new faculty name. * Type new Pro Vice-Chancellor ID. * Type new Director of Learning and Quality ID. * Click edit button. | Have updated faculty information added to database and faculty list. | Updated faculty information added to database and faculty list. |

Fig 1.7 the test results are viewable in the ‘Administrator functions’ screencast available in the group repository.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Black Box Test 8 | | Test Element: adminHome.php | Designed by: Chitalu Wedex Ilunga | |
| Data source: Database: n/a | | Objective: Test Administrator delete user functionality. | Tester: Chitalu Wedex Ilunga | |
| Test Case | Description | Tasks | Expected Result | Actual Result |
| 1.8 | Test Administrator delete user functionality. | * Click delete user link. * Click Ok on delete user prompt. | * Selected user is deleted. | Selected user is deleted. |

Fig 1.8 the test results are viewable in the ‘Administrator functions’ screencast available in the group repository.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Black Box Test 9 | | Test Element: adminHome.php | Designed by: Chitalu Wedex Ilunga | |
| Data source: Database: n/a | | Objective: Test Administrator delete course functionality. | Tester: Chitalu Wedex Ilunga | |
| Test Case | Description | Tasks | Expected Result | Actual Result |
| 1.9 | Test Administrator delete course functionality. | * Click delete course link. * Click Ok on delete course prompt. | * Selected course is deleted. | Selected course is deleted. |

Fig 1.9 the test results are viewable in the ‘Administrator functions’ screencast available in the group repository.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Black Box Test 10 | | Test Element: adminHome.php | Designed by: Chitalu Wedex Ilunga | |
| Data source: Database: n/a | | Objective: Test Administrator delete faculty functionality. | Tester: Chitalu Wedex Ilunga | |
| Test Case | Description | Tasks | Expected Result | Actual Result |
| 1.1.1 | Test Administrator delete faculty functionality. | * Click delete faculty link. * Click Ok on delete faculty prompt. | * Selected faculty is deleted. | Selected faculty is deleted. |

Fig 1.1.1 the test results are viewable in the ‘Administrator functions’ screencast available in the group repository.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Black Box Test 11 | | Test Element: index.php | Designed by: Chitalu Wedex Ilunga | |
| Data source: Database: ‘University’ -> Table:’users’ | | Objective: Test Course Leader login functionality. | Tester: Chitalu Wedex Ilunga | |
| Test Case | Description | Tasks | Expected Result | Actual Result |
| 2.1 | Test Course Leader login functionality. | * Enter Course Leader username and password. * Click login button. | Be securely Redirected to Course Leader page. (**clhome.php**) | Securely redirected to course leader page. |

Fig 2.1 the test results are viewable in the ‘Course monitoring report handling’ screencast available in the group repository.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Black Box Test 12 | | Test Element: index.php | Designed by: Chitalu Wedex Ilunga | |
| Data source: Database: ‘University’ -> Table:’users’ | | Objective: Test Course Moderator login functionality. | Tester: Chitalu Wedex Ilunga | |
| Test Case | Description | Tasks | Expected Result | Actual Result |
| 2.2 | Test Course Moderator login functionality. | * Enter Course Moderator username and password. * Click login button. | Be securely Redirected to Course Moderator page. (**cmhome.php**) | Securely redirected to course moderator page. |

Fig 2.2 the test results are viewable in the ‘Course monitoring report handling’ screencast available in the group repository.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Black Box Test 13 | | Test Element: index.php | Designed by: Chitalu Wedex Ilunga | |
| Data source: Database: ‘University’ -> Table:’users’ | | Objective: Test Director of Learning and Quality login functionality. | Tester: Chitalu Wedex Ilunga | |
| Test Case | Description | Tasks | Expected Result | Actual Result |
| 2.3 | Test Director of Learning and Quality login functionality. | * Enter Director of Learning and Quality username and password. * Click login button. | Be securely Redirected to Course Moderator page. (**dlthome.php**) | Securely redirected to Director of Learning and Quality page. |

Fig 2.3 the test results are viewable in the ‘Course monitoring report handling’ screencast available in the group repository.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Black Box Test 14 | | Test Element: index.php | Designed by: Chitalu Wedex Ilunga | |
| Data source: Database: ‘University’ -> Table:’users’ | | Objective: Test Pro Vice-Chancellor login functionality. | Tester: Chitalu Wedex Ilunga | |
| Test Case | Description | Tasks | Expected Result | Actual Result |
| 2.4 | Test Pro Vice-Chancellor login functionality. | * Enter Pro Vice-Chancellor username and password. * Click login button. | Be securely Redirected to Pro Vice-Chancellor page. (**pvchome.php**) | Securely redirected to Pro Vice-Chancellor page. |

Fig 2.4 the test results are viewable in the ‘Course monitoring report handling’ screencast available in the group repository.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Black Box Test 15 | | Test Element: index.php | Designed by: Chitalu Wedex Ilunga | |
| Data source: Database: ‘University’ -> Table:’users’ | | Objective: Test guest. | Tester: Chitalu Wedex Ilunga | |
| Test Case | Description | Tasks | Expected Result | Actual Result |
| 2.5 | Test guest login functionality. | * Enter guest username and password. * Click login button. | Be securely Redirected to Pro Vice-Chancellor page. (**guesthome.php**) | Securely redirected to guest page. |

Fig 2.5 the test results are viewable in the ‘Guest functionality screencast’ available in the group repository.

Requirements Analysis

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement Identifier | Requirement short name | Requirement Description | Priority |
| Non-Functional Requirement | NF1 | Develop and deliver in the required time. | Should Have |
| Functional Requirement | F1 | Store faculty details. | Must Have |
| Functional Requirement | F2 | Store course details within the database. | Should Have |
| Non-Functional Requirement | NF2 | Produce a user manual to help with the use of the system. | Should Have |
| Functional Requirement | F3 | Store user details within a database. | Must Have |
| Functional Requirement | F4 | Display user data | Could Have |
| Functional Requirement | F5 | Produce exception reports. | Should Have |
| Functional Requirement | F6 | Display course data | Could Have |
| Functional Requirement | F7 | Create login system | Must Have |
| Non-Functional Requirement | NF3 | Improve website usability | Should Have |
| Non-Functional Requirement | NF4 | Improve website visibility | Could Have |
| Functional Requirement | F8 | Create administrator account. | Must Have |
| Functional Requirement | F9 | Display faculty data | Could Have |
| Non-Functional Requirement | NF4 | Train users in use of the system. | Could have. |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement Identifier | Requirement short name | Requirement Description | Priority |
| Functional Requirement | F10 | Produce Course Monitoring Reports. | Must Have |
| Functional Requirement | F11 | Send Emails between users. | Should Have |
| Functional Requirement | F12 | Produce statistical reports. | Should Have |
| Functional Requirement | F13 | Produce exception reports. | Should Have |