|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Software Test Plan**  [Student Portal for Azaan]  Version: [2.0]   |  |  | | --- | --- | | Supervisor | M. Asim Ali | | Co Supervisor (if any) |  | |  |  | | Project Team Members | * Abubakar Siddiq(1312166) * Amirul Sunesara(1312169) * Aamir shehzad(1312165) | | Submission Date | 5/07/2016 | |

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Name of Person** | **Date** | **Description of change** |
| 1.0 | Abubakar Siddiq,  Amirul Sunesara,  Aamir Shahzad | 01/09/2015 | Documentation created according to Software engineering I assigned format. |
| 2.0 | Abubakar Siddiq,  Amirul Sunesara,  Aamir Shahzad | 20/03/2016 | Documentation re-created according to Software engineering II assigned format. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Distribution List

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | |
| M. Asim Ali | | Supervisor |
| N.A | | Co- Supervisor |
|  | |  |

# Document Sign-Off

|  |  |  |
| --- | --- | --- |
| **Version** | **Sign-off Authority** | **Sign-off Date** |
| 1.0 | Faria Jameel | 15/12/2015 |
| 2.0 | M. Asim Ali | 5/07/2016 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Document Information

| **Category** | **Information** |
| --- | --- |
| Customer | SZABIST |
| Project | <Project Title> |
| Document | Software Test Plan |
| Document Version | 1.0 |
| Status | Draft |
| Author(s) | <Names of all the authors of this document> |
| Approver(s) |  |
| Issue Date |  |
| Document Location |  |
| Distribution | Advisor  Project Coordinator’s Office |

Definition of Terms, Acronyms and Abbreviations

| **Term** | **Description** |
| --- | --- |
| ASP | Active Server Pages |
| DD | Design Specification |
| SDLC | Software development life cycle |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Table of Contents**

Contents

[1. Introduction 7](#_Toc450374766)

[1.1. Purpose of Document 7](#_Toc450374767)

[1.2. Intended Audience 7](#_Toc450374768)

[1.3. Document Convention 7](#_Toc450374769)

[1.4. Project Overview 7](#_Toc450374770)

[2. Scope of Testing 8](#_Toc450374771)

[3. Test Plan Strategy 9](#_Toc450374772)

[3.1. Unit Testing 10](#_Toc450374773)

[3.2. Integration Testing 10](#_Toc450374774)

[3.3. System Testing 10](#_Toc450374775)

[4. Test Environment 11](#_Toc450374776)

[5. Schedule 12](#_Toc450374777)

[6. Test Case Design and Description 13](#_Toc450374778)

[7. Major Deliverables 33](#_Toc450374779)

[8. Exit Criteria 33](#_Toc450374780)

# Introduction

## Purpose of Document

This document deals with testing phase of software developing life cycle (SDLC). This document covers all the activity conducted during testing phase. Reason behind preparing this document is to ensure that the products is delivered with quality. This document will be used by customer, design and test team.

## Intended Audience

This document is intended to be used by following group of people:

1. Customer

2. Design team

3. Testing team

## Document Convention

All the text will be in Arial font but sizes will differ in following manner:

1. Main headings will be in font size 16 with bold text.
2. Subheadings will be in font size 14 with bold text.
3. Normal text will be in font size 10.

## Project Overview

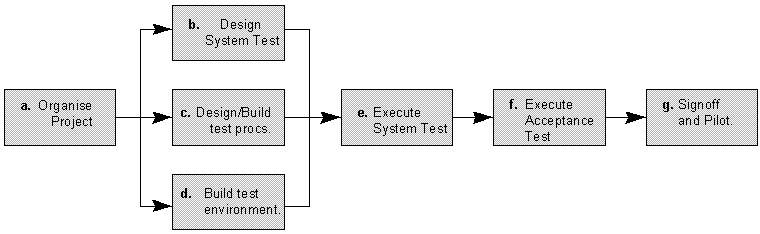
This software/application will be providing our client organization with an ease to control its resources under one umbrella. It will allow user to keep them updated with new upcoming courses, register new courses and to keep in touch with resources of their previous courses.

# Scope of Testing

Testing is done to ensure quality in product. Properly tested software results in a quality software. Following items will be tested to ensure software quality:

1. All functional requirements will be tested to ensure that all customer demands are met.
2. Non-functional requirements such as security and usability will also be tested ensure that a quality product is delivered.

# Test Plan Strategy



## Unit Testing

Definition:

Unit Test consists of testing individual programs or subroutines as they are written instead of testing the entire system after it has been written. The testing of the smaller building blocks is done first and then these blocks are combined and tested. Unit testing means testing each function independently to verify correct processing in a stand-alone environment.

Participants*:*

Abubakar Siddiq

Methodology:

1. Initial Unit Test will be conducted by the developers
2. As the completed modules are received from the development department, the test team will start the Unit Test on these modules

## Integration Testing

Definition:

Integration testing as its name implies is done with the purpose to test how individual componenets work when they are integareted.

Participants

Amirul Sunesra

Methodology:

1. Integration test will start using bottom up approach, low level components will be combined and tested first.
2. After success of low level integration higher components will be integrated and tested.

## System Testing

Definition:

System test is testing phase where complete system as a whole is tested.

Participants

Aamir Shahzad

Methodology:

1. system as a whole will be tested,
2. any issue will be dealt by highlighting and resolving the component or integration creating problem.

# Test Environment

System can be checked with basic working environment. Some of the pretest conditions are listed below:

1. A computer with a workable set of configuration.
2. Development tools such as using sql server and Microsoft visual studio must be there using which system is built.

*[This section lists the hardware and the software configuration required to perform tests. It determines the type of help required from other departments to support testing]*

# Schedule

­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 **Testing Activities Begin End Person Responsible**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Designing Test Cases 01-03-2016 20-03-2016 Amirul Sunesra

Executing Test Cases 21-03-2016 10-04-2016 Aamir Shehzad

Unit Testing 11-04-2016 20-04-2016 Abubakar Siddiq

Integration Testing 21-04-2016 30-04-2016 Amirul Sunesra

System Testing 01-05-2016 06-05-2016 Aamir Shehzad

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Test Case Design and Description

***12.1. Test Case 1***

|  |  |  |
| --- | --- | --- |
| **Test Case #:** 1.0  **System:** StudentManagementSystem  **Designed By:** Amirul  **Executed By:** Amirul  **Short Description:** SignUp Procedure allows students registered for workshops by providing their bio data and login information | **Test Case Name:** SignUp  **Subsystem:** Main Page  **Design Date: 13/4/2016**  **Execution Date: 20/4/2016** | |
| **Pre-Conditions:**  A database must hold the input values.  SignUp procedure allows server side validation and successful entry of inputs after clicking submit button. | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comments** | | 1  2  3  4  5 | Insert Empty Details | Validation Occurs | Fail |  | | Insert valid details | Input text for Password must be \* steric | Pass | Input Text type for Password is Corrected | | Insert String values for numeric values | Validation Occurs | Fail |  | | Insert Password with minimum length | Validation Occurs | Fail |  | | Insert Valid Values | Successfully submitted details | Fail |  | | | |
| **Post Conditions :**  Input elements successfully stored in database. | | |

***12.2. Test Case 1.1***

|  |  |  |
| --- | --- | --- |
| **Test Case #:** 1.1  **System:** StudentManagementSystem  **Designed By:** Amirul  **Executed By:** Amirul  **Short Description:** Login Allows Instructor, Admin,students to login from main page | **Test Case Name:** Login  **Subsystem:** Main Page  **Design Date: 13/4/2016**  **Execution Date: 20/4/2016** | |
| **Pre-Conditions:**  A database must verify the input values.  On successful verification user navigated to dashboard Panel | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comments** | | 1  2  3 | Insert Empty Details | Authentication Failed | Fail |  | | Insert valid details | Redirected to dashboard | fail |  | | Insert Invalid Details | Authentication Failed | Fail |  | | | |
| **Post Conditions :**  User hold in their respective state until session timeout. | | |

***12.2. Test Case 2***

|  |  |  |
| --- | --- | --- |
| **Test Case #:** 2.0  **System:** StudentManagementSystem  **Designed By:** Amirul  **Executed By:** Amirul  **Short Description:** SignUp Procedure allows admin to registered for instructors. | **Test Case Name:** Instructor SignUp  **Subsystem:** Administrator  **Design Date: 13/4/2016**  **Execution Date: 20/4/2016** | |
| **Pre-Conditions:**  A database must hold the input values.  SignUp procedure allows server side validation and successful entry of inputs after clicking submit button. | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comments** | | 1  2  3  4  5  6 | Insert Empty Details | Validation Occurs | Fail |  | | Insert valid details | Input text for Password must be \* steric | Pass | Input Text type for Password is Corrected | | Insert String values for numeric values | Validation Occurs | Fail |  | | Insert Password with minimum length | Validation Occurs | Fail |  | | Insert Valid Values | Successfully submitted details | Fail |  | | Insert Invalid Cell phone and telephone number length and values | Validation Occurs | Pass | Exception occurs, corrected by assigning valid length and data type | | | |
| **Post Conditions :**  Input elements successfully stored in database. | | |

***12.2. Test Case 2.1***

|  |  |  |
| --- | --- | --- |
| **Test Case #:** 2.1  **System:** StudentManagementSystem  **Designed By:** Amirul  **Executed By:** Amirul  **Short Description:** Admin can view accounts of instructors and able to delete selected accounts | **Test Case Name:** View Instructors Details  **Subsystem:** Administrator  **Design Date: 13/4/2016**  **Execution Date: 20/4/2016** | |
| **Pre-Conditions:**  Database must have details of instructors. | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comments** | | 1  2 | Open view instructor tab | Display Instructors if they are registered | Pass | They is no option indicating, no instructors are registered | | Delete Instructors by enable checkbox beside names and details | Instructors Deleted | fail |  | | | |
| **Post Conditions :**  List of instructors must be shown that remains after deleting specified instructors. | | |

***12.2. Test Case 2.2***

|  |  |  |
| --- | --- | --- |
| **Test Case #:** 2.2  **System:** StudentManagementSystem  **Designed By:** Amirul  **Executed By:** Amirul  **Short Description:** Course is Added by specifying its name and assigning an instructor | **Test Case Name:** Add Course  **Subsystem:** Administrator  **Design Date: 13/4/2016**  **Execution Date: 20/4/2016** | |
| **Pre-Conditions:**  Instructor must be already register in order to assign them respective courses | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comments** | | 1 | Write course name and assign course from selected instructor from dropdown menu | Successfully assigned with course | fail |  | | | |
| **Post Conditions :**  Instructors must be shown with assigned course when navigating to view instructors details | |

***12.2. Test Case 2.3***

|  |  |  |
| --- | --- | --- |
| **Test Case #:** 2.3  **System:** StudentManagementSystem  **Designed By:** Amirul  **Executed By:** Amirul  **Short Description:** Table represent total count of students registered for respective course | **Test Case Name:** View Student Enrollment  **Subsystem:** Administrator  **Design Date: 13/4/2016**  **Execution Date: 20/4/2016** | |
| **Pre-Conditions:**  Instructors must be assigned with courses | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comments** | | 1 | View Enrolled Students | Number of Enrolled Student student with specific courses | fail |  | | | |
| **Post Conditions :**  No Condtions | |

***12.2. Test Case 2.4***

|  |  |  |
| --- | --- | --- |
| **Test Case #:** 2.4  **System:** StudentManagementSystem  **Designed By:** Amirul  **Executed By:** Amirul  **Short Description:** Table come up with list of all students | **Test Case Name:** View Students  **Subsystem:** Administrator  **Design Date: 13/4/2016**  **Execution Date: 20/4/2016** | |
| **Pre-Conditions:**  Students must be sign up | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comments** | | 1  2 | Click on View Students | List of Students | fail |  | | Search student | Student Record related to search query | fail |  | | | |
| **Post Conditions :**  No Condtions | |

***12.2. Test Case 2.4***

|  |  |  |
| --- | --- | --- |
| **Test Case #:** 2.4  **System:** StudentManagementSystem  **Designed By:** Amirul  **Executed By:** Amirul  **Short Description:** table Shows average feedback in all aspects | **Test Case Name:** Student’s Feedback  **Subsystem:** Administrator  **Design Date: 13/4/2016**  **Execution Date: 20/4/2016** | |
| **Pre-Conditions:**  Students need to submit feedback | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comments** | | 1 | Click on Student’s Feedback | Tables shows average feedback graded by remarks | fail |  | | | |
| **Post Conditions :**  Reveals feedback anonymously | | |

***12.2. Test Case 2.5***

|  |  |  |
| --- | --- | --- |
| **Test Case #:** 2.5  **System:** StudentManagementSystem  **Designed By:** Amirul  **Executed By:** Amirul  **Short Description:** Messaging System to communicate with instructors and student | **Test Case Name:** Message  **Subsystem:** Administrator  **Design Date: 13/4/2016**  **Execution Date: 20/4/2016** | |
| **Pre-Conditions:**  Database holds incoming and outgoing message | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comments** | | 1  2  3  4  5 | Click on Messages | Tables show list of message, if there is no message then it show no message | fail |  | | Click on Compose | A form displayed with reciever,subject and message fields | fail |  | | Input Blank fields | Validation Occurs | Pass | No Validation was implemented first | | Input valid values in compose | Successfully sent the message | Fail |  | | Delete items from inbox | Successfully deleted the items | fail |  | | | |
| **Post Conditions :**  The Message is reached to sender if composed  The Message is deleted Successfully if delete from inbox | | |

***12.2. Test Case 2.6***

|  |  |  |
| --- | --- | --- |
| **Test Case #:** 2.6  **System:** StudentManagementSystem  **Designed By:** Amirul  **Executed By:** Amirul  **Short Description:** View Administrator’s profile and provide facility to edit details | **Test Case Name:** Profile  **Subsystem:** Administrator  **Design Date: 13/4/2016**  **Execution Date: 20/4/2016** | |
| **Pre-Conditions:**  User must be registered in database | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comments** | | 1  2  3 | Click on Profile | Display User’s Current Profile , textboxes are not in editable mode | fail |  | | Click on edit button | Textboxes are not in editable mode | fail |  | | Input matched passwords and valid fields | Successfully updated profile | fail |  | | | |
| **Post Conditions :**  Shows user’s data and data is updated on database | | |

***12.2. Test Case 3***

|  |  |  |
| --- | --- | --- |
| **Test Case #:** 3.0  **System:** StudentManagementSystem  **Designed By:** Amirul  **Executed By:** Amirul  **Short Description:** A table shows student’s record and beside a textbox field to insert marks of students | **Test Case Name:** Upload Marks  **Subsystem:** Instructor  **Design Date: 13/4/2016**  **Execution Date: 20/4/2016** | |
| **Pre-Conditions:**  Students must register first | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comments** | | 1  2  3 | Click on Upload marks | List of students | fail |  | | Insert marks greater than 100 | Validation Error | fair |  | | Insert blank fields | Validation Error | Pass | Problem arise due to ambiguity | | | |
| **Post Conditions :**  Upload Marks of students successfully | | |

***12.2. Test Case 3.1***

|  |  |  |
| --- | --- | --- |
| **Test Case #:** 3.1  **System:** StudentManagementSystem  **Designed By:** Amirul  **Executed By:** Amirul  **Short Description:** Instructors can upload demo lectures | **Test Case Name:** Upload Lectures  **Subsystem:** Instructor  **Design Date: 13/4/2016**  **Execution Date: 20/4/2016** | |
| **Pre-Conditions:**  Database and sufficient storage | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comments** | | 1  2  3  4 | Click on Upload Lectures | Display previously uploaded lectures and come up with form to upload lecture | fail |  | | Input valid details | Video Uploaded | fail |  | | Input not supported format | Validation error | fail |  | | Input blank fields | Validation error | fail |  | | | |
| **Post Conditions :**  Video successfully uploaded and user able to watch uploaded video | | |

***12.2. Test Case 3.2***

|  |  |  |
| --- | --- | --- |
| **Test Case #:** 3.2  **System:** StudentManagementSystem  **Designed By:** Amirul  **Executed By:** Amirul  **Short Description:** Table come up with list of all students | **Test Case Name:** View Students  **Subsystem:** Instructors  **Design Date: 13/4/2016**  **Execution Date: 20/4/2016** | |
| **Pre-Conditions:**  Students must be sign up | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comments** | | 1  2 | Click on View Students | List of Students | fail |  | | Search student | Student Record related to search query | fail |  | | | |
| **Post Conditions :**  No Condtions | |

***12.2. Test Case 3.3***

|  |  |  |
| --- | --- | --- |
| **Test Case #:** 3.3  **System:** StudentManagementSystem  **Designed By:** Amirul  **Executed By:** Amirul  **Short Description:** Messaging System to communicate with instructors and student | **Test Case Name:** Message  **Subsystem:** Instructor  **Design Date: 13/4/2016**  **Execution Date: 20/4/2016** | |
| **Pre-Conditions:**  Database holds incoming and outgoing message | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comments** | | 1  2  3  4  5 | Click on Messages | Tables show list of message, if there is no message then it show no message | fail |  | | Click on Compose | A form displayed with reciever,subject and message fields | fail |  | | Input Blank fields | Validation Occurs | fail |  | | Input valid values in compose | Successfully sent the message | Fail |  | | Delete items from inbox | Successfully deleted the items | fail |  | | | |
| **Post Conditions :**  The Message is reached to sender if composed  The Message is deleted Successfully if delete from inbox | | |

***12.2. Test Case 3.4***

|  |  |  |
| --- | --- | --- |
| **Test Case #:** 3.4  **System:** StudentManagementSystem  **Designed By:** Amirul  **Executed By:** Amirul  **Short Description:** View Instructor’s profile and provide facility to edit details | **Test Case Name:** Profile  **Subsystem:** Instructor  **Design Date: 13/4/2016**  **Execution Date: 20/4/2016** | |
| **Pre-Conditions:**  User must be registered in database | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comments** | | 1  2  3 | Click on Profile | Display User’s Current Profile , textboxes are not in editable mode | fail |  | | Click on edit button | Textboxes are not in editable mode | fail |  | | Input matched passwords and valid fields | Successfully updated profile | fail |  | | | |
| **Post Conditions :**  Shows user’s data and data is updated on database | | |

***12.2. Test Case 4***

|  |  |  |
| --- | --- | --- |
| **Test Case #:** 4.0  **System:** StudentManagementSystem  **Designed By:** Amirul  **Executed By:** Amirul  **Short Description:** Student can register multiple offered courses | **Test Case Name:** Course Registration  **Subsystem:** Student  **Design Date: 13/4/2016**  **Execution Date: 20/4/2016** | |
| **Pre-Conditions:**  A Database that holds registration details | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comments** | | 1  2  3 | Click Course registration | List of Offered course | fail |  | | Checking multiple/Single option | Multiple or single option selected | fail |  | | Click Registration | Successfully Registered Course | fail |  | | | |
| **Post Conditions :**  Students are enrolled in course and are able to get aware of uploaded course marks. | | |

***12.2. Test Case 4.1***

|  |  |  |
| --- | --- | --- |
| **Test Case #:** 4.1  **System:** StudentManagementSystem  **Designed By:** Amirul  **Executed By:** Amirul  **Short Description:** Student can view their registered courses and able to de-register them. | **Test Case Name:** Registered Course  **Subsystem:** Student  **Design Date: 13/4/2016**  **Execution Date: 20/4/2016** | |
| **Pre-Conditions:**  Student must register course before view registered courses. | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comments** | | 1  2 | Click on Registered courses | List of Registered Courses | fail |  | | Click on De-Register Course | Course will be de-registered successfully | fail |  | | | |
| **Post Conditions :**  Student is able to register again after de-registering the course | | |

***12.2. Test Case 2***

|  |  |  |
| --- | --- | --- |
| **Test Case #:** 4.2  **System:** StudentManagementSystem  **Designed By:** Amirul  **Executed By:** Amirul  **Short Description:** Student will send feedback to administrator anonymously. | **Test Case Name:** Send Feedback  **Subsystem:** Student  **Design Date: 13/4/2016**  **Execution Date: 20/4/2016** | |
| **Pre-Conditions:**  Student must sign up for workshop. A Database to store feedback. | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comments** | | 1  2  3 | Click on Send Feedback | Feedback form will be shown | fail |  | | Enable multiple check boxes In same row | Unable to perform this action | fail |  | | Check one box in each row | Successfully send feedback | fail |  | | | |
| **Post Conditions :**  Student can only send one feedback | | |

***12.2. Test Case .3***

|  |  |  |
| --- | --- | --- |
| **Test Case #:** 4.3  **System:** StudentManagementSystem  **Designed By:** Amirul  **Executed By:** Amirul  **Short Description:** students can watch uploaded lectures by all instructors. | **Test Case Name:** Watch Lectures  **Subsystem:** Student  **Design Date: 13/4/2016**  **Execution Date: 20/4/2016** | |
| **Pre-Conditions:**  Lectures must be uploaded by instructors | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Step** | **Action** | **Expected System Response** | **Pass/Fail** | **Comments** | | 1  2 | Click on Watch lectures | A list of uploaded lectures | fail |  | | Click on linkbutton beside each lecture | Video Frame will popup | fail |  | | | |
| **Post Conditions :**  All Students are accessible of lectures whether they are uploaded or not | | |

# Major Deliverables

Following documents will be delivered at the end of test phase:

1. Test plan
2. Test cases
3. Test summary report

# Exit Criteria

Test phase will be considered complete if following criteria’s are met:

1. All test case results in success
2. No severe problems are identified
3. All functional requirement are working properly
4. All the problems identified are workable