**ATTENDENCE MANAGEMENT SYSTEM**

**TEST REPORT**

**DATE: 06/04/2017**

VERSION HISTORY

|  |  |  |
| --- | --- | --- |
| **Version #** | **Implemented**  **By** | **Revision**  **Date** |
| 1.0 | *Group 5* | *06/04/2017* |

**Table of Contents**

[1.0 Introduction 3](#_Toc479062434)

[1.1 Purpose 3](#_Toc479062435)

[2.0 Test PLAN 3](#_Toc479062436)

[3.0 Test CASE Selection 3](#_Toc479062437)

[4.0 Test Results 4](#_Toc479062438)

[4.1 Functionality 1 4-5](#_Toc479062439)

[4.2 Functionality 2 5](#_Toc479062440)

[4.3 Functionality 3 7](#_Toc479062441)

[5.0 HEURISTIC UX tESTING 8](#_Toc479062442)

[6.0 Test Instances 7](#_Toc479062443)

[6.1 Resolved Test Incidents 7](#_Toc479062444)

[6.2 Unresolved Test Incidents 7](#_Toc479062445)

# 1.0 **Introduction**

## 1.1 Purpose

This **ATTENDENCE MANAGEMENT SYSTEM** Test Report provides a summary of the results of test performed as outlined within this document.

# **2.0** **TEST PLAN**

# The test plan for this project can be summed up into the following major goals:

# 1. To check whether the actual software follows the SRS completely or not

# and if not then accommodate the changes in the report.

# 2. To check the software for any errors, bugs and unexpected

# behavior and if there, then changes in the report.

# 3. To see if all functionality works in the project.

# 4. To see if UI/UX of the software is consistent and follows the standard so everyone can understand it easily.

# 5. To give the precise suggestions to overcome the error part.

# **3.0** **Test Case Selection**

# Test cases were created to test the following functionalities of the software:

# 1**. Admin’s functionalities**: Admin is the first one who can add Students, Instructors and Courses for the entire project and if there is any error in this, than the whole system will not work.

# 2. **Student’s functionalities**: Here the student can select the courses and view his/her attendance, so this functionality should be checked with different test cases ,if there is any problem in selection of courses then the Instructor will not able to take his attendance in that particular course.

# 3**. Instructor’s functionalities**: Here the Instructor can add the attendance of the students and updated the attendance in case of any discrepancy, this is the major functionality of the system and should be checked against various test cases and if any error occurred then report in the test cases specified in the test report.

# 4.0 **Test Results**

# Number of tests conducted: 11

Number of tests failed: 8

Number of tests passed: 3

## Admin’s functionalities

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Intended Result | Actual Result | Pass/Fail | Severity of Defect [Low/ Medium/ High] | Summary of Defect | Comments |
| T1.1 | When course name is given it should have been understood. | It takes negative and some special characters also. | Fail | Medium | Course names should not need to have negative values |  |
| T1.2 | There will  be only  one  instructor  for one  course. | Accepting more than one instructor for one course | Fail | High | SRS has assumption that one course can have only one instructor but in actual system, a course can have multiple instructor |  |
| T1.3 | Roll number have some meaning it should be like b30. | Roll number should not be like \*\*\*\*. | Fail | Medium | If roll number is like this we cannot make distinguish between students by roll number them. |  |
| T1.4 | Email & Mobile number should be of valid format | It shows an error message when we type some ambiguous inputs. | Pass | N/A | When user tries some ambiguous inputs it shows some error message on display. |  |
| T1.5 | When we write some ambiguous input than numeric in semester then it always shows zero. | Successfully added. | Fail | Medium | Semester can be represented like in roman letters but it takes always zero when we take anything other than numeric value. |  |

## Student’s functionalities

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Intended Result | Actual Result | Pass/Fail | Severity of Defect [Low/ Medium/ High] | Summary of Defect | Comments |
| T2.1 | Courses can be  registered  only once  by student. | Student  Can register  Course any time | Fail | High | Student can change their  Registered course  at any time  because they can register  Course more than once. |  |
| T2.2 | Students should not have same roll no. | Shows error message. | Pass | N/A | It shows an error message when we type the existing students roll no. |  |
| T2.3 | Successfully deleted during deletion of students if something ambiguous input is deleted. | Successfully deleted | Fail | Low | Instead of showing an error message when user id is incorrect ,it shows successfully deleted but nothing actually is being deleted |  |

## . Instructor’s functionalities

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Intended Result | Actual Result | Pass/Fail | Severity of Defect [Low/ Medium/ High] | Summary of Defect | Comments |
| T3.1 | During login, password should be what is during sign up. | Successful in login. | Fail | High | As password is a security issue which we are protected. It Should be actual whatever be given during sign up. It also accept the uppercase letters also. | When we are taking UPPERCASE instead of lowercase then the password is also accepted. |
| T3.2 | Any Date of birth is accepted. | Date of Birth is shown of the future dates also and accepted. | Fail | Medium | Date of Birth for any user cannot be in future values. It should be before present date |  |
| T3.3 | Instructor should be able to view attendance. | Instructor can view attendance | pass | N/A | It shows the desired attendance sheet of the students. |  |

# 5.0 HEURISTIC UX TESTING

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case ID | Issue | Heuristics | Severity |
| UT1 | User is unable to distinguish between login page of instructor and student | Visibility and System status | Low |
| UT2 | Wrong serial number in view instructor UI | Consistency and standards | Low |
| UT3 | Undefined function(Add attendance) in student view attendance UI | Aesthetic and minimalist design | Low |
| UT4 | At signup if user enter a wrong information after error message whole information remove rather than wrong information | Recognition rather than recall | Low |
| UT5 | Showing code implementation error at UI | Aesthetic and minimalist design,  Visibility of System | High |

# 6.0 Test Instances

[Provide a brief description of the unexpected results, problems, or defects that occurred during the testing.]

## Resolved Test Incidents

*[Identify all resolved test incidents and summarize their resolutions. Reference may be made to Test Incident Reports that describe in detail the unexpected results, problems, or defects reported during testing, along with their documented resolutions, which may be included as an appendix to this document.]*

*[This will be filled by the dev team.]*

|  |  |  |
| --- | --- | --- |
| Test Case ID | Summary of Defect | Resolution Taken |
| *T2.3* |  |  |
| *T1.5* |  |  |
| *T1.1* |  |  |
| *T3.1* |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Unresolved Test Incidents

[Identify all unresolved test incidents and provide a plan of action for their resolution. Reference may be made to Test Incident Reports that describe in detail the unexpected results, problems, or defects reported during testing, which may be included as an appendix to this document.]

*[This will be filled by the dev team.]*

|  |  |  |
| --- | --- | --- |
| Test Case ID | Summary of Defect | Why not resolved |
|  |  |  |
|  |  |  |
|  |  |  |