**Attendance Management System**

**Test Report**

**Version<1.1>**

**<02/04/2018>**



**Submitted By:-**

**Kavish Gambhir (B16CS010)**

**Kuldeep Singh Jangir (B16CS011)**

**Version History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version #** | **Implemented By** | **Revision Date** | **Short description** |
| **1.0** | **Kavish Gambhir & Kuldeep Singh Jangir** | **02 April 2018** | **Attendance Management System for Software Engineering Course : Initial Tests** |
| **1.1** | **Kavish Gambhir & Kuldeep Singh Jangir** | **23 April 2018** | **Integration Bugs Resolved 4.4 , 4.7 , 4.8 , 4.13 , 4.14 , 4,15 , 4,16** |

**Table of Contents**

[**1.0 INTRODUCTION...............................................................................................................5**](about:blank)

[1.1Purpose............................................................................................................................5](about:blank)

[**2.0 TEST PLAN........................................................................................................................5**](about:blank)

[**3.0 DETAILS OF UNIT TESTING..............................................................................................5**](about:blank)

[3.1 User :: getName()......................................................................5](about:blank)

1. Feature To Be Tested
2. Pass/Fail Criteria

[3.2 User :: viewDefaulters()......................................................................5](about:blank)

1. Feature To Be Tested
2. Pass/Fail Criteria

[3.3 User :: viewAttendance()......................................................................](about:blank)6

1. Feature To Be Tested
2. Pass/Fail Criteria

[3.4 User :: loadPresentCountList()......................................................................](about:blank)6

1. Feature To Be Tested
2. Pass/Fail Criteria

[3.5 User :: storePresentCountList()......................................................................](about:blank)7

1. Feature To Be Tested
2. Pass/Fail Criteria

[3.6 Interface\_handler :: \_register()......................................................................](about:blank)7

1. Feature To Be Tested
2. Pass/Fail Criteria
3. Password Strength Criteria

[3.7 Faculty :: addStudent()......................................................................](about:blank)9

1. Feature To Be Tested
2. Pass/Fail Criteria

[3.8 Faculty :: deleteStudent()......................................................................](about:blank)9

1. Feature To Be Tested
2. Pass/Fail Criteria

[3.9 Faculty :: addAttendance()......................................................................](about:blank)10

1. Feature To Be Tested
2. Pass/Fail Criteria

[3.10 Faculty :: modifyAttendance()......................................................................](about:blank)11

1. Feature To Be Tested
2. Pass/Fail Criteria

[3.11 Admin :: addFaculty()......................................................................](about:blank)11

1. Feature To Be Tested
2. Pass/Fail Criteria

[3.12 Interface\_handler :: loginFaculty()...........................................................](about:blank)12

1. Feature To Be Tested
2. Pass/Fail Criteria

[3.13 Admin :: deleteFaculty()......................................................................](about:blank)12

1. Feature To Be Tested
2. Pass/Fail Criteria

[3.14 Faculty :: isValidDate( year, month, day)............................................](about:blank)13

1. Feature To Be Tested
2. Pass/Fail Criteria

[**4.0 DETAILS OF SYSTEM TESTING.......................................................................................**](about:blank)**13**

[4.1 Details of Sysytem Testing.....................................................................................](about:blank)13

**5.0 UNRESOLVED TEST INSTANCES.....................................................................................14**

**1.0 INTRODUCTION**

**1.1 PURPOSE**

This *Attendance Management System Unit and System* Test Report provides a summary of the results of test performed as outlined within this document.The purpose of making this test report is to verify each function and its integration with the system and the surroundings. Each functionality required is met with the standards and is working fine or not. Also it finds and discover the defects and errors that may have occurred during development or due to any other reason. It helps to find the faults as well. This Test report also states the delivered product is as per specifications of SRS document and if not then point of deviation and reason as well so that corrective measures can be taken.

**2.0 TEST PLAN**

There are various types of test plans but we are doing testing level specific test plans and under this also we are following **unit test plan** and  **system test plan**. The idea is doing the unit testing of each function and then test the integration of system as whole. We are using Bottom-Up strategy for testing.

**3.0 DETAILS OF UNIT TESTING**

3.1 User :: getName()

1. Feature To Be Tested : Return name of user with return type as string
2. Pass/Fail Criteria : Returned string is name of user

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number | Input/Local Variables | Expected Output | Actual Output | Pass/Fail | Verified |
| 3.1 | An object of user with name “kuldeep” | Kuldeep | Kuldeep | Pass | Pass |

3.2 User :: viewDefaulters()

1. Feature To Be Tested : Prints list of students (Roll Number) who has attendance less than 85% with their respective attendance
2. Pass/Fail Criteria : List of students printed are actually having attendance less than 85%. The printed attendance match with actual attendance

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number | Input/Local Variables | Expected Output | Actual Output | Pass/Fail |  |
| 3.2 | Required Files storing attendance of two days for two students.  B16CS010 | 0 | 1  B16CS011 | 1 | 1 | B16CS010 50 % | B16CS010 50 % | Pass | Pass |

3.3 User :: viewAttendance()

1. Feature To Be Tested : Prints All the attendance with Date and respective Absentees of that date.
2. Pass/Fail Criteria : Printed students were mentioned absent on that date in the files.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number | Input/Local Variables | Expected Output | Actual Output | Pass/Fail |  |
| 3.3 | Required Files storing attendance of two dates for two students.  21-2-2018  B16CS010 0  B16CS011 1  23-2-2018  B16CS010 1  B16CS011 0 | 21-2-2018 -- B16CS010  23-2-2018 -- B16CS011 | 21-2-2018 -- B16CS010  23-2-2018 -- B16CS011 | Pass | Pass |

3.4 User :: loadPresentCountList()

1. Feature To Be Tested : Loads the file presentCount.txt to map prescentCountList. Checks for existence of file also
2. Pass/Fail Criteria : If file is present then map is loaded with data from file prescentCount.txt. If file is not present then empty map.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number | Input/Local Variables | Expected Output | Actual Output | Pass/Fail |  |
| 3.4.1 | Required File presentCount.txt storing Roll Number and presents for two students.  B16CS001 1  B16CS002 2 | Map presentCountList is loaded | Map presentCountList is loaded | Pass | Pass |
| 3.4.2 | File presentCount.txt is not present | Map is empty | Map is Empty | Pass | Pass |

3.5 User :: storePresentCountList()

1. Feature to be tested : Store the map presentCountList to file presentCount.txt(Database)
2. Item Pass/Fail Criteria : File contains the data is same as map.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number | Input/Local Variables | Expected Output | Actual Output | Pass/Fail | Verified |
| 3.5.1 | Map has Roll Number and presents for two students.  B16CS001 1  B16CS002 2 | File has  B16CS001 1  B16CS002 2 | File has  B16CS001 1  B16CS002 2 | Pass | Pass |

3.6 Interface\_handler :: \_register()

1. Feature to be tested : Registers new faculty and stores it into facultyMap and database. Requires admin Password. Takes input Faculty Name , faculty Password. Only Strong Passwords are accepted. User re-enters the entered password to confirm registration (max 3 times allowed)
2. Item Pass/Fail Criteria : Faculty is registered if conditions are satisfied else goes to homeView().
3. Password Strength Criteria : 6 <= length of password <= 13, at least One Capital Letter and at least one Numeric letter

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number | Input/Local Variables | Expected Output | Actual Output | Pass/Fail |  |
| 3.6.1 | Admin password entered is “qwerty” | Calls homeView() function | Calls homeView() function | Pass | Pass |
| 3.6.2 | Admin Password is “cs223” | Asks for name and Password of faculty | Asks for name and Password of faculty | Pass | Pass |
|  | Entered name is “Kuldeep Singh” and Password is “Kuldeep” | Weak password. Asks for Password again | Weak password. Asks for Password again | Pass | Pass |
|  | Entered Password is “Kuldeep123” | Accepts password and asks to re-enter the password | Accepts password and asks to re-enter the password | Pass | Pass |
|  | Re-entered password is “Kuldeep1234” | Password does not match. Enter again. | Password does not match. Enter again. | Pass | Pass |
|  | Re-entered password is “Kuldeep123” | Faculty object created and addFaculty() function called in database\_handler.  Faculty added to map facultyMap and to file faculty\_details.txt | Faculty object created and addFaculty() function called in database\_handler.  Faculty added to map facultyMap and to file faculty\_details.txt | Pass | Pass |

3.7 Faculty :: addStudent()

1. Feature to be tested : takes Roll Number from user, verify its existence. Adds only if not present, takes name from user, add student to database by calling function from databse\_handler
2. Item Pass/Fail Criteria : Student Added to studentsMap , presentCountList , respective files

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number | Input/Local Variables | Expected Output | Actual Output | Pass/Fail | Verified |
| 3.7.1 | Input: B16CS020 | Nikhil Srivastava | Student updated in studentsMap, presentCountList, presentCount.txt  Nikhil Srivastava succesfully registered | Student updated in studentsMap, presentCountList, presentCount.txt  Nikhil Srivastava succesfully registered | Pass | Pass |
| 3.7.2 | Input: B16CS020 | Student already registered as Nikhil Srivastava | Student already registered as Nikhil Srivastava | Pass | Pass |

3.8 Faculty :: deleteStudent()

1. Feature to be tested : Takes Roll Number from faculty, to be deleted. Verify existence of student, deletes only if student present. Deletes student by calling delete function from database\_handler.
2. Item Pass/Fail Criteria : student removed from studentsMap, studentsList.txt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number | Input/Local Variables | Expected Output | Actual Output | Pass/Fail | Verified |
| 3.8.1 | Input: B16CS020 | y | Student deleted from studentsMap, studentsList.txt.  Nikhil Srivastava is de-registered from course | Student deleted from studentsMap, studentsList.txt.  B16CS020 is de-registered from course | Pass | Pass |
| 3.8.2 | Input: B16CS020 | B16cs020 is not registered in the course | B16cs020 is not registered in the course | Pass | Pass |

3.9 Faculty :: addAttendance()

1. Feature to be tested : Attendance for a Date is added by giving input the roll number of students absent on that date. Validity of date is also checked. A file with name date.txt is created in folder “date”. Presents of students is updated in presentCountList as well as in file presentCount.txt. the Date is added to lectureDates annd lectureDates.txt
2. Item Pass/Fail Criteria : attendance is added for date in file and number of presents of students is also update. The result is reflected in the list of defaulters.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number | Input/Local Variables | Expected Output | Actual Output | Pass/Fail | Verified |
| 3.9.1 | Input: 29 | 2 | 2018 | Invalid Date | Invalid Date | Pass | Pass |
| 3.9.2 | Input: 15 | 2 | 2018 | 2 | B16CS010 | B16CS011 | File 15-2-2018.txt is created in folder date. Presents Count is updated for both students. | File 15-2-2018.txt is created in folder date. Presents Count is updated for both students. | Pass | Pass |
| 3.9.3 | Input: 15 | 2 | 2018 | 2 | B16CS010 | B16CS020 | File 15-2-2018(1).txt is created in folder date. Presents Count is updated for B16CS010 but not for B16CS020 as B16CS020 is not registered | File 15-2-2018(1).txt is created in folder date. Presents Count is updated for B16CS010 but not for B16CS020 as B16CS020 is not registered | Pass | Pass |

3.10 Faculty :: modifyAttendance()

1. Feature to be tested : A valid date , roll number of student , new attendance status ( 0 for absent and 1 for present ) is entered. if there was a lecture on that date and student is registered then, attendance of student is updated in both database as well as files, else error message is shown.
2. Item Pass/Fail Criteria : Updated attendance is reflected in defaulters list and in file containing attendance of that date.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number | Input/Local Variables | Expected Output | Actual Output | Pass/Fail | Verified |
| 3.10.1 | Input: 29 | 2 | 2018 | Lecture on this date does not exist | Lecture on this date does not exist | Pass | Pass |
| 3.10.2 | Input: 15 | 2 | 2018 | 2 | B16CS020 | 0 | Student roll number is wrong | Student roll number is wrong | Pass | Pass |
| 3.10.3 | Input: 15 | 2 | 2018 | 2 | B16CS010 | 0 | Same attendance is already in database | Same attendance is already in database | Pass | Pass |
| 3.10.4 | Input: 15 | 2 | 2018 | 2 | B16CS010 | 1 | Modification of B16CS010 is done to 1 on date 15-2-2018. | Modification of B16CS010 is done to 1 on date 15-2-2018. | Pass | Pass |

3.11 void Admin :: addFaculty()

1. Feature to be tested : \_register function is called to add faculty
2. Item Pass/Fail Criteria : \_register function is called to add faculty

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number | Input/Local Variables | Expected Output | Actual Output | Pass/Fail | Verified |
| 3.11.1 | Input: NA | \_register function is called | \_register function is called | Pass | Pass |
|  | Input: cs223 | Shivam Nirne | Shivam123 | Shivam123 | Shivam Nirne succesfully registered as faculty | Shivam Nirne succesfully registered as faculty | Pass | Pass |

3.12 Interface\_handler :: loginFaculty()

1. Feature to be tested : Faculty registered is logged in with correct credentials input
2. Item Pass/Fail Criteria : If faculty is registered and input credentials matched then logged in else return to homeView

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number | Input/Local Variables | Expected Output | Actual Output | Pass/Fail | Verified |
| 3.12.1 | Input: Shivam nirne | Shivam123 | Invalid Name | Invalid Name | Pass | Pass |
| 3.12.2 | Input: Shivam Nirne | Shivam1234 | Invalid Password | Invalid Password | Pass | Pass |
| 3.12.3 | Input: Shivam Nirne | Shivam123 | Welcome Dr. Shivam nirne | Welcome Dr. Shivam nirne | Pass | Pass |

3.13 Admin :: deleteFaculty()

1. Feature to be tested : Deletes a Faculty ( name and password) from faculty\_details.txt and from facultyMap. This faculty is no longer able to login as faculty.
2. Item Pass/Fail Criteria : Faculty is deleted from database and files. Faculty deleted is unable to login after deletion.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number | Input/Local Variables | Expected Output | Actual Output | Pass/Fail | Verified |
| 3.13.1 | Input: Shivam nirne | No such faculty present | No such faculty present | Pass | Pass |
| 3.13.2 | Input: Shivam Nirne | Shivam Nirne is removed from Course | Shivam Nirne is removed from Course | Pass | Pass |
| 3.13.3 | Trying to login as faculty.  Interface\_handler :: loginFaculty()  Input : Shivam Nirne | Shivam123 | Invalid Name | Invalid Name | Pass | Pass |

3.14 Faculty :: isValidDate( year, month, day)

1. Feature to be tested : returns true if a date is valid else returns false if date is invalid. Invalid date means such a date can never exist.
2. Item Pass/Fail Criteria : True is returned for a valid date and if true is returned the the date is valid, same as for false and invalid date.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number | Input/Local Variables | Expected Output | Actual Output | Pass/Fail | Verified |
| 3.13.1 | Input: 2018 | 2 | 29 | False | False | Pass | Pass |
| 3.13.2 | Input: 2016 | 2 | 29 | True | True | Pass | Pass |
| 3.13.3 | Input: 2018 | 2 | 15 | True | True | Pass | Pass |

**4.0 DETAILS OF SYSTEM TESTING**

4.0 System Testing

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step #** | **Test Step** | **Intended Result** | **Actual Result** | **Pass/Fail** | **Comments** | **Resolution** |
| 1 | Login as Admin -> Add Faculty | New Faculty added | New Faculty is added | Pass | Verified |  |
| 2 | Login As Admin -> Log out from Admin | Logged Out -> Home View | Logged Out -> Home View | Pass | Verified |  |
| 3 | Login As Faculty -> Add Student | New Student is Added | New Student is Added | Pass | Verified |  |
| 4 | Login As Faculty -> Modify Student | Student Name is modified | Student Name is modified | Pass | Failed | Bug Fixed |
| 5 | Login As Student -> Delete Student | Student is deleted from course | Student is deleted from course | Pass | Verified |  |
| 6 | Login As Faculty -> Add Attendance | Attendance is Added | Attendance is Added | Pass | Verified |  |
| 7 | Login As Faculty -> Modify Attendance | Attendance is Modified | Attendance is Modified | Pass | Failed if modified for the same status | Bug Fixed |
| 8 | Login As Faculty -> Change Password | Password of Faculty is Changed | Password of Faculty is Changed | Pass | Failed password is changed but unable to re-login | Bug Fixed |
| 9 | View Defaulters | Defaulters List with respective % Attendance is displayed | Defaulters List with respective % Attendance is displayed | Pass | Pass |  |
| 10 | View Attendance | Attendance of all the lectures is displayed | Attendance of all the lectures is displayed | Pass | Pass |  |
| 11 | Login As Admin -> modify faculty | Faculty name is changed | Faculty name is changed | Pass | Pass |  |
| 12 | Login As Admin -> remove faculty | Faculty is removed from course and no longer able to login as faculty | .Faculty is removed from course and no longer able to login as faculty | Pass | Pass |  |
| 13 | Login As faculty -> Log out | Faculty is logged Out -> Home View | Faculty is logged Out -> Home View | Pass | Pass but logged out printed thrice | Bug Fixed |
| 14 | Login As faculty->Register a student ->modify attendance | Status changed to absent, present | Student roll number is wrong | Failed | User is already registered still not finding student roll number | Bug Fixed |
| 15 | Login as faculty->modify attendance of a particular student | Date, Roll number and 0 is displayed | Only date is printed | Failed | Re-updating the same status again for student | Bug Fixed |
| 16 | Login as faculty-> change password->Re-login | Password changed successfully  Logged in | Password changed successfully  Invalid password | Failed | Space separated password is updated | Bug Fixed |

**5.0 Unresolved Test Instances**

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Summary Of Defect** | **Why Not Resolved** |
|  |  |  |