|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Software Requirement Specifications**  [Student Portal for Azaan]  Version: [2.0]   |  |  | | --- | --- | | Supervisor | M. Asim Ali | | Co Supervisor (if any) |  | |  |  | | Project Team Members | Amirul Sunesara  Abu Bakar Siddiq  Aamir Shahzad | | Submission Date | 26/03/2016 | |

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

[Following table will contain list of people whom the document will be distributed after every sign-off]

|  |  |  |
| --- | --- | --- |
| **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 | 26/03/2016 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Table of Contents

Contents

[Document History 2](#_Toc446705984)

[Distribution List 3](#_Toc446705985)

[Document Sign-Off 4](#_Toc446705986)

[Table of Contents 5](#_Toc446705987)

[1. Introduction 7](#_Toc446705988)

[1.1 Purpose of Document 7](#_Toc446705989)

[1.2 Intended Audience 7](#_Toc446705990)

[1.3 Abbreviations 7](#_Toc446705991)

[1.4 Document Convention 8](#_Toc446705992)

[2. Overall System Description 9](#_Toc446705993)

[2.1 Project Background 9](#_Toc446705994)

[2.2 Project Scope 9](#_Toc446705995)

[2.3 Not In Scope (not applicable) 9](#_Toc446705996)

[2.4 Project Objectives 9](#_Toc446705997)

[2.5 Stakeholders 10](#_Toc446705998)

[2.6 Operating Environment 10](#_Toc446705999)

[2.7 System Constraints 10](#_Toc446706000)

[2.8 Assumptions & Dependencies 10](#_Toc446706001)

[3. Project Feasibility Analysis 11](#_Toc446706002)

[3.1 Economic Feasibility (not applicable) 11](#_Toc446706003)

[3.2 Technical Feasibility 11](#_Toc446706004)

[3.3 Operational Feasibility 11](#_Toc446706005)

[3.4 Schedule Feasibility 11](#_Toc446706006)

[3.5 Conclusion of Feasibility Analysis 11](#_Toc446706007)

[4. Primary and Secondary Research 12](#_Toc446706008)

[4.1 Primary Research not applicable) 12](#_Toc446706009)

[4.1.1 Observation not applicable) 12](#_Toc446706010)

[4.1.2 Interviews not applicable) 12](#_Toc446706011)

[4.1.3 Rationale for Interviews not applicable) 12](#_Toc446706012)

[4.2 Academic Research 12](#_Toc446706013)

[4.2.1 Development Methodology 12](#_Toc446706014)

[4.2.2 Development Tools (font end and Back end) 12](#_Toc446706015)

[4.2.3 Secondary Research(not applicable) 13](#_Toc446706016)

[5. Context Diagram 14](#_Toc446706017)

[6. External Interface Requirements 15](#_Toc446706018)

[6.1 Hardware Interfaces 15](#_Toc446706019)

[6.2 Software Interfaces 15](#_Toc446706020)

[6.3 Communications Interfaces 16](#_Toc446706021)

[7. Functional Requirements 17](#_Toc446706022)

[7.1 Functional Hierarchy 17](#_Toc446706023)

[7.2 List of all functional requirements 18](#_Toc446706024)

[7.3 List of Actors and Use Cases 19](#_Toc446706025)

[7.3.1 Use case diagram and use case description 19](#_Toc446706026)

[8. Proposed Objects List from Use Cases 20](#_Toc446706027)

[9. Initial Class Diagram 33](#_Toc446706028)

# Introduction

## 1.1 Purpose of Document

The product is a web portal to be built as a course management system for an institute called Azaan. This product aims to provide solution to institute by making its management of its most precious resources, student easier. Portal is built to bridge the gap between student and instructors and to make interaction between them easier. It also aims to provide the organization with a platform from where it can overlook its resources remotely.

## 1.2 Intended Audience

As its name states software resource specification is a made to clarify the requirements that are demanded by the client. It defines all the requirements of the client in every aspect. This document is made for the software developing team and is intended to be used by following people:

1. Designers

To get correct set of requirements on which they can design the software.

1. Developers(programmers)

It can be used as a reference to requirements.

1. Testing team

Testing team can use it to verify or match the end product with requirements.

1. Client

To verify that the requirement specifications match what he/she demanded.

To get a clear understanding of the document it is recommended to study it in its organized manner.

## 1.3 Abbreviations

Not Used

## 

## 1.4 Document Convention

|  |  |
| --- | --- |
| **Name** | **Properties** |
| Main Heading | Bold, Italic, Left aligned, Arial 16pt |
| Sub Heading | Bold, Italic, Left aligned, Arial 12pt |
| Text | Arial 12pt, Justified align |

# Overall System Description

## Project Background

Need for such an application was felt because our client organization was expanding an needs its resources to be available at any moment under one umbrella so we decided to develop a web application. The application must be multiplatform compatible and should be free from all external dependencies.

## Project Scope

This application aims to provide an online course management system to our client. It will maintaining database for its students which will used to keep record of things such as:

1. Number of students enrolled
2. Exam marks of student
3. Personal details of student

Portal will also be used as a communicative tool between students and instructors. Communication or interaction can be done by the help of following tools find in the application:

1. Feedback forms(students will be using it to give feedback to instructors)
2. In-built messaging system

Reason behind developing a web portal is to control resources which are scattered out and remote. Web portal will allow instructor to keep in touch with his students no matter wherever he is as he can upload lectures, marks and communicate with them using messaging system.

## Not In Scope (not applicable)

## Project Objectives

The product is a web portal to be built as a course management system for an institute called Azaan. This product aims to provide solution to institute by making its management of its most precious resources, student easier. Portal is built to bridge the gap between student and instructors and to make interaction between them easier. It also aims to provide the organization with a platform from where it can overlook its resources remotely.

## Stakeholders

Following are the stakeholders of the system:

* Students: Students will be the end user of the system. Whole system is developed for their convince. Any change in system will affect students.
* Administrator: Administrator is a person responsible for handling the system. He will be the one controlling system from the back once it is deployed.
* Instructor: instructor is also one of the end users for the system. His domain will include updating marks, uploading videos and messaging.

## Operating Environment

Software needs to be a web based portal which can be operated on an average browser using average internet connection. Pages must be kept light in content so that their content size can be reduced. Due to vast use of mobile phone usage in current era, application is expected to be used on mobile phone browsers so web pages must be optimized in a manner so that the content can be visible on small screens.

## System Constraints

Following constraints must be considered while developing the application:

1. It should be designed and develop in a way so that its load time can be minimum.
2. Development must not take time of more than 2.5 months.
3. Cost affective technologies must not be taken into account due to budget issues.
4. No developer/developing firms trademark, advertising or credit notes to place on part of the application

## Assumptions & Dependencies

Following are the assumptions and dependencies for the system:

* It is assumed that every user contains basic knowledge regarding use of computers and browser.
* As the system is web-based application its usage will be dependent on availability of internet, hence every user is assumed to have active internet connection.

# Project Feasibility Analysis

## Economic Feasibility (not applicable)

## Technical Feasibility

This project will be built using following technologies for their respective purposes:

|  |  |
| --- | --- |
| Technology | Use |
| Bootstrap | Styling |
| Asp.net | Web programming |
| Sql server | database |

Development team contains specialist in all of the above fields so their will be no hindrance will developing application using such technologies hence it will be feasible to develop this application technically.

## Operational Feasibility

Operational results of the system will be highly productive as it will be resulting in gathering in all the resources under one umbrella that can be visited remotely. Application will also make working of organization more fast hence resulting in more productivity. As the system is leading towards more productivity it is said to operational feasible.

## Schedule Feasibility

Client has given development team a time of 12 weeks (3 months approx.) which will be divided by as follows:

|  |  |
| --- | --- |
| Stage | Time |
| Analysis | 2 weeks |
| Planning | 3 weeks |
| Development | 4weeks |
| Testing | 1 week |
| Deployment | 1 week |
| Reserve Time | 1 week |
| Total time | 12 weeks |

As the above table shows that the project can be completed within given time period with a reserve week as well so scheduling this project will be feasible.

## Conclusion of Feasibility Analysis

As the system is feasible in every aspect towards the developers, it can be staed to be feasible for development. It is providing developer with economic benefits, experience, furnishing their skills and providing numerous benefits to client hence it feasible to be developed.

# Primary and Secondary Research

## Primary Research not applicable)

### Observation ( not applicable)

### Interviews (not applicable)

### Rationale for Interviews (not applicable)

## Academic Research

### Development Methodology

#### Appropriate Methodologies available

Following methodologies were available for developing application:

* Waterfall
* Agile
* Prototyping

#### Selection of Methodology

We will be using agile methodology to develop our application.

#### Methodology Reasoning

Reason behind adopting agile methodology is that it is easy to adopt and results in rapid application development. It reduces the time required to develop and application significantly as it doesn’t require documentation for every minor step. Another reason for adopting agile is that it is one of the finest and latest methods currently being used by industry, so in order to keep our projects in line with industry trends , we will be using agile.

### Development Tools (font end and Back end)

#### Comparison of Different Tools

Tools available for database:

* Sql server
* MySQL
* Oracle

Tools available for web programming;

* Asp.net
* Jsp
* Php

Tools available for styling:

* Simple css
* Bootstrap

#### Selection of Appropriate Tool

Following tools are choose for their respective purposes:

* Database(Microsoft SQL server 2013)
* Web programming(asp.net)
* Styling (bootstrap).
  + - 1. **Development Tool Reasoning**

Reason behind choosing every technology/tool is mentioned below:

* Sqlserver:

SQL server is choose over its available alternatives because of its compatibility with asp.net as both can be used under same window while developing. Another reason behind choosing SQL server is its compact size as compared to its available alternatives like oracle which comparatively

Heavy to handle as it utilizes more space.

* Asp.net:

Asp.net is a technology, easily to implement and maintain and produces light weight webpages with higher features. Unlike its alternates such as jsp and php it is easy to program and maintain asp.net. It also contains less external dependencies as compared to it competitors.

* Bootstrap:

In the current era use of mobile devices is increasing with every passing day so it is compulsory that web application must have a clear view on mobile devices hence bootstrap is used for this purposes as it alters webpage styling once it is opened on mobile device.

### Secondary Research (not applicable)

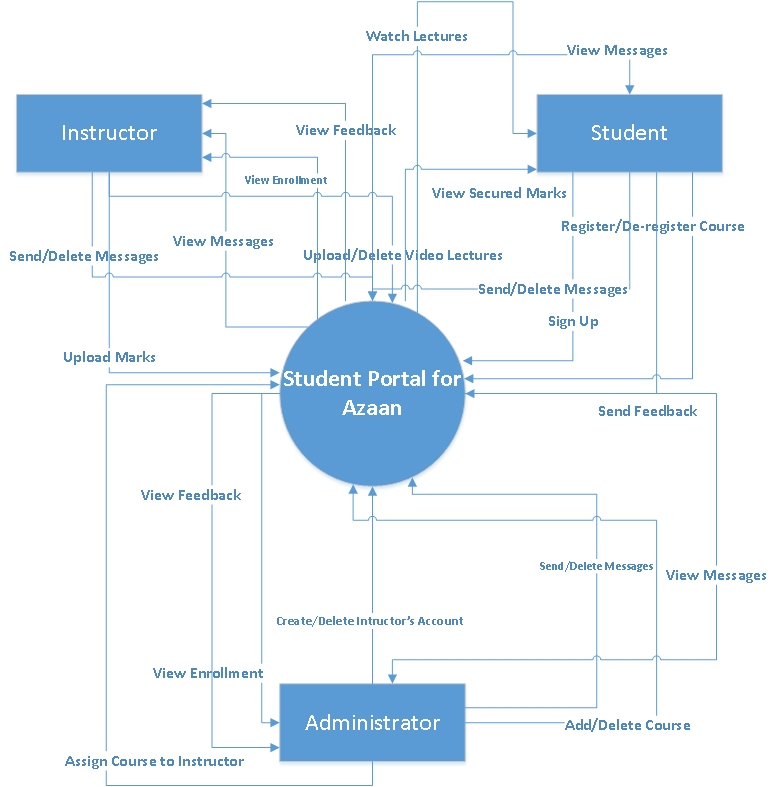
#### Books (not applicable)

#### Web URLs ( not applicable)

#### Periodicals ( not applicable)

#### Journals (not applicable)

# Context Diagram



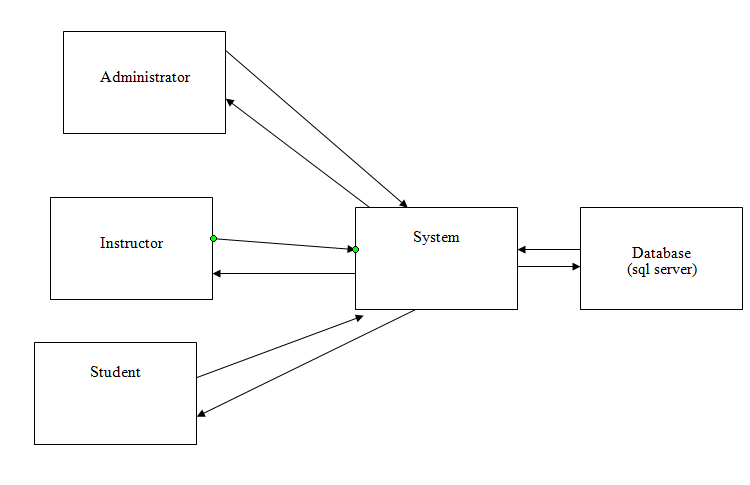
]

# External Interface Requirements

## Hardware Interfaces

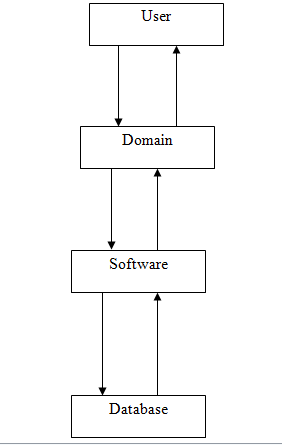
As the software is web application its hardware requirements are comparatively low and it doesn’t require any specific hardware device to function properly. Application will be perfectly operating with basic hardware requirements.

## Software Interfaces



As the diagram illustrates the interface of software and its components, the users will be communicating to the system which will then be fetching, inserting and updating data in database.

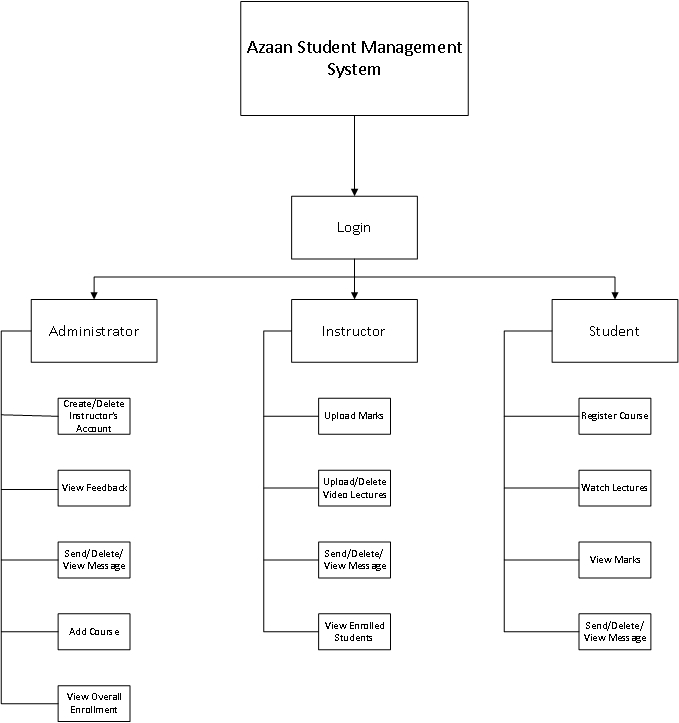
## Communications Interfaces



The diagram above shows the communication model/interface of the application. As prominent in the diagram, the dataflow will be will have database on one hand and end user on the other. User will be using application hosted on a domain/ server by using a suitable browser. if a user requests any data from database or prompts for its updation domain will be doing it in database.

# Functional Requirements

## Functional Hierarchy



## List of all functional requirements

1. Administrator

Administrator account can only be created by the programmer as this account is limited in number. Administer can be said the master of whole system as he can overlook every activity. Administrator will have privileges to add/delete instructor, add new course and review feedback report.

2. Instructor

Instructor account will be created by administrator. Courses will be assigned to instructor by administrator and he will have rights to upload marks, upload video lectures, view students and review feedback report. Instructor will also be using messaging protocol to communicate with students

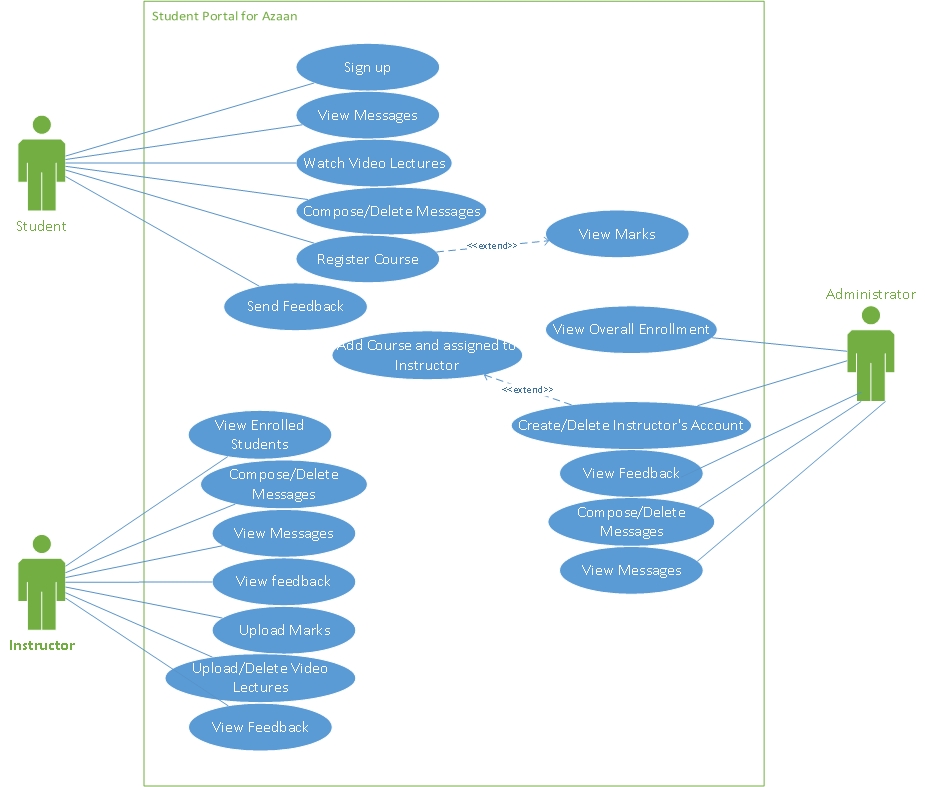
3. Student

Students are the main art of the system around which it is built. Any person can get an student account using signup form available on homepage. Once signup form is filled successfully, student will be provided with login credentials.

Upon successful login user can register new courses, view his previous marks, communicate with instructor using messaging system and give feedback regarding his experiences.

## List of Actors and Use Cases

### Use case diagram and use case description



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<1: Sign Up>** | | | | |
| **Use case Id:** | | 1 | | |
| **Actors:**  Student | | | | |
| **Feature:** Student Sign Up Procedure | | | | |
| **Pre-condition:** | | N.A | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Students fills signup form | | | Successfully sign up, login details enter in database |
| **2.** | Students fills signup form with invalid inputs | | | Validation error |
|  |  | | |  |
| **Alternate Scenarios:** | | | | |
| **1a: N.A**    **2a:** Re-fill signup form with correct inputs | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1** | Student can login after signup process | | | |
|  |  | | | |
|  |  | | | |
| **Use Case Cross referenced** | | | N.A | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<2: Instructor Sign Up>** | | | | |
| **Use case Id:** | | 2 | | |
| **Actors:**  Administrator | | | | |
| **Feature:** Creation of instructor’s account by Administrator authority | | | | |
| **Pre-condition:** | | Administrator must be logged in | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Administrator fills correct signup form | | | Successfully sign up, login details enter in database |
| **2.** | Administrator fills signup form with invalid inputs | | | Validation error |
|  |  | | |  |
| **Alternate Scenarios:** | | | | |
| **1a:** N.A    **2a:** Re-fill the form with correct inputs | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1** | Instructor can login after signup process | | | |
|  |  | | | |
|  |  | | | |
| **Use Case Cross referenced** | | | N.A | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<3: Add Course >** | | | | |
| **Use case Id:** | | 3 | | |
| **Actors:**  Administrator | | | | |
| **Feature:** Add Course and assign an instructor to it | | | | |
| **Pre-condition:** | | Instructor’s account must be created in order to assign course | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Type course name | | | - |
| **2.** | Select instructor name from dropdown menu | | | Name is selected |
| **3.** | Submit | | | Instructor is successfully assigned with course |
| **Alternate Scenarios:** | | | | |
| **1a:** If invalid or empty course name is entered, validation error occurs.    **2a:** if instructor name is set to default option, validation error occurs. | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1** | Instructor will be able to view student enrollment. | | | |
| **2** | Instructor can upload marks. | | | |
|  |  | | | |
| **Use Case Cross referenced** | | | 2 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<4: Compose/Delete Messages>** | | | | |
| **Use case Id:** | | 4 | | |
| **Actors:**  Instructor, Administrator, Student | | | | |
| **Feature:** User can send messages to each other and delete messages form inbox | | | | |
| **Pre-condition:** | | User must be logged in | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Compose message | | | Message sent to receiver |
| **2.** | Delete message | | | Message deleted from inbox |
|  |  | | |  |
| **Alternate Scenarios:** | | | | |
| **1a:**Select a Person which categorize different users.  **1b:**Select name of person  **1c:** Enter Subject, message body  **1d:** Send Message  **1e:** Invalid Form will cause validation error  **2a:**Check Messages you wish to delete  **2b:**Delete Selected Messages | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1** | Inbox must be updated | | | |
|  |  | | | |
|  |  | | | |
| **Use Case Cross referenced** | | | N.A | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<5: View Messages>** | | | | |
| **Use case Id:** | | 5 | | |
| **Actors:**  Student, Administrator, Instructor | | | | |
| **Feature:** Users can view inbox | | | | |
| **Pre-condition:** | | User must be logged in | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | View Message | | | Display Inbox |
|  |  | | |  |
|  |  | | |  |
| **Alternate Scenarios:** | | | | |
| **1a:** N.A | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **N.A** | N.A | | | |
|  |  | | | |
|  |  | | | |
| **Use Case Cross referenced** | | | N.A | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<6: Register Course>** | | | | |
| **Use case Id:** | | 6 | | |
| **Actors:**  Student | | | | |
| **Feature:** Student register course that is offered by administration | | | | |
| **Pre-condition:** | | Instructor must be assigned with course | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Check a course | | | Course checked |
| **2.** | Register course | | | Successful registration |
|  |  | | |  |
| **Alternate Scenarios:** | | | | |
| **1a:** If none is checked, validation error. Repeat step 1    **2a:** N.A | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1** | Students can send feedback after registration | | | |
|  |  | | | |
|  |  | | | |
| **Use Case Cross referenced** | | | 3 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<7: Watch Video Lectures>** | | | | |
| **Use case Id:** | | 7 | | |
| **Actors:**  Students | | | | |
| **Feature:** Students can watch lectures that is uploaded by instructors | | | | |
| **Pre-condition:** | | Lectures must be uploaded | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Watch lectures | | | Display lecture |
|  |  | | |  |
|  |  | | |  |
| **Alternate Scenarios:** | | | | |
| **1a: N.A** | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **N.A** | N.A | | | |
|  |  | | | |
|  |  | | | |
| **Use Case Cross referenced** | | | <Related use cases, which use or are used by this use case> | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<8:Send Feedback>** | | | | |
| **Use case Id:** | | 8 | | |
| **Actors:**  Students | | | | |
| **Feature:** Students can send feedback to instructors and administrators about course progress | | | | |
| **Pre-condition:** | | Students must be registered | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Check appropriate remarks | | | Options checked |
| **2.** | Submit feedback | | | Feedback submitted |
|  |  | | |  |
| **Alternate Scenarios:** | | | | |
| **2a:** if an option is unchecked, validation error occurs | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **1** | Student cannot re-submit feedback | | | |
|  |  | | | |
|  |  | | | |
| **Use Case Cross referenced** | | | N.A | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<9: Upload/View Marks>** | | | | |
| **Use case Id:** | | 9 | | |
| **Actors:**  Instructor, Student | | | | |
| **Feature:** After tests instructors can upload marks and student can view marks | | | | |
| **Pre-condition:** | | User must be logged in , student must be registered , instructor must be assigned with course | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Instructor Upload the marks | | | Marks uploaded |
| **2.** | Student view secured marks | | | Display marks |
|  |  | | |  |
| **Alternate Scenarios:** | | | | |
| **1a:** Marks must be less than equal to 100 or greater equal to 0  **1b:** Validation error occurs if empty space if left or step 1a is not followed  **2a:** Notify if marks is not uploaded. | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **N.A** | N.A | | | |
|  |  | | | |
|  |  | | | |
| **Use Case Cross referenced** | | | 3 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<10: Upload/Delete Video Lectures>** | | | | |
| **Use case Id:** | | 10 | | |
| **Actors:**  Instructors | | | | |
| **Feature:** Instructors can upload video lectures that can be seen by students | | | | |
| **Pre-condition:** | | User must be logged in | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | Upload Lecture | | | Lecture uploaded |
| **2.** | Delete Lecture | | | Lecture Deleted |
|  |  | | |  |
| **Alternate Scenarios:** | | | | |
| **1a:** Enter name and details of video  **1b:**Choose video file path  **1c:** Any format of file is supported  **1d:** Validation error if above is not correctly input  **2a: N.A** | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **N.A** | N.A | | | |
|  |  | | | |
|  |  | | | |
| **Use Case Cross referenced** | | | N.A | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<11: View Enrolled Students>** | | | | |
| **Use case Id:** | | 11 | | |
| **Actors:**  Instructor | | | | |
| **Feature:** Instructors can view enrolled students data in their course | | | | |
| **Pre-condition:** | | User must be logged in, instructors must be assigned with course | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | View Enrolled Students | | | Display Enrolled Students |
|  |  | | |  |
|  |  | | |  |
| **Alternate Scenarios:** | | | | |
| **1a:** Instructor can search students by their name | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **N.A** | N.A | | | |
|  |  | | | |
|  |  | | | |
| **Use Case Cross referenced** | | | N.A | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<12: View Feedback>** | | | | |
| **Use case Id:** | | 12 | | |
| **Actors:**  Instructor, Administrators | | | | |
| **Feature:** Users can view statistics of feedback | | | | |
| **Pre-condition:** | | User must be logged in | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | View Feedback | | | Display feedback |
|  |  | | |  |
|  |  | | |  |
| **Alternate Scenarios:** | | | | |
| **1a: N.A** | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **N.A** | N.A | | | |
|  |  | | | |
|  |  | | | |
| **Use Case Cross referenced** | | | N.A | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<13: View Overall Enrollment>** | | | | |
| **Use case Id:** | | 13 | | |
| **Actors:**  Administrator | | | | |
| **Feature:** User can view students intended to participate in courses | | | | |
| **Pre-condition:** | | User must be logged in | | |
| **Scenarios** | | | | |
| **Step#** | **Action** | | | **Software Reaction** |
| **1.** | View Enrollment | | | Display enrolled students |
|  |  | | |  |
|  |  | | |  |
| **Alternate Scenarios:** | | | | |
| **1a:** User can also search for student | | | | |
| **Post Conditions** | | | | |
| **Step#** | **Description** | | | |
| **N.A** | N.A | | | |
|  |  | | | |
|  |  | | | |
| **Use Case Cross referenced** | | | N.A | |

# Proposed Objects List from Use Cases

1. Users
2. Feedback
3. Marks
4. Video Lectures
5. Student
6. Administrator
7. Instructor
8. Registration
9. Enrollment
10. Communication

# Initial Class DiagramC:\Users\Ami\Desktop\Class.jpg

# List of Non-functional Requirements

* 1. Performance Requirements

Following process requirements must be accomplished by the application:

* As being a web application one of the main constraints or one of the worst scenarios is the slow internet connection which occurs occasionally due to various reasons. Portal should be designed in a way that slow internet connectivity affects its performance.
* On the other hand waiting time after a click annoys the user; therefore system should be designed in highly responsive manner.
  1. Safety Requirements

Application must be designed to be safe, which means that its interactions with any sort of unrequired third party systems or websites should be avoided. Interaction with unwanted applications might expose system and its users to danger in the following ways:

* Users platform might be exposed to viruses and other malfunctions
* It might provide mischief makers to breach the security of the system
* It might slow down the performance of the system
* It will affect the usability of the software
  1. Security Requirements

Data is one of the most important assets to any organization. Looking to this, overall security for the portal must be built or developed. Access of any unauthorized person to data or in the portal must be restrained. Person must be automatically logged off once he or she closes his browser.

* 1. User Documentation

Following documents will be delivered with the delivery of software:

* User manual
* Video guide(if required).

1. References (Not Applicable)
2. Appendices (Not Applicable)