University of Central Missouri – CIS

Anirudh Ambatipati

Advanced Systems Project

March 27, 2017

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Authors** | **Version** | **Change Reference** |
| 03/09/2017 | Anirudh Ambatipati | 0.1 | First Version - A Draft |
| 03/11/2017 | Anirudh Ambatipati | 0.2 | Second Version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Participants**

**Authors**

|  |  |  |
| --- | --- | --- |
| **Name** | **Representing** | **Role** |
| Anirudh Ambatipati |  | Graduate Student |
|  |  |  |

**Reviewers**

|  |  |  |
| --- | --- | --- |
| **Name** | **Representing** | **Role** |
| Dr. Prasad Rudramuniyaiah |  | Professor |
|  |  |  |

Table of Contents

**Project Description**4

**Use case**5

Use Case Description5-14

Use Case Diagram14

**E-R Diagram**15

**Class diagram** 16

**Critical Process**16

**System Components and their Relationships**17

**Unit Testing**17-21

**Scenario Level Testing**21-25

**Project Description**

University of central Missouri Computer Information System program is primarily designed based on the fact that all organizations, large or small, for-profit, non-profit or charitable, multi-national rely on timely, accurate and useful information to enhance decision making. But, in the current market has a lack of people who are good at these decision making skills So, UCM has created a program that focuses on educating students how understanding the technology for an essential organizational success. UCM CIS engages students in all categories of IT industry like gathering user requirements, analyzing those requirements, designing a solution, developing software or configuring network components according to the design, testing the solution or supporting the solution.

In this project, I would like to come up with a UCM CIS website where different users of the CIS department like Director, Advisor, Student can login to the University CIS website to do some there role based functionalities and also can access the some common functionalities of the website where any user can access them if the user is registered with university. This would eradicate the use of multiple website by different users of CIS department. Instead all the users of department can access the university website by only one website. This role based mapping of website to users is handled through there login credentials provided by the University.

One of the Major Roles in the CIS department is played by Director; director has many administrative authorities over the website like creating a Student, creating an advisor, editing/adding a program to the CIS department etc. A director can create a student based on his GRE score and background criteria, once the student is added he can send a mail to the student to his/her personal mail id providing the login credentials to the student to login. All passwords are randomly generated and MD5CryptoServiceProvider encrypted. Director can add a new advisor to the system to a particular concentration.

Advisor can be added to the system by the director. The advisor can perform some functionality like fetching student information. Once he fetches the information he can check the student Pre-requisite score, GRE Score and other factors he can decide whether the student can take the course or not and he can update pre-requisite status to met or not met and enter some notes on the students status also.

Student can login to the system once got admission form the university. Director will send the student the related login credentials to the student’s personal mail id. Once the student is logged in he can change the password according to university password rules. Student will have to submit the academic code of conduct after login. Students can also submit the prerequisite exam, score will be updated to database.

This project also in-cooperates shopping or classifieds website where a user can login to it and post desired advertisement on the website and can also view and browse the already posted advertisement on website. User has to provide some contact information and description for the product he wants to add on the website as classified advert information.

**Use Case**

This section includes use cases descriptions and use case diagram.

Use Case Description:

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-1 | | |
| **Use Case Name:** | User login in to the System | | |
| **Created By:** | Anirudh Ambatipati | **Last Updated By:** | Anirudh |
| **Date Created:** | 04/01/2017 | **Last Revision Date:** |  |
| **Actors:** | Student: submit pre enrollment quiz and Academic Code of Conduct and can shop on One Stop Shop  Director: Approve and do much of key functionality as per director role.  Advisor: Approve student enrollment and shop on the website | | |
| **Description:** | The website provides user(\students, Faculty) to login | | |
| **Trigger:** | Users login to system  Student: | | |
| **Pre Conditions:** | User is registered with the system | | |
| **Post Conditions:** | **Success:** The User is authenticated and the system displays a corresponding home page upon user logins.  **Failure:** User is unable to log in for one or more reasons. | | |
| **Normal Flow:** | 1. Enter Username and Password 2. Validate Username and Password. 3. View Home page | | |
| **Alternate Flow:** | 1. Invalid Username and Password 2. Re-enter the Username and Password 3. Normal flow continues | | |
| **Exceptions:** | * Power goes off while entering username and password * User Computer Crashes * Server down * Network Issues | | |
| **Includes:** |  | | |
| **Frequency of Use:** | On demand | | |
| **Special Requirements:** | Performance and security | | |
| **Assumptions:** | User understands English. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-2 | | |
| **Use Case Name:** | Director can add Student/Advisor as a User | | |
| **Created By:** | Anirudh Ambatipati | **Last Updated By:** | Anirudh |
| **Date Created:** | 04/01/2017 | **Last Revision Date:** |  |
| **Actors:** | New User | | |
| **Description:** | The Director can create a User can be an advisor/student to access the system. | | |
| **Trigger:** | Student: Can login from the details sent through mail.  Advisor: Can login from the details sent through mail. | | |
| **Pre Conditions:** | user is not registered with the system | | |
| **Post Conditions:** | **Success:** user enters successful information and is returned to the corresponding homepage as a user.  **Failure:** User is unable to log in for one or more reasons. | | |
| **Normal Flow:** | 1. Enter Username and Password 2. Validate Username and Password. 3. View Home page | | |
| **Alternate Flow:** | 1. Invalid Username and Password 2. Re-enter the Username and Password 3. Normal flow continues | | |
| **Exceptions:** | * Power goes off while entering username and password * User Computer Crashes * Server down * Network Issues | | |
| **Includes:** |  | | |
| **Frequency of Use:** | On demand | | |
| **Special Requirements:** | Performance and security | | |
| **Assumptions:** | User understands English. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-3 | | |
| **Use Case Name:** | Student can submit the ACC and Pre-enrollment form | | |
| **Created By:** | Anirudh Ambatipati | **Last Updated By:** | Anirudh |
| **Date Created:** | 03/07/2017 | **Last Revision Date:** |  |
| **Actors:** | Director | | |
| **Description:** | The system provides director to add, delete and update the programs | | |
| **Trigger:** | If the Director wants he can update, delete, and new program to database. | | |
| **Pre Conditions:** | 1. Director should be able to login to his own webpage. 2. Webpage should allow him to update the program.(as Director role) | | |
| **Post Conditions:** | **Success:** Director can update any program he wants to. | | |
| **Normal Flow:** | 1. Director Logins 2. Director can navigate through his own webpage. 3. Looks through all the programs available already in the system 4. Chooses a program that he wants to update. 5. Chooses a program that he wants to delete. 6. Creates a program he wants to add to the system. 7. Once the delete, update, create is done database should be update 8. The updates done by the director are visible when he logins next 9. Navigate to home page 10. Logout | | |
| **Alternate Flow:** | 1. Go to Fetch student page 2. Go to create advisor page 3. Add a new student through add/edit student page 4. Delete any advisor/student. | | |
| **Exceptions:** | * Director cannot access the page due to access issues. * Director changes are not updated in the database due to Database connection issues * Power goes off creating/updating a program * Server down * Network Issues | | |
| **Includes:** |  | | |
| **Frequency of use:** | On demand | | |
| **Special Requirements:** | Performance and security | | |
| **Assumptions:** | User understands English. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-4 | | |
| **Use Case Name:** | Director can create a student and sends an Email to student | | |
| **Created By:** | Anirdh Ambatipati | **Last Updated By:** | Anirudh |
| **Date Created:** | 03/07/2017 | **Last Revision Date:** |  |
| **Actors:** | Director | | |
| **Description:** | Director can add a student to the system and send email to the new student with his login details. | | |
| **Trigger:** | User wants to sell mutual funds. | | |
| **Pre Conditions:** | 1. Director must be able to login to system. 2. Director must be able to access add/edit student page | | |
| **Post Conditions:** | **Success:** Student added to system, and has email with login details. | | |
| **Normal Flow:** | 1. Director logins 2. Go to his Homepage 3. looks for edit/delete student page 4. Enter the details of the student. 5. After he enters student details he should be able to submit. 6. Once he submits the details he should be redirected to send email page. 7. Email should be having pre texted message with login details of student. 8. Email should be sent to Student email   .   1. Submit the Student details and added to system 2. Logout | | |
| **Alternate Flow:** | 1. Go to Fetch student page 2. Go to create advisor page 3. Go to add/edit program 4. Delete any advisor/student. | | |
| **Exceptions:** | * Director cannot access the page due to access issues. * Director changes are not updated in the database due to Database connection issues * Power goes off creating/updating a program * Server down * Network Issues | | |
| **Includes:** | Email Id of the student should be available with director to send email | | |
| **Frequency of use:** | On demand | | |
| **Special Requirements:** | Performance and security | | |
| **Assumptions:** | User understands English. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-5 | | |
| **Use Case Name:** | Student Login and submits ACC | | |
| **Created By:** | Anirudh Ambatipati | **Last Updated By:** | Anirudh |
| **Date Created:** | 03/07/2017 | **Last Revision Date:** |  |
| **Actors:** | Student | | |
| **Description:** | Student able to logins to the system with the director provided credentials and submit ACC | | |
| **Trigger:** | User student logins to system and submits Academic Code of Conduct | | |
| **Pre Conditions:** | 1. Student must receive an email with credentials to login with. 2. Student must have a ACC form to submit | | |
| **Post Conditions:** | **Success:** Student can be able to submit the ACC | | |
| **Normal Flow:** | 1. Student logins 2. Student can access his portal 3. Student can view ACC and pre-exam portal. 4. Fills the ACC form 5. After filling we can able to submit ACC 6. ACC form submission details are stored in the DB 7. Logout | | |
| **Alternate Flow:** | 1. Go to Home page 2. Go to pre-exam page | | |
| **Exceptions:** | * ACC is not available to submit * Power goes off while submitting * Server down * Network Issues | | |
| **Includes:** |  | | |
| **Frequency of use:** | On demand | | |
| **Special Requirements:** | Performance and security | | |
| **Assumptions:** | User understands English. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-6 | | |
| **Use Case Name:** | Advisor can login to system | | |
| **Created By:** | Anirudh Ambatipati | **Last Updated By:** | Anirudh |
| **Date Created:** | 03/07/2017 | **Last Revision Date:** |  |
| **Actors:** | Advisor | | |
| **Description:** | Advisor can login the system from preset credentials from director | | |
| **Trigger:** | Advisor login the system and review student profile | | |
| **Pre Conditions:** | 1. Advisor can login the system with login credentials | | |
| **Post Conditions:** | **Success:** Can review the submitted student details | | |
| **Normal Flow:** | 1. Advisor can login the system with login credentials provided by the Director. 2. Advisor can review the student details. 3. Advisor can update the pre requisite status of student 4. Advisor can update the details of student 5. Advisor can change the password 6. Advisor can logout | | |
| **Alternate Flow:** | 1. Can move on to password change page and notes page | | |
| **Exceptions:** | * There are no student assigned to the advisor * No students are admitted to the concentration | | |
| **Includes:** |  | | |
| **Frequency of use:** | On demand | | |
| **Special Requirements:** | Performance and security | | |
| **Assumptions:** | Advisor credentials are provided earlier | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-7 | | |
| **Use Case Name:** | Advisor can write notes and fetch notes on student status | | |
| **Created By:** | Anirudh Ambatipati | **Last Updated By:** | Anirudh |
| **Date Created:** | 03/07/2017 | **Last Revision Date:** |  |
| **Actors:** | Advisor | | |
| **Description:** | Advisor can update the notes on student regarding student pre-requisite exam. | | |
| **Trigger:** | Advisor can be able to view student details | | |
| **Pre Conditions:** | 1. Student must have submitted the pre-requisite exam | | |
| **Post Conditions:** | **Success:** Advisor can submit a note on the student performance and edit notes also. | | |
| **Normal Flow:** | 1. Advisor can login the system with login credentials provided by the Director. 2. Advisor can review the student details. 3. Advisor can update the pre requisite status of student 4. Advisor can update the details of student 5. Advisor can change the password 6. Advisor can logout | | |
| **Alternate Flow:** | 1. Go to the folder where the downloaded data is stored and manually run the bulk insert script. | | |
| **Exceptions:** | * There are no student assigned to the advisor * No students are admitted to the concentration | | |
| **Includes:** |  | | |
| **Frequency of use:** | Daily | | |
| **Special Requirements:** | Performance and security | | |
| **Assumptions:** | Student has submitted the pre-requisite exam. | | |

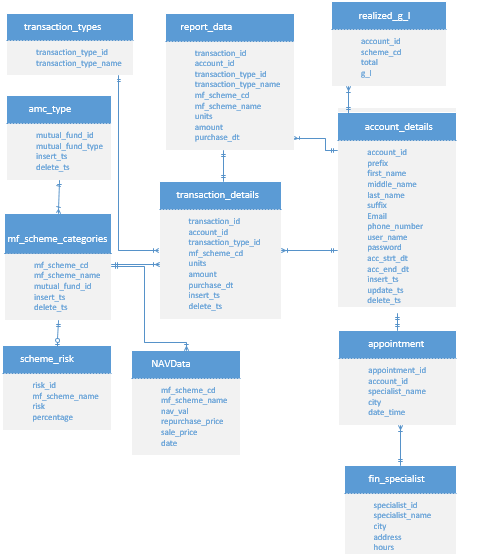
|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-8 | | |
| **Use Case Name:** | User (director, student, advisor) can login to UCM shop | | |
| **Created By:** | Anirudh Ambatipati | **Last Updated By:** | Anirudh |
| **Date Created:** | 03/07/2017 | **Last Revision Date:** |  |
| **Actors:** | User | | |
| **Description:** | User can login to the system to access the UCM shop website. | | |
| **Trigger:** | User can login to the webpage and view all the details on the page | | |
| **Pre Conditions:** | 1. User must be able to login to the system | | |
| **Post Conditions:** | **Success:** User can see the shop page | | |
| **Normal Flow:** | 1. User logins 2. Go to shop page 3. Check for items he needed to buy 4. Contact the seller 5. Select the item he wants to buy 6. Logout the system | | |
| **Alternate Flow:** |  | | |
| **Exceptions:** | * Not able to access the one shopping website * Network issues. | | |
| **Includes:** |  | | |
| **Frequency of use:** | On demand | | |
| **Special Requirements:** | Performance and security | | |
| **Assumptions:** | Some adds are already posted and login possible | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-9 | | |
| **Use Case Name:** | User can login to UCM shop and post his add | | |
| **Created By:** | Anirudh Ambatipati | **Last Updated By:** | Anirudh |
| **Date Created:** | 03/07/2017 | **Last Revision Date:** |  |
| **Actors:** | User | | |
| **Description:** | User can login to the system and post a add in the UCM shop | | |
| **Trigger:** | User wants to post an advertisement in the UCM post | | |
| **Pre Conditions:** | 1. User can login to the system and access the post page | | |
| **Post Conditions:** | **Success:** User gets posts his add and adds contact details also. | | |
| **Normal Flow:** | 1. User logins 2. Go to post page 3. Add the item details 4. Submit the advertisement on the item. 5. Post details are added to the system database 6. Submit the advertisement | | |
| **Alternate Flow:** | 1. Enter shopping page | | |
| **Exceptions:** | * There are no advisors available. * Network issues. | | |
| **Includes:** |  | | |
| **Frequency of use:** | On demand | | |
| **Special Requirements:** | Performance and security | | |
| **Assumptions:** | User can login to the page | | |

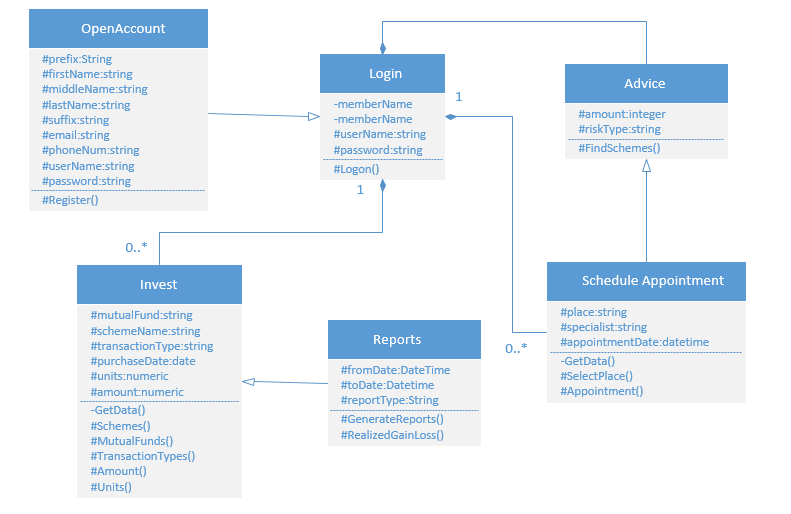
Use Case diagram

C:\Users\Anirudh raj\Downloads\Use_case.png

**Entity Relationship Diagram**

****

**Class Diagram**

****

**Critical Process**

This project has two critical processes.

1. Email to New Student: As a Director of the Department can create a student and add his/her profile to the database. Director has to send an e-mail to the student with the login credentials as student can access the website only through the provided details. Once the director has created a student and hits submit button it will automatically take him to a send email page with prefilled data with student Username and auto-generated random encrypted password in the email body. Director can send email by adding any additional comments or send as is by adding student personal id and providing director mail id password as authorization token for sending the email.

1. Shopping website: we daily see many mails floating in for international mailing list with students mailing to sell/giveaway some kind of products. So, I have added a functionality in this website to avoid that mail flow to the university mail box, instead students who are registered with the website can now post there sell/buy/giveaway request on the website itself so, any a seller has to provide valid selling/contact information along with images. And buyer can contact the seller from the information he sees on the shopping website.

**List of System Components and their Relationship**

1. Windows task Scheduler: This is the only component in this project where user cannot directly interact with while using this website. This is used to schedule jobs for downloading data and inserting data into database.

**Unit Testing**

**Add/editStudent.aspx.cs :**

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: | University of Central Missouri -CIS | Test Designed by: | Anirudh Ambatipati |
| Module Name | Add/edit student page | Test Designed date | 04-11-2017 |
| Module Description | Test to add and edit student by director role. | Test Executed by: | Anirudh Ambatipati |
| Version | 0.1 | Test Execution date: | 04-11-2017 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Steps** | **Test Data** | **Scripts called** | **Expected Result** | **Actual Result** | **P/F** |
| 1.Login | Username:70064700  Password:director700 |  | Able to login | Successfully logged in and navigated to home page | P |
| 2. Navigate to add/edit student page |  |  | Able to go to add/edit student page | Successfully navigated to add/edit student page | P |
| 3.On page load, edit/add student page should be displayed to the director | Director must enter some kind of student information like student 700# number, prereq code, scores etc. | NA | Director must be able to submit all the student details he entered and form submission must be success | Student Successfully added to the webiste database and edit/add student form submitted | P |
| 4.On page load, edit/add student page should be displayed to the director | Try to submit the form without entering any data to the edit/add student page | NA | Validations should kick in and form should not be submitted with empty data | Form submission should be unsuccessfully with validation error | P |
| 5. submit the page with phone number, 700# number as text instead of numbers | Try to enter invalid data to the page in the phone number,700# number textboxes | NA | Regular expression validations should work and error should be thrown | Form submission should be unsuccessfully with validation error | P |
|  |  |  |  |  |  |
| 6.Enter invalid Mail Id | Try to enter a in valid email id like [anirudhraj123gmail.com](mailto:anirudhraj123@gmail.com) |  | Validation should error out invalid email id | Validation error to user | P |
|  |  |  |  |  |  |
| 7.Upon submit button, record must be inserted in Student details table |  |  | Director is able to submit the data and data is saved to DB | Successfully inserted a record into Student\_detail table | P |
|  |  |  |  |  |  |

**SendEmail.aspx.cs:**

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: | University of Central Missouri -CIS | Test Designed by: | Anirudh Ambatipati |
| Module Name: | Send Email page | Test Designed date | 02-11-2017 |
| Module Description: | Send email to the student page after the student details are successfully submitted | Test Executed by: | Anirudh Ambatipati |
| Version: | 0.1 | Test Execution date: | 03-01-2017 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Steps | Test Data | Scripts called | Expected Result | Actual Result | P/F |
| 1.Login | Username: 70064700  Password:  Director700 |  | Able to login | Successfully logged in and navigated to home page | P |
| 2.Navigate to Add/edit Student page |  |  | Able to go to Add/edit page | Successfully navigate to Add/edit Student page | P |
| 3.Enter valid Student details in the form | All valid student data should be entered |  | User must be able to enter all valid data of student. | Data should be successful added to form | P |
| 4.Entered data should be submitted from Add/edit Student page | Valid Student data |  | Submit student data form Add/edit Student page to send email page | Page should successfully submit and redirect to Send Email page | P |
| 5.Data from add/edit Student page should be pre-populated to the send email page | Data entered in the form should be carried through Send email page |  | All the prepopulated data from the form should be filled in the email body and heading of the page | Successfully sees data he entered in the previous page in the email body | P |
| 6.The password populated should auto-generated and encrypted. | Password should be auto-generated. |  | The password should be added to the email body along with 700# number as a credential for student to login and password must be encrypted | Password is Successfully encrypted and added to database | p |
| 7.Enter student email and sender email id, password to the Send email form | Should be able to enter student email, sender email and password. |  | The Director should enter student email id , his credentials to send a mail to student | User is able to enter the details as required. | p |
| 8.After entering all data must be able to send an email to student. | After hitting send email button email should be send to student |  | Email with login credentials must be send to student mail id. | User is able to send the mail successfully | P |

**Add/deleteProgram.aspx.cs**

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: | University of Central Missouri -CIS | Test Designed by: | Anirudh Ambatipati |
| Module Name | Add/delete program page | Test Designed date | 03-11-2017 |
| Module Description | Test add/delete program page to edit or add a new program by director | Test Executed by: | Anirudh Ambatipati |
| Version | 0.1 | Test Execution date: | 23-11-2017 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Steps** | **Test Data** | **Scripts called** | **Expected Result** | **Actual Result** | **P/F** |
| 1.Login | Username: 70064700  Password:  Director700 |  | Able to login | Successfully logged in and navigated to Add/delete program page | P |
| 2.Navigate to Add/delete program page | User must be navigating to Add/delete program |  | Able to go Add/delete program page | Successfully navigate to advice page | P |
| 3.Edit/delete button is available to user | User must be able to see edit/delete program |  | User should be able to see edit/delete button for program | Successfully edits or deletes a program | P |
| 4.Able to edit an already existing program | User is able to edit a program |  | User should be able to edit a program | Successfully edits a program | P |
| 5.Able to delete a program form the table | User is able to delete a program |  | User is able to select the delete button and delete the program | Successfully deletes a program | P |
| 6.Edited program should be effected in the Data base | User edits a program detail |  | Edited program details are updated in the background database | Data is successfully added to database | P |
| 7.Delete a Program should effect the database record | User must be able to delete a program |  | Once the user deletes a program it must affect the database and gets deleted in database also. | Data is successfully deleted from database also. | P |
| 8.User must be able to add new program to database | User must enter new program details |  | Should be able to add a program to database. | Successfully adds a program | P |

**StudentHome.aspx.cs**

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: | University of Central Missouri -CIS | Test Designed by: | Anirudh Ambatipati |
| Module Name | Student Home page to test student functionality | Test Designed date | 03-11-2017 |
| Module Description | Test the student role functionality of the website | Test Executed by: | Anirudh Ambatipati |
| Version | 0.1 | Test Execution date: | 23-11-2017 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Steps** | **Test Data** | **Scripts called** | **Expected Result** | **Actual Result** | **P/F** |
| 1.Student Login | Username: 700647703  Password:Student@123 |  | Student should be able to login | Successfully logged in and navigated to Student Home page | P |
| 2.Navigate to Student Home page | Student must able to navigate to home page |  | Able to navigate to home page | Successfully navigate to home page | P |
| 3.Home page should have Academic code of conduct | User must be able to submit Academic code of conduct form |  | User is able to submit the Academic code of conduct | ACC is submitted successfully | P |
| 4.User is able to navigate to the submit student quiz page | User is able to navigate to student quiz page and questions are preloaded to the page |  | User able to able to student pre requisite quiz | Successfully submits the student quiz successfully | P |
| 5.Student quiz marks are added to the database | When student submits the quiz Student marks are added to the database |  | Student is able to submit the quiz page and marks are submitted to the database | Successfully submits the quiz and marks are updated in the student page | P |
| 6.Student is able to navigate to change password page | User is able to navigate to page |  | User is able to change password from the change password form | User is able to submit the change password page | P |
| 7.User changed password is updated In the database | User must be able to change the password and enter new password |  | Once the user enter a new password and submits the form password must be updated in the database | Password is successfully added to the database | P |

**AdvisorHome.aspx.cs**

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: | University of Central Missouri -CIS | Test Designed by: | Anirudh Ambatipati |
| Module Name | Advisor Home page to test Advisor functionality | Test Designed date | 03-11-2017 |
| Module Description | Test add/delete program page to edit or add a new program by director | Test Executed by: | Anirudh Ambatipati |
| Version | 0.1 | Test Execution date: | 23-11-2017 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Steps** | **Test Data** | **Scripts called** | **Expected Result** | **Actual Result** | **P/F** |
| 1.Student Login | Username: 700647001  Password:Advisor\_123 |  | Student should be able to login | Successfully logged in and navigated to Student Home page | P |
| 2.Navigate to Advisor Home page | Advisor must able to navigate to home page |  | Able to navigate to home page | Successfully navigate to home page | P |
| 3.Home page should have Fetch student information tab | User must be able to enter student 700# number in fetch student tab |  | User is able to fetch student data to the student details tab | Student details are successfully fetched by Advisor | P |
| 4.User is able to fetch the data but only able to edit the pre requisite status of the student | User must be able to enter student 700# number in fetch student tab |  | User able to fetch the student data but only able to edit student pre-req status tab and notes tab | Successfully edits pre-req status tab and notes tab | P |
| 5.User is able add notes to the student table from page | User is able to add data in the note tab of the page |  | Advisor is able to add data in the note tab based on the student pre-req status | Successfully adds notes on the student status | P |
| 6.Data entered by advisor should be added in student note table | User must enter data to the page |  | User entered data must be added to the database student note table | Data is successfully added to the website database | P |

**ShopHome.aspx.cs**

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: | University of Central Missouri -CIS | Test Designed by: | Anirudh Ambatipati |
| Module Name | User is access the UCM Shop functionality | Test Designed date | 03-11-2017 |
| Module Description | Test the UCM shop functionality of the website | Test Executed by: | Anirudh Ambatipati |
| Version | 0.1 | Test Execution date: | 23-11-2017 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Steps** | **Test Data** | **Scripts called** | **Expected Result** | **Actual Result** | **P/F** |
| 1.User Login | Username: 700647703  Password:Student@123 |  | User should be able to login | Successfully logged in and navigated to UCM Home page | P |
| 2.Navigate to UCM Shop page only after user logins | User can be able to navigate to UCM Shop page |  | User can only be navigate to UCM shop page after he/she logins | Successfully navigate to UCM Shop page | P |
| 3.User must be asked to go to Buyer page or seller page | User must be able to navigate to his choice of webpage |  | User is able to navigate either to seller page or buyer page | User is successfully navigated to page of his/her choice | P |
| 4.user is able to data of post he want to display on the university website from post page when he selects buyer page | User has to navigate to buyer page and have post details he wants to enter |  | User is able to enter the post details he wants to enter to the website | User successfully enter the data and submits the form | P |
| 5.User enter invalid data in the post page and user fails to submit the page | User has to navigate to buyer page and have post details he wants to enter |  | User enter invalid data in the form and validations kicks off and fails to submit the data | User is not able to submit the data a validations are done on the page | P |
| 6.User added post is visible on the UCM shop home page | User must enter data to the page |  | User entered data should be added to database and should be visible on the UCM shop home page | Post is visible on the home page | P |

**Scenario Level Testing**

This Scenario level testing includes user journey from login page, to different pages of the website based on the role of the user logged on.

**Role based**

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: | University of Central Missouri –CIS | Test Designed by: | Anirudh Ambatipati |
| Module Name | Overall flow of the website based on the user role | Test Designed date | 23-11-2017 |
| Version | 0.1 | Test Executed by: | Anirudh Ambatipati |
|  |  | Test Execution date: | 04-11-2017 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **(P/F)** |
| 1. | Login   |  |  | | --- | --- | | 1. | Navigate to login page | | 2. | Student page | | 3. | Advisor Page | | 4. | UCM shop page | | 2.User:700647000  3.Password: Director700  2.User:700647001  3.Password: Advisro\_700  2.User:700647703  3.Password: Student@123 | 1. User should be able to navigate to login page.  2.User should be able to login | User is navigated to login page and successfully able to login. | P |
| 2. | Director   |  |  | | --- | --- | | 1. | Navigate to Director Home page | | 2. | Navigate to Add/edit Program page | | 3. | Navigate to Add/edit Advisor page | | 4. | Fetch student status page | | 5 | Add/ edit student page | | 6 | Must be able to access UCM shop page | | 7 | Must be able to access UCM Shop post page | | |  | | --- | | 2.Must be able to fetch the data in degree table  ii. must be able to edit the data of table  iii. enter new data to table of the database | | 3.must be able to add an advisor the system  i. user must be able to add an advisor to the database | | 4.User must be able to fetch the student status  ii. student status data should be available in the DB | | 5.Must be able to navigate to student add/edit page  ii. must be able add or delete a student | | 6 User must be able to login to shop page and find the posts on the website | | 7. User must be able to able to access the post page to post an advertisement on the website | | 1. User should be able to navigate to Login page.  2. User should be able to see submit the add/delete program page with data he/she entered.  3. User should be able to submit the add student page and student must be added to the system.  4. User must be able to send an email to the student with login credentials  5. User must be able to add a Advisor to the system  6. User must be able to navigate to UCM shop page after logging.  7. User must be able to add a post to the page through UCM post page. | 1. User is navigated to Login page.  2. Sees and edits all the available degrees.  3. Adds a new advisor to the system successfully.  4 Adds a new student to the website successfully  5. sends an email to the student successfully.  6. successfully posts an add to the page of UCM shop from the Post page  7. Successfully sees the add information on the UCM shop page. | P |
| 3. | Student   |  |  |  | | --- | --- | --- | | 1 | Navigate to login page | | | 2 | Navigate to Student Quiz page | | | 3 | Navigate to ACC submit page | | | 4 | Navigate to change password page | | | 5 | | Must be able to access UCM shop page | | 6 | | Must be able to access UCM Shop post page | | |  | | --- | | 2.Must be able to submit the quiz page | | 3.must be able to submit the ACC form of the website | | 4. Must be able to change the password | | 5 User must be able to login to shop page and find the posts on the website | | 6. User must be able to able to access the post page to post an advertisement on the website | | 1. User should be able to login to reports page.  2. User should be able to submit the Student quiz page .  3.user student quiz details are added to the DB  4. User should be able change his/her password.  5. User must be able to navigate to UCM shop page after logging.  6. User must be able to add a post to the page through UCM post page. | 1. User is successfully navigated to reports page.  2. User is successfully submits to student quiz page  3. User is successfully submits to student quiz page.  4. User is successfully submits to student change password page  5. successfully posts an add to the page of UCM shop from the Post page  6. Successfully sees the add information on the UCM shop page.  . | P |
| 4. | Advisor   |  |  | | --- | --- | | 1 | Navigate to login page | | 2 | Navigate to Fetch Student page | | 3 | Navigate to edit student page | | 4 | Navigate to fetch student notes page | | |  | | --- | | 2.Must be able to submit the fetch student page | | 3. must be able to submit the edit student information.  i. must be able to add pre req status student  ii. must be able to add notes to student  iii. must be able to submit edited information to DB | | 5 User must be able fetch student notes from student notes table | | 6 User must be able to login to shop page and find the posts on the website | | 7. User must be able to able to access the post page to post an advertisement on the website | | 1. User should be able to navigate to login page.  2. User should be fetch the student information based on 700#.  3. User should be able to pre req status of the student.  4. User should be able to enter notes to the student.  5. User must be able to navigate to UCM shop page after logging.  6. User must be able to add a post to the page through UCM post page. | 1. User is successfully navigated to login page.  2. User is successfully submits to fetch student page  3. User is successfully submits to edit student page  4. user changes are successfully added to the DB  5. successfully posts an add to the page of UCM shop from the Post page  6. Successfully sees the add information on the UCM shop page. | P |