

# **Student World**

**Final Year Project**

**Session 2015-2019**

A project submitted in partial fulfillment of the degree of

BS in Software Engineering



Department of Software Engineering

Faculty of Computer Science & Information Technology

The Superior College, Lahore

Spring 2019

|                              |                                                                                                                                                     |                   |                          |            |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|--------------------------|------------|
| Type (Nature of project)     | [ <input checked="" type="checkbox"/> ] <b>Development</b> [ <input type="checkbox"/> ] <b>Research</b> [ <input type="checkbox"/> ] <b>R&amp;D</b> |                   |                          |            |
| Area of specialization       | <b>WED DEVELOPMENT</b>                                                                                                                              |                   |                          |            |
| FYP ID                       | <b>FYP-BSSE-F18-021</b>                                                                                                                             |                   |                          |            |
| <b>Project Group Members</b> |                                                                                                                                                     |                   |                          |            |
| Sr.#                         | Reg. #                                                                                                                                              | Student Name      | Email ID                 | *Signature |
| (i)                          | BSEM-F15-017                                                                                                                                        | Hafiz Usman Ilyas | Usman.ilyas01@gmail.com  |            |
| (ii)                         | BSEM-F15-058                                                                                                                                        | Nida Gulzar       | Nidagulzar7462@gmail.com |            |
| (iii)                        |                                                                                                                                                     |                   |                          |            |

\*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

### Plagiarism Free Certificate

This is to certify that, I **Hafiz Usman Ilyas** S/D of Muhammad Ilyas, group leader of FYP under registration no **BSEM-F15-017** at Software Engineering Department, The Superior College, Lahore. I declare that my FYP report is checked by my supervisor.

Date: 22-08-2019      Name of Group Leader: **Hafiz Usman Ilyas**      Signature: \_\_\_\_\_

Name of Supervisor: **Dr. Sheheryar Malik Farhat**

Co-Supervisor: **Ma'am Tayyaba**

Designation: Professor

Designation: Lecturer

Signature: \_\_\_\_\_

Signature: \_\_\_\_\_

HoD: Dr. Arfan Jaffar

## Change Record

| Author(s)   | Version | Date        | Notes                                                | Supervisor's Signature |
|-------------|---------|-------------|------------------------------------------------------|------------------------|
| Usman, Nida | 1.0     | 20 DEC 2018 | <Original Draft>                                     |                        |
| Usman, Nida | 1.1     | 07 JAN 2019 | Changes in SRS,<br>Functional Requirements<br>Format |                        |
| Usman, Nida | 1.1     | 07 JAN 2019 | Changes in UML<br>Diagrams                           |                        |
|             |         |             | <Added Project Plan>                                 |                        |
|             |         |             | <Changes Based on<br>Feedback from<br>Supervisor>    |                        |
|             |         |             |                                                      |                        |
|             |         |             |                                                      |                        |
|             |         |             |                                                      |                        |
|             |         |             |                                                      |                        |
|             |         |             |                                                      |                        |

## APPROVAL

---

### PROJECT SUPERVISOR

Comments: \_\_\_\_\_

\_\_\_\_\_

Name: \_\_\_\_\_

Date: \_\_\_\_\_ Signature: \_\_\_\_\_

\_\_\_\_\_

### PROJECT MANAGER

Comments: \_\_\_\_\_

\_\_\_\_\_

Date: \_\_\_\_\_ Signature: \_\_\_\_\_

### HEAD OF THE DEPARTMENT

Comments: \_\_\_\_\_

\_\_\_\_\_

Date: \_\_\_\_\_ Signature: \_\_\_\_\_

## **Dedication**

*I want to dedicate this work to those students who wants to learn sand do something even though they have no resources. I also want to dedicate this project to all of my friends who have crossed this stage without any resources and faced these types of difficulties.*

## **Acknowledgements**

All thanks and gesture of recognition are to Almighty Allah, who grant us with the wisdom, quality, learning and bravery to finish this errand. We might want to pay huge amounts of thanks and offer my regards to our Supervisor Dr. Sheheryar Malik for his valuable direction, advises and support all through the learning procedure that he gave us a helpful course to finish our venture.

## **Executive Summary**

We are developing a source that works on Website. This source totally depends on facilitating students in Study and other Educational Material. In this project we provide students with a specific source so that he/she can gather their study material from a single source and don't have to use other sources like Google Drive, Google Classroom and Dropbox etc. It provides feature in which every student of a particular class can communicate with each other in a single chat room. It provides students with extra materials of their course like video tutorials, lectures and books etc. A Particular student can provide his/her own lectures if he wants too and he also can help other students in study. A main benefit for this student is that anybody can hire him/her as a tutor. The main feature is to notify every student on their mobile numbers other than email whenever someone uploads the data.

## Table of Contents

|                                                       |    |
|-------------------------------------------------------|----|
| Dedication.....                                       | 5  |
| Acknowledgements .....                                | 6  |
| Executive Summary .....                               | 7  |
| List of Figures .....                                 | 11 |
| List of Tables .....                                  | 12 |
| Chapter 1 .....                                       | 14 |
| Introduction .....                                    | 14 |
| Chapter 1: Introduction.....                          | 15 |
| 1.1. Background.....                                  | 15 |
| 1.2. Motivations and Challenges .....                 | 15 |
| 1.3. Goals and Objectives .....                       | 16 |
| 1.4. Literature Review/Existing Solutions .....       | 17 |
| 1.5. Proposed Solution .....                          | 17 |
| 1.6. Project Plan.....                                | 17 |
| 1.6.1. Work Breakdown Structure .....                 | 18 |
| 1.7. Roles & Responsibility Matrix .....              | 19 |
| 1.8. Gantt Chart .....                                | 20 |
| Chapter 2 .....                                       | 21 |
| Software Requirement Specifications .....             | 21 |
| 2.1. Introduction .....                               | 22 |
| 2.1.1. Purpose .....                                  | 22 |
| 2.1.2. Document Conventions .....                     | 22 |
| 2.1.3. Intended Audience and Reading Suggestions..... | 22 |
| 2.1.4. Product Scope.....                             | 22 |
| 2.1.5. References.....                                | 22 |
| 2.2. Overall Description.....                         | 23 |
| 2.2.1. Product Perspective .....                      | 23 |
| 2.2.2. Product Functions.....                         | 23 |
| 2.2.3. User Classes and Characteristics.....          | 27 |
| 2.2.4. Operating Environment .....                    | 27 |
| 2.2.5. Design and Implementation Constraints.....     | 27 |
| 2.2.6. User Documentation.....                        | 27 |
| 2.2.7. Assumptions and Dependencies.....              | 28 |
| 2.3. External Interface Requirements .....            | 28 |
| 2.3.1. User Interfaces .....                          | 28 |
| 2.3.2. Hardware Interfaces.....                       | 30 |
| 2.3.3. Software Interfaces.....                       | 30 |
| 2.3.4. Communications Interfaces .....                | 30 |
| 2.4. System Features .....                            | 30 |
| 2.4.1. System Feature 1 .....                         | 32 |
| 2.4.2. System Feature 2 .....                         | 32 |
| 2.4.3. System Feature 3.....                          | 32 |
| 2.4.4. System Feature 4.....                          | 33 |
| 2.4.5. System Feature 5.....                          | 33 |
| 2.4.6. System Feature 6.....                          | 34 |
| 2.4.7. System Feature 7 .....                         | 34 |
| 2.4.8. System Feature 8.....                          | 34 |



|                                                             |    |
|-------------------------------------------------------------|----|
| 2.4.9. System Feature 9 .....                               | 35 |
| 2.5. Other Nonfunctional Requirements .....                 | 35 |
| 2.5.1. Performance Requirements .....                       | 35 |
| 2.5.2. Safety Requirements .....                            | 36 |
| 2.5.3. Security Requirements .....                          | 36 |
| 2.5.4. Software Quality Attributes .....                    | 36 |
| The framework won't be separated effortlessly. ....         | 36 |
| 2.5.5. Business Rules .....                                 | 36 |
| 2.6. Other Requirements .....                               | 36 |
| Chapter 3 .....                                             | 39 |
| Use Case Analysis .....                                     | 39 |
| 3.1. Use Case Model .....                                   | 40 |
| 3.2. Fully Dressed Use Cases .....                          | 41 |
| Chapter 4 .....                                             | 53 |
| System Design .....                                         | 53 |
| 4.1. Architecture Diagram .....                             | 54 |
| 4.2. Domain Model .....                                     | 54 |
| 4.3. Entity Relationship Diagram with data dictionary ..... | 56 |
| 4.4. Class Diagram .....                                    | 59 |
| 4.5. Sequence / Collaboration Diagram .....                 | 60 |
| 4.6. Operation contracts .....                              | 67 |
| 4.6.1 Operation Contract#1 .....                            | 67 |
| 4.6.2 Operation Contract#2 .....                            | 67 |
| 4.6.3 Operation Contract#3 .....                            | 67 |
| 4.6.4 Operation Contract#4 .....                            | 68 |
| 4.6.5 Operation Contract#5 .....                            | 68 |
| 4.6.6 Operation Contract#6 .....                            | 69 |
| 4.6.7 Operation Contract#7 .....                            | 69 |
| 4.6.8 Operation Contract#8 .....                            | 69 |
| 4.6.9 Operation Contract#9 .....                            | 70 |
| 4.6.10 Operation Contract#10 .....                          | 70 |
| 4.6.11 Operation Contract#11 .....                          | 70 |
| 4.6.12 Operation Contract#12 .....                          | 71 |
| 4.7. Activity Diagram .....                                 | 72 |
| 4.8. State Transition Diagram .....                         | 73 |
| 4.9. Component Diagram .....                                | 74 |
| 4.10. Deployment Diagram .....                              | 75 |
| Chapter 5 .....                                             | 76 |
| Implementation .....                                        | 76 |
| 5.1. Important Flow Control/Pseudo codes .....              | 78 |
| 5.2. Components, Libraries, Web Services and stubs .....    | 79 |
| 5.3. Deployment Environment .....                           | 79 |
| 5.4. Tools and Techniques .....                             | 79 |
| 5.5. Best Practices / Coding Standards .....                | 80 |
| Chapter 6 .....                                             | 81 |
| Testing and Evaluation .....                                | 81 |
| 6.1. Use Case Testing .....                                 | 82 |
| 6.2. Equivalence partitioning .....                         | 91 |
| 6.3. Boundary value analysis .....                          | 91 |
| 6.4. Data flow testing .....                                | 93 |

|                                                      |     |
|------------------------------------------------------|-----|
| 6.5. Unit testing .....                              | 95  |
| 6.5.1 UNIT TESTING.....                              | 95  |
| 6.5.2 USABILITY TESTING.....                         | 98  |
| 6.5.3 COMPATIBILITY TESTING .....                    | 99  |
| 6.6. Integration testing .....                       | 99  |
| 6.6.1 SYSTEM TESTING.....                            | 99  |
| 6.6.2 SECURITY TESTING .....                         | 99  |
| 6.7. Performance testing.....                        | 100 |
| 6.8. Stress Testing.....                             | 100 |
| Chapter 7 .....                                      | 101 |
| Summary, Conclusion and Future Enhancements .....    | 101 |
| 7.1. Project Summary.....                            | 102 |
| 7.2. Achievements and Improvements.....              | 102 |
| 7.3. Lessons Learnt .....                            | 102 |
| 7.4. Future Enhancements/Recommendations .....       | 103 |
| Appendices .....                                     | 104 |
| Appendix A: Information / Promotional Material ..... | 105 |
| Reference and Bibliography .....                     | 117 |

## List of Figures

|                                                     |    |
|-----------------------------------------------------|----|
| Figure 1 Gantt chart .....                          | 20 |
| Figure 2 GUI 1.....                                 | 28 |
| Figure 3 GUI 2.....                                 | 28 |
| Figure 4 GUI 3.....                                 | 29 |
| Figure 5 Communication Interfaces.....              | 30 |
| Figure 6 Use Case Diagram.....                      | 41 |
| Figure 7 Architecture Diagram .....                 | 54 |
| Figure 8 Domain Model .....                         | 55 |
| Figure 9 Entity Relationship diagram .....          | 56 |
| Figure 10 Class diagram.....                        | 59 |
| Figure 11 Sequence Diagram Registration.....        | 60 |
| Figure 12 Sequence Diagram Login .....              | 61 |
| Figure 13 Sequence Diagram Logout .....             | 62 |
| Figure 14 Sequence Diagram Add/View Subject .....   | 62 |
| Figure 15 Sequence Diagram Add/View Lecture.....    | 63 |
| Figure 16 Sequence Diagram Add/View Video .....     | 64 |
| Figure 17 Sequence Diagram View Tutor.....          | 65 |
| Figure 18 Sequence Diagram Send Notifications ..... | 65 |
| Figure 19 Sequence Diagram Admin Side .....         | 66 |
| Figure 20 Activity Diagram.....                     | 72 |
| Figure 21 State transition diagram .....            | 73 |
| Figure 22 Component diagram .....                   | 74 |
| Figure 23 deployment diagram .....                  | 75 |
| Figure 24 Use Case Testing Register .....           | 83 |
| Figure 25 Use Case Testing Login.....               | 84 |
| Figure 26 Use Case Testing Add Course .....         | 85 |
| Figure 27 Use Case Testing Add Lecture .....        | 86 |
| Figure 28 Use Case Testing View Lecture.....        | 86 |
| Figure 29 Use Case Testing Add Video.....           | 87 |
| Figure 30 Use Case Testing View Video .....         | 88 |
| Figure 31 Use Case Testing Chatroom .....           | 88 |
| Figure 32 Use Case Testing Add Notification.....    | 89 |
| Figure 33 Use Case Testing View Notification.....   | 90 |
| Figure 34 Use Case Testing View Tutors .....        | 90 |
| Figure 35 Data Flow Testing Login.....              | 93 |
| Figure 36 Data Flow Testing Register .....          | 94 |

## List of Tables

|                                                             |    |
|-------------------------------------------------------------|----|
| Table 1 GUI Screens .....                                   | 29 |
| Table 2 Function Requirement Register.....                  | 32 |
| Table 3 Function Requirement Login .....                    | 32 |
| Table 4 Function Requirement Logout .....                   | 33 |
| Table 5 Function Requirement Select Subject.....            | 33 |
| Table 6 Function Requirement View Detail.....               | 34 |
| Table 7 Function Requirement Add Lecture .....              | 34 |
| Table 8 Function Requirement View Lecture .....             | 34 |
| Table 9 Function Requirement Delete Lecture.....            | 35 |
| Table 10 Function Requirement Add Video .....               | 35 |
| Table 11 Fully Dressed Use Case Register .....              | 43 |
| Table 12 Fully Dressed Use Case Login.....                  | 44 |
| Table 13 Fully Dressed Use Case Logout .....                | 44 |
| Table 14 Fully Dressed Use Case Add Subject .....           | 45 |
| Table 15 Fully Dressed Use Case View Subject .....          | 46 |
| Table 16 Fully Dressed Use Case Add Lecture.....            | 47 |
| Table 17 Fully Dressed Use Case View Lecture .....          | 48 |
| Table 18 Fully Dressed Use Case Add Video .....             | 49 |
| Table 19 Fully Dressed Use Case View Lecture .....          | 50 |
| Table 20 Fully Dressed Use Case Send Notification .....     | 51 |
| Table 21 Fully Dressed Use Case Tutor.....                  | 51 |
| Table 22 Fully Dressed Use Case View Tutor .....            | 52 |
| Table 23 Data Dictionary User .....                         | 57 |
| Table 24 Data Dictionary Admin.....                         | 57 |
| Table 25 Data Dictionary Subject.....                       | 57 |
| Table 26 Data Dictionary Tutor.....                         | 58 |
| Table 27 Data Dictionary Chatroom.....                      | 58 |
| Table 28 Data Dictionary Notification.....                  | 59 |
| Table 29 Use Case Testing Register .....                    | 83 |
| Table 30 Use Case Testing Login.....                        | 84 |
| Table 31 Use Case Testing Logout.....                       | 84 |
| Table 32 Use Case Testing Add Course.....                   | 85 |
| Table 33 Use Case Testing Add Lecture.....                  | 85 |
| Table 34 Use Case Testing View Lecture .....                | 86 |
| Table 35 Use Case Testing Add Video .....                   | 87 |
| Table 36 Use Case Testing View Video .....                  | 87 |
| Table 37 Use Case Testing Chatroom.....                     | 88 |
| Table 38 Use Case Testing Add Notification .....            | 89 |
| Table 39 Use Case Testing View Notification .....           | 89 |
| Table 40 Use Case Testing View Tutors.....                  | 90 |
| Table 41 Equivalence Partitioning Registration .....        | 91 |
| Table 42 Equivalence Partitioning Login.....                | 91 |
| Table 43 Boundary Value Analysis Email .....                | 92 |
| Table 44 Boundary Value Analysis Password.....              | 92 |
| Table 45 Test Cases for Unit Testing.....                   | 96 |
| Table 46 Test Cases for Black Box Testing of Document ..... | 97 |

|                                                               |    |
|---------------------------------------------------------------|----|
| Table 47 Test Cases for Black Box Testing of Hyperlinks ..... | 97 |
| Table 48 Test Cases for Black Box Testing of Graphics .....   | 97 |
| Table 49 Test Cases for Black Box Testing of Forms .....      | 98 |
| Table 50 Usability Testing .....                              | 98 |
| Table 51 Compatibility Testing .....                          | 99 |

# Chapter 1

## Introduction

## **Chapter 1: Introduction**

Many students cannot manage their study material data. They face many problems in communications with each other. It is difficult for them to find relative data about any course. Students don't care about the data they receive from their class representative or from any other source. We provide solution for each and every problem mentioned above. We provide them with a platform where they can chat with the subject relevant students. We provide them with extra study material which helps them in their educational career. By the help of this source they become efficient and more responsible.

### **1.1. Background**

Education has invariably played a very important role in human life and it's been conducted in numerous ways in which counting on the culture and placement. Throughout the past century's men have traveled from country to country for business, migration, war and education. Want to avoid ancient education was invariably tremendous and invention of web brought this idea nearer to reality.

Online Study may be a reasonably electronic exchange of study connected documents or lectures that permits students to straight forwardly receive study material and learn employing an internet program. Generally, students get irritated by attending to the various webpages associated with their topic and even if they notice their material it generally not complete or it's paid or it's a money-based package.

Online study moves the physical study and receiving lectures to a whole new standard. Online study may be a style of open learning. It covers big selection of technology and learning methodology, as well as each "Online Courses" and "surfing the internet" learning.

### **1.2. Motivations and Challenges**

Computers and internet have also motivated the education system entirely. Online study now plays an important role in the education system. The use of internet is common so it motivates the students. The things that motivates us the most is that everything is available on internet e.g. books, audio video lectures, video tutorials, different courses and many more things. Nowadays the concept of online study portals in an institution is spreading rapidly and we are also providing a portal with extra and most beneficial features.

The challenges we face now a days in online study are these that during the searching of our study related materials we have to survey many web pages that consumes our so much time. Even though we find our relevant data it's not complete. Sometimes a student wants to find online courses, there are many free courses available but they aren't that good. If he is able to find good ones they are either paid or trial based.

So, these are some of the challenges we face in our education and also these are the thing that motivated us in creating this project.

### **1.3. Goals and Objectives**

**Goal:**

Our main goal is to provide each and every student with the help they need in their study including the lectures, video tutorials, PPT slides, complete courses etc.

**Time and energy efficiency:**

It is time saving and energy efficient. Students without any physical effort can login to their accounts and can have access to any type of study material they need. They don't have to visit someone else for the lectures.

**Accessibility:**

Students can have access to their lectures, tutorials and many other things while sitting anywhere.

**Scope:**

The scope of our project is that students can manage their study material easily and they can have chatrooms available for each course to communicate with other students. They can also find most accurate and relevant data. A Particular Student can upload their own lectures and will also be able to help other students and the benefit they get is that they can be hired by anyone as a tutor.

This project has well accurately and perfectly designed database so that no data can be misplaced or lost. It has well designed front end that will attract the viewers and looks cool. It is a friendly user interface.



## **1.4. Literature Review/Existing Solutions**

We have many platforms available online that provide us with the helping materials for our study. Many of which provides us only with the information about educational career like when will the admission opens, Past Papers, Location of Institutions and Career Counseling for example: [ilmkiduniya.com](http://ilmkiduniya.com).

Many platforms only provide material related to the subject only on trial or Paid. For Example; [udemy.com](http://udemy.com).

Many Platforms only provide us only with the information about our University like where are our classes or when will our next lecture takes place for example; University Management System.

But we are taking multiple points from these platforms and combining them into one so that we can provide student of all the country with a source that will helps them in their career counseling, providing free online lectures, Helping Material (PPT Slides), Information, Stores University's own classrooms data Like: (Google Classroom).

Our Competitors will be the low-level colleges who provides diplomas of different fields, online paid Platforms who provide materials about courses only for trial or by paying.

## **1.5. Proposed Solution**

First of all, we are providing with a platform that will help every student in their educational career finding relevant materials of their subjects. Many students visit many websites to find their material but they only get trial periods and lectures just containing the information. We are providing them with totally free and complete lectures containing videos, PPT Slides, PDF, and Word Documents etc. We are also providing them with a chatroom relevant to their subject that means a different chatroom for different courses so it will help them communicate with the other relevant students.

## **1.6. Project Plan**

We are creating a plan that will helps us in completing our project and it includes many different steps that will tell us about the accuracy of the project. And we will be able to know the time of the project completion and how it will be completed.

### 1.6.1. Work Breakdown Structure

- **Project Management**
  - Work Breakdown Structure (WBS)
  - Roles & Responsibility Matrix
  - Develop Course Plan
  - Develop Schedule
- **Reports / Documentation**
  - Final Documentation Introduction
  - SRS
  - Requirements Analysis
  - System Design
  - Implementation
  - Testing & Performance Evaluation
- **Replace Existing Website**
  - Compile Research on different Website Function
  - Coordinate Brainstorming Meeting
  - Develop List of Needs based on Research
- **System**
  - **Development Environment**
    - IDE (ASP.Net/C#)
    - Version Control
    - Server
    - Database
  - **Develop Website Content**
    - Compile Website Copy
    - Compile Photos and Images
    - Compile Communication
    - Compile Server Performance
  - **Coordinate Training**
    - Training for Website Maintenance

- Develop a 'How-to' Manual
- **Promote Website**
  - Place an Ad
  - Post on Facebook, Twitter

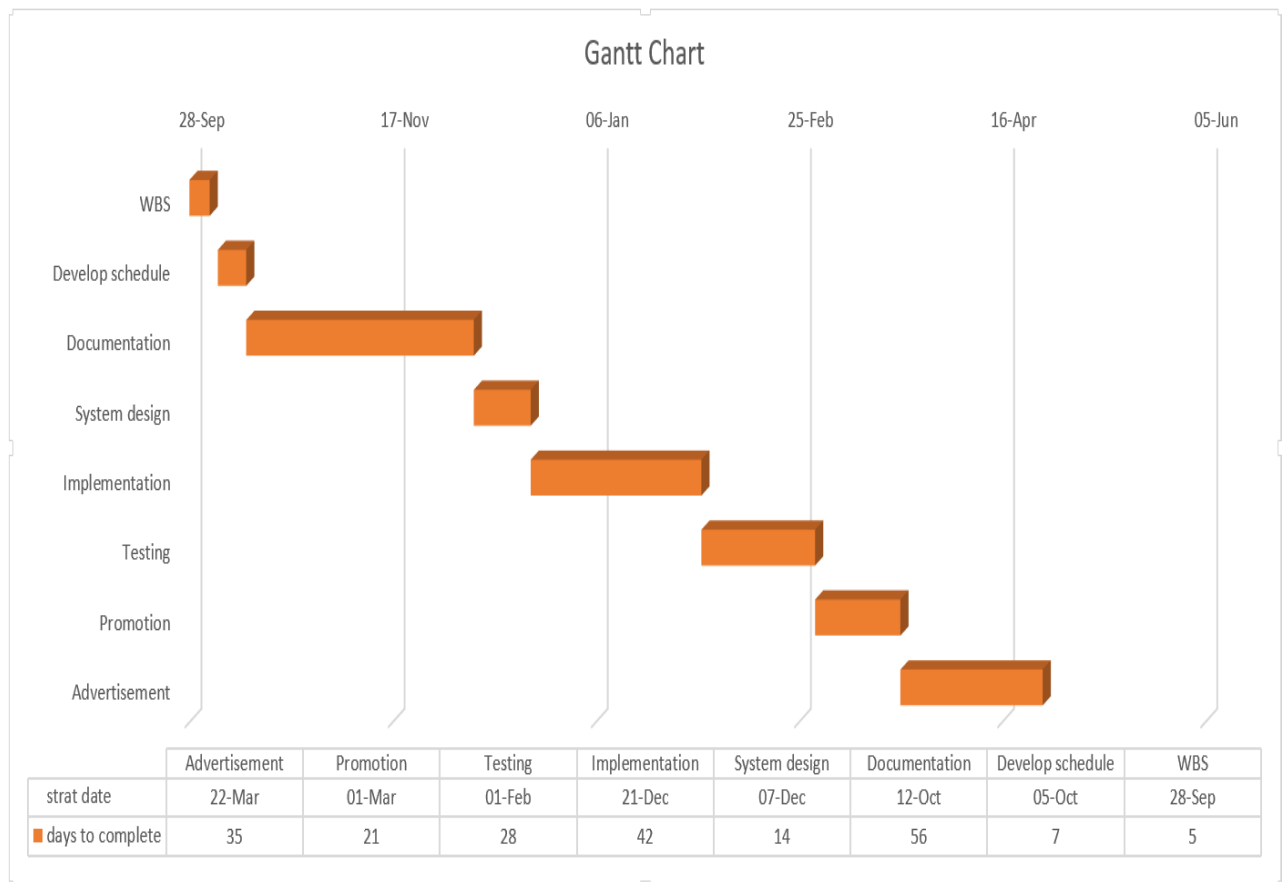
Research blog forms free listing etc.

## 1.7. Roles & Responsibility Matrix

The purpose of roles & responsibility matrix is to identify who will do what.

| WBS # | WBS Deliverable  | Activity # | Activity to Complete the Deliverable | Duration (# of Days) | Responsible Team Member(s) & Role(s) |
|-------|------------------|------------|--------------------------------------|----------------------|--------------------------------------|
| 1.1   | WBS              | 1          | Meeting                              | 1 Week               | Hafiz Usman Ilyas                    |
| 1.4   | Develop Schedule | 1          | Meeting                              | 1 Week               | Hafiz Usman, Nida Gulzar             |
| 2.1   | Documentation    | 2          | Session                              | 8 Weeks              | Nida gulzar,Hafiz Usman              |
| 2.4   | System Design    | 2          | Meeting                              | 2 Weeks              | Hafiz Usman                          |
| 2.5   | Implementation   | 2          | Meeting                              | 6 Weeks              | Hafiz Usman                          |
| 2.6   | Testing          | 2          | Meeting                              | 4 Weeks              | Nida Gulzar                          |
| 4.4   | Promotion        | 4          | Marketing                            | 3 Weeks              | Nida Gulzar                          |
| 4.4.2 | Advertisement    | 4          | Marketing                            | 5 Weeks              | Hafiz Usman                          |

## 1.8. Gantt Chart



*Figure 1 Gantt chart*

# Chapter 2

## **Software Requirement Specifications**

## Chapter 2: Software Requirement Specifications

### 2.1. Introduction

Many students cannot manage their study material data. They face many problems in communications with each other. It is difficult for them to find relative data about any course. Students don't care about the data they receive from their class representative or from any other source. We provide solution for each and every problem mentioned above. We provide them with a platform where they can chat with the subject relevant students. We provide them with extra study material which helps them in their educational career. By the help of this source they become efficient and more responsible.

#### 2.1.1. Purpose

The main purpose of this project is to provide each and every student with the help they need in their studies including lectures, Courses, Video Tutorials and many other resources. Particular student can also add/search/view their own lectures.

#### 2.1.2. Document Conventions

Standard rule of documentation has been followed in order to standardize the work in order to create efficient **Student World** to help students in their studies.

#### 2.1.3. Intended Audience and Reading Suggestions

This report is for individuals like Software Engineers, Client, Developer, Database Designer.

#### 2.1.4. Product Scope

Scope of our project is that a student comes to the portal and search their desired document/lecture view/read it and gain knowledge from it.

#### 2.1.5. References

- IEEE 830-1998 standard for writing SRS document.
- I Sommerville, Software Engineering, 8<sup>th</sup> ed, Addison- Wesley, 2007.

## 2.2. Overall Description

### 2.2.1. Product Perspective

This project intends to remove the conflicts between students and education. Students can achieve their material related to study from this portal. It intends to facilitate the students with everything they need.

### 2.2.2. Product Functions

#### Website:

**Register:** Teachers and Students can register themselves on the portal.

**Login:** Registered Members can login their accounts.

**Logout:** After Login and spending the needy time on the portal they can logout.

**Add Subject:** After Login members can add/enroll subjects.

**View Enroll Subject:** Other registered members can view the enrolled subjects.

**Add Subject Detail:** The details of the related subject will be added after enrollment.

**View Lectures:** Registered members can view lectures and all the other material related to the subject or even if they need help in any kind of study.

**Add Lectures:** Teachers, students or any members can add lectures to the portal so it helps other students in the study.

**Add/View Videos:** All registered members can add new videos to the portal related to the study and all other members can view these videos.

**Send Notifications:** After uploading of lectures, videos and any other activity in the portal the notification will be sent to the students so they can stay up-to-date.

**Chatroom:** Chatroom is available for all the members so they can chat with each other about the related subject or lecture.

**Add Tutor:** Members can register themselves or add themselves as a tutor so they can be hired by someone.

**Search/View Tutor:** Members can particularly search tutors for themselves or someone else.

## Decide what section is computerized, and how

- Every Process will be computerized
- With the help of this, Admin can print a report any time less time consuming
- A central DB is required and will be installed in sever that sever requires online implementation.

## Determine detail of dataflow

### Login detail

- Email (12 char)
- Password (19 int)

### Register member detail

- Name (25 char)
- Email (25 char)
- Password (25 char)
- Address (20 letters)
- Contact (14 char)

### Subject Detail

- Subject\_id(6digit)
- Subject\_name(25char)
- Subject\_detail(20 letters)
- User\_id(6digits)

### Video Detail

- Video\_id(6digit)
- Subject\_id(6digit)
- User\_id(6digit)
- Video\_lenght(varchar 50)
- Video\_link (40 char)

### Delete Detail

- Tutor\_id(6digits)
- Subject\_id (6digits)
- Lecture\_id(6digits)
- Video\_id(6digits)

### Tutor Detail

- Tutor (6digits)
- Tutor\_name (25 char)



- Subject\_name(20 char)
- Subject\_id(6digits)
- Tutor\_detail(20 letters)

**Lecture Detail**

- Lectures\_id (6digit)
- lecture\_name (25 char)
- Lecture\_detail (40 char)
- Lecture\_size (40char)
- Lecture\_size(40char)

**Logout Detail**

- Logout detail (20 char)

**Define the logic of processes**

- Login table Detail
- Register table Detail
- Subject Detail
- Video Detail
- lecture Detail
- Tutor Detail
- Delete Detail
- Logout member table detail

**Define the data store**

- Login table contain (3steps)
- Page layout contain (3steps)
- Subject table contain (3steps)
- Tutor table contain (3steps)
- Video table contain (3steps)
- lecture table contain (3steps)
- Delete table contain (3 steps)
- Register table contain (3 steps)
- Logout member table contain (3steps)

**Define Physical resources**

- Sequential File
- MY-SQL

## Database required with separate table

### Login Page Table

- Login page with primary key (pk)

### Page Layout Table

- Page layout with primary key (pk)

### Subject Table

- Subject Detail with primary key (pk)
- User detail (fk)

### Tutor Table

- Tutor detail with primary key (pk)
- Subject detail (fk)

### Videos Table

- Videos Detail with primary key (pk)
- Subject detail (fk)

### Lecture Table

- Lecture detail with primary key (pk)
- Subject detail (fk)

### Delete Detail Table

- Delete Detail with primary key (pk)

### Logout table

- Logout table no with primary key (pk)

## Perform sizing

- 4-megabyte hardware required
- Login table contain (25 char)
- Subject table contain (13 kilobyte)
- Delete Subject table contain (5 kilobyte)
- Tutor detail table contain (30 byte)
- Logout member table contain (0byte)

## Determine the hardware Requirement

- One disk drive to store the backup
- Printer to print a report
- Pc with desktop to view the data

- Server or LAN cables to connect systems

### 2.2.3. User Classes and Characteristics

There are 3 kinds of client of this product

#### **Overall population**

Overall population can utilize the framework to see the Subject, their Lectures and amount accessible. General client can't store the Lecture.

#### **Clients**

Clients are utilizing for review and get the lectures.

#### **Administrator**

Administrator can include, alter and erase Users, Subject, Lectures and Videos and give administrations to the client. Chairman can see the everyday routine.

### 2.2.4. Operating Environment

Operating Environment Win 8/10

Platform .Net Framework & IIS / C#

Database SQL Server

#### **For Users:**

Internet Browser

Internet Connection

The Student World is a site which will work in every single well-known program for a model we are taking Microsoft Internet Explorer, Google Chrome, Mozilla Firefox with Flash Player and JavaScript

### 2.2.5. Design and Implementation Constraints

- Client detail save into the database.
- MS SQL Server will be utilized as SQL motor and database.
- The Online Student World System is running 24 hours per day.
- Users may access from any PC that has Internet perusing abilities and an Internet association.
- Users must have their right usernames and passwords to go into their online records and do activities

### 2.2.6. User Documentation

User must know detail of our Website, how to use it or how to deploy it. User use internet to visit our website or for needing friendly environment.

## 2.2.7. Assumptions and Dependencies

- Microsoft SQL Server to store the database.
- ASP.net/C# to build up the Product

## 2.3. External Interface Requirements

### 2.3.1. User Interfaces

Client of the site will be given the Graphical UI; there is no order line interface for any elements of the item.

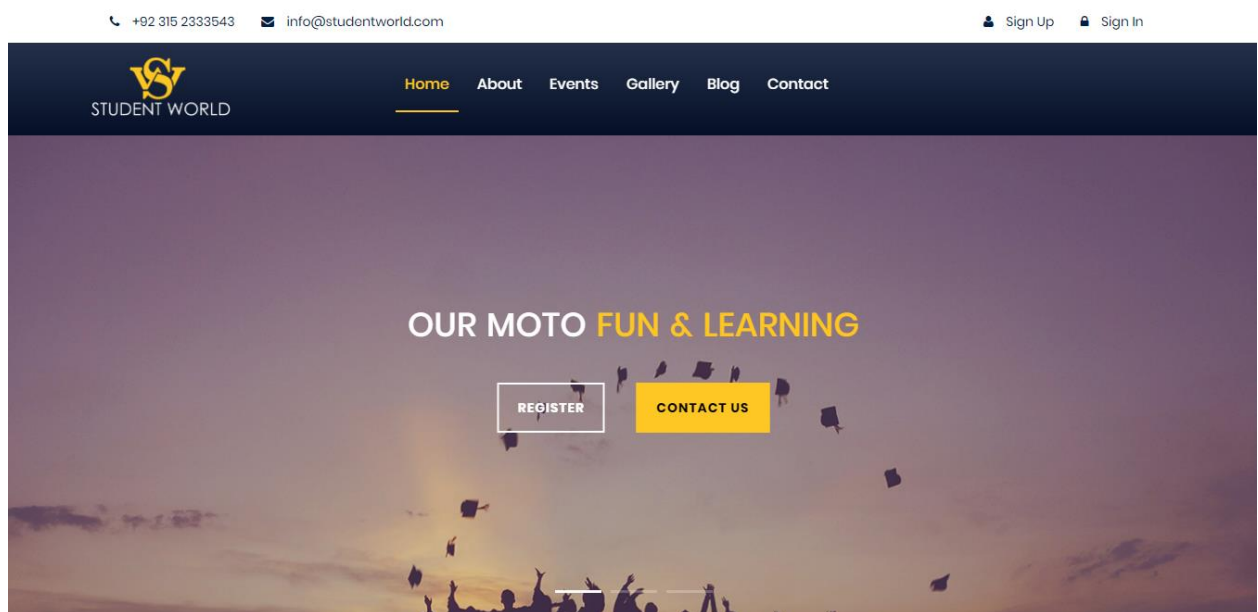


Figure 2 GUI 1

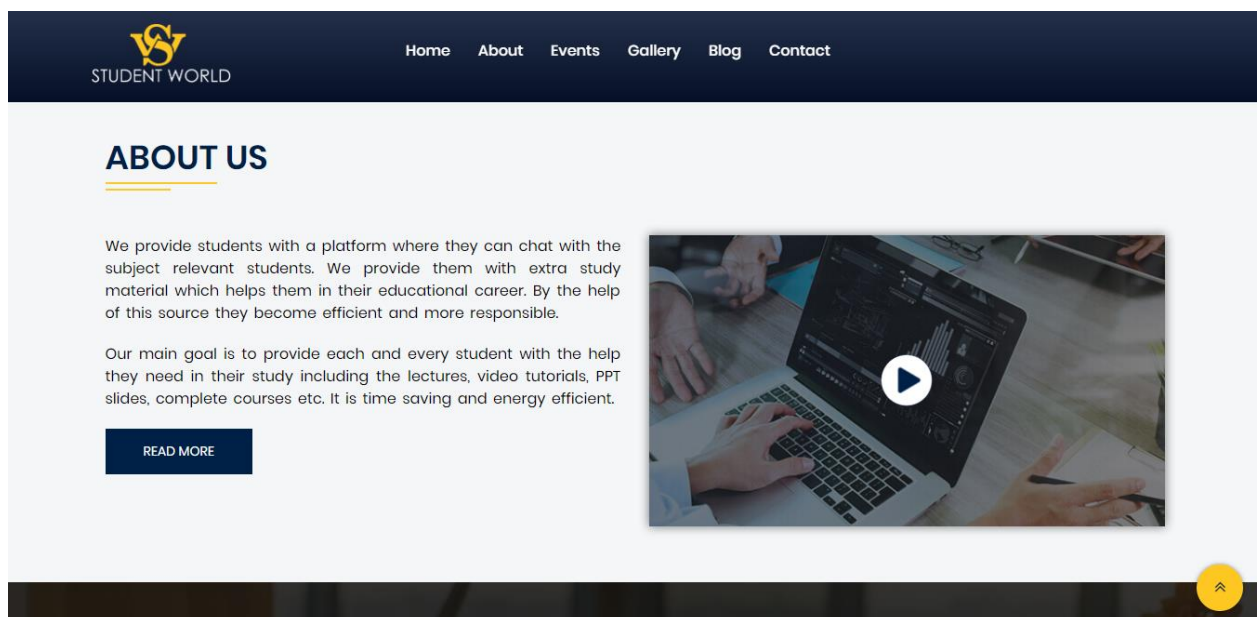


Figure 3 GUI 2

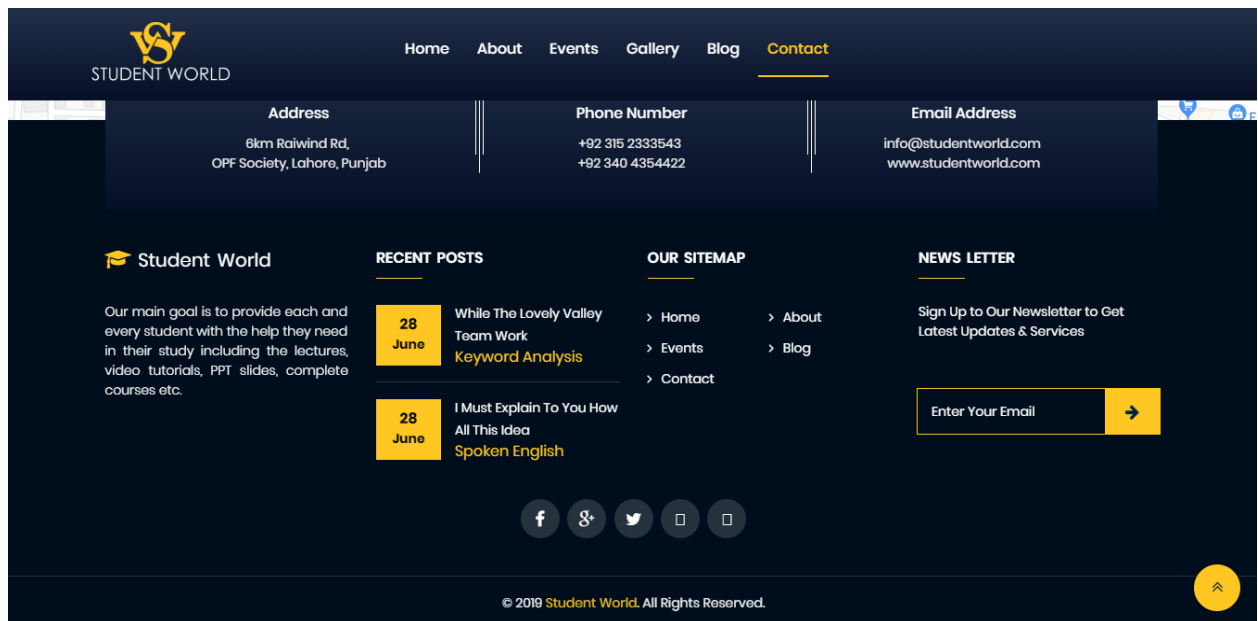


Figure 4 GUI 3

| Screen Name                 | Description                                                                                                                                                                                                                                          |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Login                       | In that case the client isn't enlisted yet; he can enter the points of interest and enlist. Which requests that the client write his username and secret key. If the client entered his username or Password Incorrect then error message occurs.    |
| Search                      | The registered user of <b>Student World</b> can enter the kind of subject he is looking for and after that he can scan for the required lecture by entering the name of the subject. User can also search/view subject and tutor by their specialty. |
| Subject view                | The subject classifications see with capacity. <b>Student World</b> include/alter or erase classification from the rundown.                                                                                                                          |
| Add subject to account      | User can add subject into the account which he wants to study                                                                                                                                                                                        |
| Delete subject from account | After adding the subject into the account user can also delete the selected subject from the account.                                                                                                                                                |
| lecture Method              | User selects the lecture method and proceeds to place the order.                                                                                                                                                                                     |
| Administer seats            | Availability                                                                                                                                                                                                                                         |
| Administer user             | Create moodily and delete user, change password                                                                                                                                                                                                      |
| Report                      | Select, view, add, save and delete reports                                                                                                                                                                                                           |

Table 1 GUI Screens

### 2.3.2. Hardware Interfaces

- Equipment prerequisites for Insurance on web will be same for both the gatherings which are takes after:
- Processor
- Ram: - 128 MB or above.
- HD: - 20 GB or above.
- Just the prescribed setup (fundamental necessities of a PC framework) no other particular equipment is required to run the product.

### 2.3.3. Software Interfaces

All the databases for the Student World system will be configured using access/MySQL 2018 – 2027. These databases include Subjects, lectures, Videos, Tutor and user information. These can be modified by the end user. The Student World database will include the user, Subjects, lectures, Videos and Tutor info. These detailed information databases will contain all the detail such as first name surname ID number, Gender and phone number.

### 2.3.4. Communications Interfaces

The two gatherings ought to be associated through either by LAN or WAN for the correspondence. Sender Communication channels Receiver.



*Figure 5 Communication Interfaces*

## 2.4. System Features

#### Website:

**Register:** Teachers and Students can register themselves on the portal.

**Login:** Registered Members can login their accounts.

**Logout:** After Login and spending the needy time on the portal they can logout.

**Add Subject:** After Login members can add/enroll subjects.

**View Enroll Subject:** Other registered members can view the enrolled subjects.

**Add Subject Detail:** The details of the related subject will be added after enrollment.

**View Lectures:** Registered members can view lectures and all the other material related to the subject or even if they need help in any kind of study.

**Add Lectures:** Teachers, students or any members can add lectures to the portal so it helps other students in the study.

**Add/View Videos:** All registered members can add new videos to the portal related to the study and all other members can view these videos.

**Send Notifications:** After uploading of lectures, videos and any other activity in the portal the notification will be sent to the students so they can stay up-to-date.

**Chatroom:** Chatroom is available for all the members so they can chat with each other about the related subject or lecture.

**Add Tutor:** Members can register themselves or add themselves as a tutor so they can be hired by someone.

**Search/View Tutor:** Members can particularly search tutors for themselves or someone else.

### **Technical Features:**

#### **Website**

Software Requirements:

- Windows 10
- SQL 2012
- Visual studio 2013

Equipment Components:

- Processor – i3
- Hard Disk – 5 GB
- Memory – 1GB RAM
- Graphical Tools: Photoshop CC 2015, Adobe artist CC

### **Technical Features:**

#### **Application**

Software Requirements:

- Android Mobile

### 2.4.1. System Feature 1

#### Function Requirement 1 Register

“User can register into the System”

|                      |                                                                                  |
|----------------------|----------------------------------------------------------------------------------|
| Identifiers          | FR-001                                                                           |
| Title                | Allows Register access online                                                    |
| Requirement          | User name, Password, City, Email                                                 |
| Rational             | Register to the system and provide security to personal information.             |
| Restriction and risk | Correct data must be entered during registration, otherwise user don't register. |
| Dependency           | Pc, Connection with server                                                       |
| Priority             | Safety, timing                                                                   |

*Table 2 Function Requirement Register*

### 2.4.2. System Feature 2

#### Function Requirement 2 Login Function

“User can login into the System”

|                      |                                                                       |
|----------------------|-----------------------------------------------------------------------|
| Identifiers          | FR-002                                                                |
| Title                | Allows login access online                                            |
| Requirement          | User name, Password                                                   |
| Rational             | Login to the system and provide security to personal information.     |
| Restriction and risk | User name and password must be correct otherwise user don't register. |
| Dependency           | Pc, Connection with server                                            |
| Priority             | Safety, timing                                                        |

*Table 3 Function Requirement Login*

### 2.4.3. System Feature 3

#### Function Requirement 3 Logout Function



“After visiting the website user can logout”

|                      |                                                                    |
|----------------------|--------------------------------------------------------------------|
| Identifiers          | FR-003                                                             |
| Title                | Allows Logout access online                                        |
| Requirement          | None                                                               |
| Rational             | Logout to the system and provide security to personal information. |
| Restriction and risk | User must login.                                                   |
| Dependency           | Pc, Connection with server                                         |
| Priority             | Safety, timing                                                     |

*Table 4 Function Requirement Logout*

#### 2.4.4. System Feature 4

### Function Requirement 4

#### Select subject Function

“User can select subject and add to account”

|                      |                                          |
|----------------------|------------------------------------------|
| Identifiers          | FR-004                                   |
| Title                | Add a subject to user account            |
| Requirement          | Subject name, time, Subject Detail, date |
| Rational             | A user must add subject into account     |
| Restriction and risk | User must add subject                    |
| Dependency           | Valid information                        |
| Priority             | subject add in user account              |

*Table 5 Function Requirement Select Subject*

#### 2.4.5. System Feature 5

### Function Requirement 5

#### View Detail Function

“User can view subject user can also view the detail of the Subjects and Tutor.”

|                      |                                                                  |
|----------------------|------------------------------------------------------------------|
| Identifiers          | FR-005                                                           |
| Title                | Allows user to view detail                                       |
| Requirement          | User login and then view detail of subject                       |
| Rational             | Login to the system and view detail                              |
| Restriction and risk | User must login to check the detail otherwise detail don't show. |
| Dependency           | Pc, Connection with server                                       |

|          |                |
|----------|----------------|
| Priority | Safety, timing |
|----------|----------------|

*Table 6 Function Requirement View Detail***2.4.6. System Feature 6****Function Requirement 6****Add lecture Function**

“User has ability to Add Lecture to the available Subject.”

|                      |                                                                            |
|----------------------|----------------------------------------------------------------------------|
| Identifiers          | FR-006                                                                     |
| Title                | Access to add lecture on schedule                                          |
| Requirement          | Lecture name, subject type, date                                           |
| Rational             | User has ability to Add Lecture to the available Subject.                  |
| Restriction and risk | Current subject, the available lecture or current timing must be adjusted. |
| Dependency           | Valid information, available lecture is valid                              |
| Priority             | Modified in order                                                          |

*Table 7 Function Requirement Add Lecture***2.4.7. System Feature 7****Function Requirement 7****View lecture Function**

User can View the Lecture of the available subject.

|                      |                                                        |
|----------------------|--------------------------------------------------------|
| Identifiers          | FR-007                                                 |
| Title                | Access to view lectures on schedule                    |
| Requirement          | subject name, lecture type, date                       |
| Rational             | User has ability to View lecture of available subject. |
| Restriction and risk | User must login to view lecture detail.                |
| Dependency           | Valid information                                      |
| Priority             | View subject detail                                    |

*Table 8 Function Requirement View Lecture***2.4.8. System Feature 8****Function Requirement 8**

### Delete lecture Function

User can Delete the Lecture which he has entered.

|                      |                                                                             |
|----------------------|-----------------------------------------------------------------------------|
| Identifiers          | FR-008                                                                      |
| Title                | Access to delete lecture on schedule                                        |
| Requirement          | Lecture Name, Subject Detail, date                                          |
| Rational             | User has ability to the delete lecture.                                     |
| Restriction and risk | Current subject, the available lectures or current timing must be adjusted. |
| Dependency           | Valid information, available lecture is valid.                              |
| Priority             | Modified in order.                                                          |

*Table 9 Function Requirement Delete Lecture*

#### 2.4.9. System Feature 9

### Function Requirement 9

#### Add video Function

After login user can add Videos of the specific lectures.

|                      |                                              |
|----------------------|----------------------------------------------|
| Identifiers          | FR-009                                       |
| Title                | Allow user to add video to lecture.          |
| Requirement          | Type of video, video no, video name          |
| Rational             | Allow user to add video into subject easily. |
| Restriction and risk | None                                         |
| Dependency           | None                                         |
| Priority             | None                                         |

*Table 10 Function Requirement Add Video*

## 2.5. Other Nonfunctional Requirements

### 2.5.1. Performance Requirements

With a specific end goal to keep up a worthy speed at most extreme number of transfers permitted from a specific client will be any number of clients and can get to the framework whenever. Likewise, associations with the servers will be founded on the criteria of properties of the client like his area, and server will work 24X 7 times.

### 2.5.2. Safety Requirements

All exchanges, logged data, refreshes, client exercises are reinforcement.

### 2.5.3. Security Requirements

Any adjustment (insert, update and delete) for the Database might be synchronized and done just by the System manager.

### 2.5.4. Software Quality Attributes

#### **Reliability**

The framework can be utilized by different clients simultaneously. Any client can get to the framework with utilizing even on a low execution PC.

#### **Availability**

The framework is accessible 24 hours of the day.

#### **Maintainability**

The framework should give the ability to go down the Data.

#### **Portability**

Client can sign in to the framework whenever.

#### **Robustness**

The framework won't be separated effortlessly.

### 2.5.5. Business Rules

There are mostly two kinds of clients utilizing the framework, for example,

- User
- Administrator

**Administrator** has the full authorization of controlling the framework.

**User** has authorization to manage study material and tutors.

## 2.6. Other Requirements

At the point undertaking the principal matter to build up is to survey the plausibility of a task or item. It is essentially utilized as a measure how down to earth or gainful the advancement of a product framework will be to you (or association). This movement repeats for the duration of the life cycle.

There are numerous kinds of possibilities:

- Technical
- Economic
- Specification
- Information
- Motivational
- Legal and Ethical

## Technical Feasibility

**Student World** site require following advances:

**Server:** A server will be required which will have all information of customer. It checks client name and passwords of the considerable number of clients, the areas went by the clients, an occasion created by client e.g. data about some strike in the way.

## Economic Feasibility

**Student World** site require these fundamental to meet to work:

### Cost of System:

- 1 Android Phone (GPS bolster) @ Rs.20, 000
- Internet Connection @Rs.1000/month

### Advantages

- 24 hours accessible

## Schedule Feasibility

We isolate the undertaking into various modules and prerequisite modules will be constructed and afterward would be coordinated to make an appropriate working application.

## Specification Feasibility

This web base Application is for Windows Users. Web Application would be free of cost. Required Specifications are Windows (8, 7), Visual Studio, Rational Rose and MS Office. Some Specifications are as take after

- Mobile telephone that help GPS
- Pentium 4 Processors
- 80 GB hard plates
- 1 GB RAM

Here enrolled are some Non-Functional Requirements:

- GUI
- User amicable
- Performance

### **Motivational Feasibility**

The thought is given by an instructor and we are engaged to defeat the thought. As this web base application appear to be extremely useful for individuals.

### **Legal & Ethical Feasibility**

This web base application would be accessible for all clients and will meet legitimate and moral principles. As the client who will produce occasions would be a valid client with client data accessible so it is imagined that it would be a decent affair to utilize this application.

# Chapter 3

## Use Case Analysis

## Chapter 3: System Analysis

This framework is tied in changing over the educational framework from manual to online system. Students can register themselves on this platform and add their relevant subjects and then add lectures related to those subjects and video tutorials and also students can register themselves as tutors so that other students or visitors on website can hire them. Administrator or User both can send notifications to the specific students about lectures and videos. Administrator prepares and manages the day by day report about the platform.

### 3.1. Use Case Model

A utilization case is an accumulation of related achievement and disappointment situations that depict on-screen characters utilizing the framework to help an objective. A situation is a particular succession of activities and associations amongst performing artists and the framework under discourse.



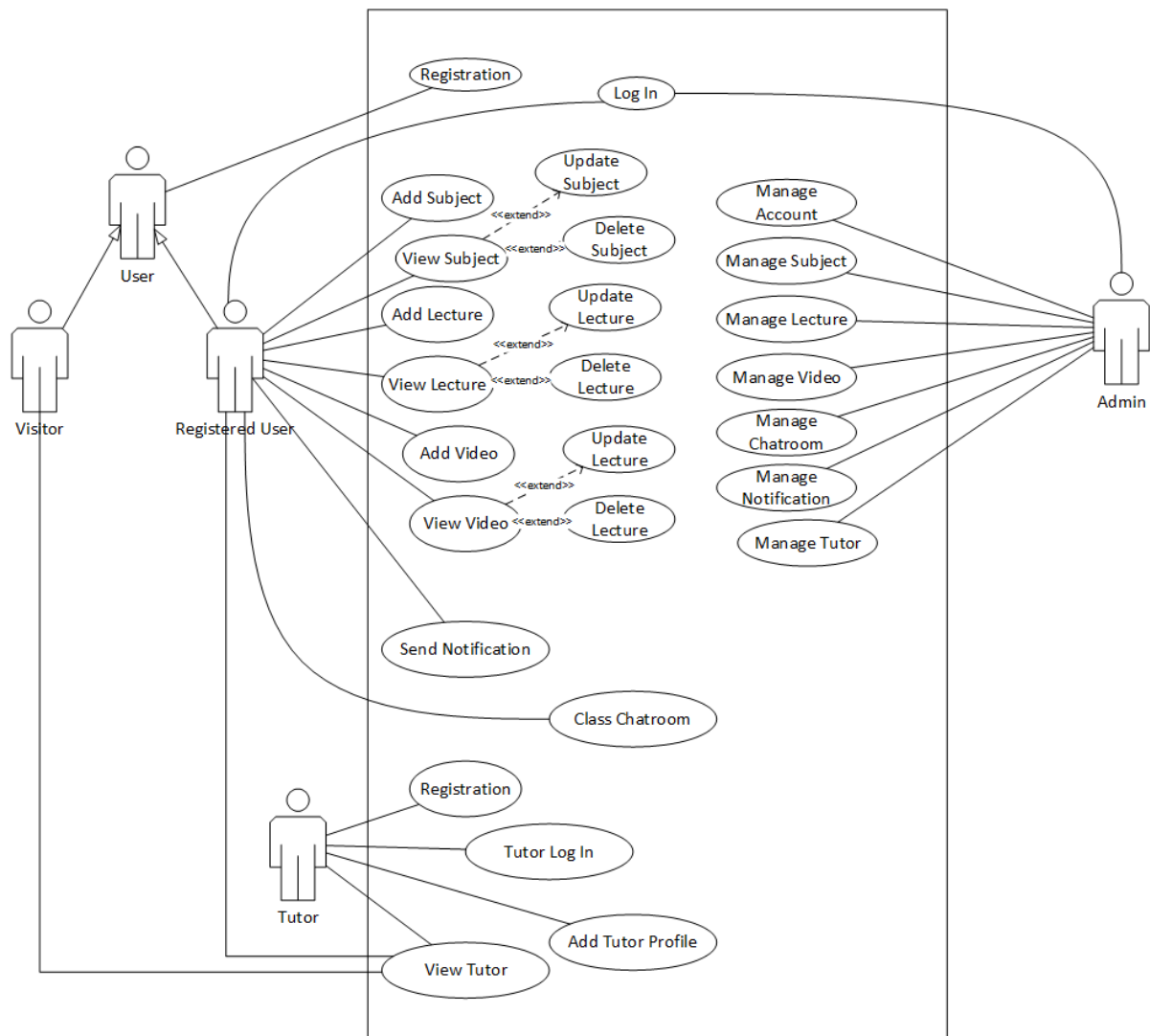


Figure 6 Use Case Diagram

### 3.2. Fully Dressed Use Cases

#### Register:

| Use Case Section | Comment                            |
|------------------|------------------------------------|
| Use Case id      | UC-01                              |
| Use Case Name    | Register                           |
| Scope            | New User should be able to sign up |
| Level            | User Register himself              |
| Primary Actor    | Unregistered User                  |
| Stake Holder     | Admin                              |

|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Precondition          | None                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Post Condition        | Successfully Signed Up                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Main Success Scenario | <ul style="list-style-type: none"> <li>• User requested admin for registration</li> <li>• Admin click on new user registration</li> <li>• Admin filled the form of user registration</li> <li>• System register the user</li> <li>• System generate user name and default password for that user.</li> <li>• System email user name and default password to user email</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                      |
| Extensions            | <ul style="list-style-type: none"> <li>• Admin didn't receive registration request from user <ul style="list-style-type: none"> <li>○ User restart the application and he/she again fill the form for requesting registration</li> </ul> </li> <li>• Admin didn't get a form for registration due to any reason <ul style="list-style-type: none"> <li>○ Admin delete the cache</li> <li>○ If problem still remain than admin restart the application</li> </ul> </li> <li>• Admin entered the wrong user data in registration form</li> <li>• Admin immediately delete that user</li> <li>• Admin click on add user</li> <li>• Admin again fill the form for new user and then he/she created the user</li> <li>• System already have registered that user</li> </ul> |

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     | <ul style="list-style-type: none"> <li>• System show alert to the admin that the user already exists</li> <li>• Admin cancel the dialog and also close the registration for adding that user</li> <li>• Admin cancel the registration process at any step</li> <li>• Admin press the cancel registration button</li> <li>• System restore its previous state</li> <li>• User didn't get the user name and default password</li> <li>• User send the complaint mail to the system</li> <li>• Admin check the system that the right information is added or not</li> <li>• Admin again request system to generate username and password for that user</li> <li>• System send the username and password through provided mail</li> </ul> |
| Special Requirement | Security                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

*Table 11 Fully Dressed Use Case Register*

### Log In:

| Use Case Section | Comment                                 |
|------------------|-----------------------------------------|
| Use Case id      | UC-02                                   |
| Use Case Name    | Log In                                  |
| Scope            | Registered User must be able to sign in |
| Level            | User Log In himself                     |
| Primary Actor    | Registered User                         |
| Stake Holder     | Admin                                   |

|                       |                                                                                                                                                                                                                         |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Precondition          | Must be Registered                                                                                                                                                                                                      |
| Post Condition        | Successfully Logged In                                                                                                                                                                                                  |
| Main Success Scenario | <ul style="list-style-type: none"> <li>• This function starts when the user wants to login.</li> <li>• The user presses the login button.</li> <li>• The system provides the all details and login the user.</li> </ul> |
| Extensions            | If the customer enters the wrong password then the statement will appear “the wrong password it is “.                                                                                                                   |
| OSpecial Requirement  | Security, Efficiency                                                                                                                                                                                                    |

*Table 12 Fully Dressed Use Case Login*

#### **Logout:**

| Use Case Section      | Comment                                                                                                                                                                                      |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case id           | UC-03                                                                                                                                                                                        |
| Use Case Name         | Logout                                                                                                                                                                                       |
| Scope                 | Registered User must be able to logout                                                                                                                                                       |
| Level                 | User Logout himself                                                                                                                                                                          |
| Primary Actor         | Registered User                                                                                                                                                                              |
| Stake Holder          | Admin                                                                                                                                                                                        |
| Precondition          | Must be logged in                                                                                                                                                                            |
| Post Condition        | Successfully Logged out                                                                                                                                                                      |
| Main Success Scenario | <ul style="list-style-type: none"> <li>• This function starts when the user wants to logout</li> <li>• The user presses the logout button.</li> <li>• The system logout the user.</li> </ul> |
| Extensions            | None                                                                                                                                                                                         |
| Special Requirement   | Security, Efficiency                                                                                                                                                                         |

*Table 13 Fully Dressed Use Case Logout*

**Add Subject:**

| Use Case Section      | Comment                                                                                                                                                                                                                                                                                     |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case id           | UC-04                                                                                                                                                                                                                                                                                       |
| Use Case Name         | Add Subject                                                                                                                                                                                                                                                                                 |
| Scope                 | Registered User able to add subjects                                                                                                                                                                                                                                                        |
| Level                 | User add subject                                                                                                                                                                                                                                                                            |
| Primary Actor         | Registered User                                                                                                                                                                                                                                                                             |
| Stake Holder          | Registered User                                                                                                                                                                                                                                                                             |
| Precondition          | Must be logged in                                                                                                                                                                                                                                                                           |
| Post Condition        | Successfully add subject                                                                                                                                                                                                                                                                    |
| Main Success Scenario | <ul style="list-style-type: none"> <li>• This function starts when user want to add new subject.</li> <li>• User login in to the system and click on add subject</li> <li>• User adds the details of subject and then click save</li> <li>• The system successfully adds subject</li> </ul> |
| Extensions            | <ul style="list-style-type: none"> <li>• If the user adds the subject which already exist then the system show message "Subject already exist."</li> <li>• User then again adds details of the subject with a unique code or a different name and adds the subject.</li> </ul>              |
| Special Requirement   | Efficiency                                                                                                                                                                                                                                                                                  |

*Table 14 Fully Dressed Use Case Add Subject***View Subject**

| Use Case Section | Comment                               |
|------------------|---------------------------------------|
| Use Case id      | UC-05                                 |
| Use Case Name    | View Subject                          |
| Scope            | Registered User able to view subjects |

|                       |                                                                                                                                                                                                                                                                                |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Level                 | User view subject                                                                                                                                                                                                                                                              |
| Primary Actor         | Registered User                                                                                                                                                                                                                                                                |
| Stake Holder          | Registered User                                                                                                                                                                                                                                                                |
| Precondition          | Must be logged in                                                                                                                                                                                                                                                              |
| Post Condition        | Successfully view subject                                                                                                                                                                                                                                                      |
| Main Success Scenario | <ul style="list-style-type: none"> <li>• This function starts when a user wants to view a subject.</li> <li>• User enter the subject name or details and press view button.</li> <li>• User successfully views subject.</li> </ul>                                             |
| Extensions            | <ul style="list-style-type: none"> <li>• If the user enters detail of a subject that doesn't exist then the system shows the message "Subject doesn't exist."</li> <li>• User then make sure that the subject he wants to view exists and re-enter all the details.</li> </ul> |
| Special Requirement   | Efficiency                                                                                                                                                                                                                                                                     |

*Table 15 Fully Dressed Use Case View Subject*

### Add Lecture

| Use Case Section | Comment                                       |
|------------------|-----------------------------------------------|
| Use Case id      | UC-06                                         |
| Use Case Name    | Add Lecture                                   |
| Scope            | Registered User able to add lectures          |
| Level            | User add lectures                             |
| Primary Actor    | Registered User                               |
| Stake Holder     | Registered User                               |
| Precondition     | Must be logged in and have a relevant subject |
| Post Condition   | Successfully add lecture                      |

|                       |                                                                                                                                                                                                                                                                                                                                            |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Main Success Scenario | <ul style="list-style-type: none"> <li>• This function starts when user want to add new lecture.</li> <li>• User login in to the system and opens the relevant subject for which user wants to add new lecture.</li> <li>• User adds the details of lecture and then click save</li> <li>• The system successfully adds lecture</li> </ul> |
| Extensions            | <ul style="list-style-type: none"> <li>• If the user adds the lecture with the same details then the system show message "Lecture with these details already exist."</li> <li>• User then again adds details of the lecture and adds the lecture.</li> </ul>                                                                               |
| Special Requirement   | Efficiency                                                                                                                                                                                                                                                                                                                                 |

*Table 16 Fully Dressed Use Case Add Lecture*

### View Lecture

| Use Case Section      | Comment                                                                                                                                                                       |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case id           | UC-07                                                                                                                                                                         |
| Use Case Name         | View Lecture                                                                                                                                                                  |
| Scope                 | Registered User able to View added lectures                                                                                                                                   |
| Level                 | User view lectures                                                                                                                                                            |
| Primary Actor         | Registered User                                                                                                                                                               |
| Stake Holder          | Registered User                                                                                                                                                               |
| Precondition          | Must be logged in and have an existing lecture                                                                                                                                |
| Post Condition        | Successfully view lecture                                                                                                                                                     |
| Main Success Scenario | <ul style="list-style-type: none"> <li>• This function starts when user want to view lecture of relevant subject.</li> <li>• User login in to the system and opens</li> </ul> |

|                     |                                                                                                                                                                                                                                                                                |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     | <p>the relevant subject for which user wants to view lecture.</p> <ul style="list-style-type: none"> <li>• User adds the details of lecture and then click view</li> <li>• The user successfully view lecture</li> </ul>                                                       |
| Extensions          | <ul style="list-style-type: none"> <li>• If the user enters detail of a lecture that doesn't exist then the system shows the message "lecture doesn't exist."</li> <li>• User then make sure that the lecture he wants to view exists and re-enter all the details.</li> </ul> |
| Special Requirement | Efficiency                                                                                                                                                                                                                                                                     |

*Table 17 Fully Dressed Use Case View Lecture*

### Add Video

| Use Case Section      | Comment                                                                                                                                                                                                                |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case id           | UC-08                                                                                                                                                                                                                  |
| Use Case Name         | Add Video                                                                                                                                                                                                              |
| Scope                 | Registered User able to Add Video                                                                                                                                                                                      |
| Level                 | User add Video                                                                                                                                                                                                         |
| Primary Actor         | Registered User                                                                                                                                                                                                        |
| Stake Holder          | Registered User                                                                                                                                                                                                        |
| Precondition          | Must be logged in and have an existing lecture/subject                                                                                                                                                                 |
| Post Condition        | Successfully add video                                                                                                                                                                                                 |
| Main Success Scenario | <ul style="list-style-type: none"> <li>• This function starts when user want to add new video.</li> <li>• User login in to the system and opens the relevant subject for which user wants to add new video.</li> </ul> |



|                     |                                                                                                                                                                                                                                                      |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     | <ul style="list-style-type: none"> <li>• User adds the details of video and then click save</li> <li>• The system successfully adds video</li> </ul>                                                                                                 |
| Extensions          | <ul style="list-style-type: none"> <li>• If the user adds the video with the same details then the system show message "Video with these details already exist."</li> <li>• User then again adds details of the Video and adds the video.</li> </ul> |
| Special Requirement | Efficiency                                                                                                                                                                                                                                           |

*Table 18 Fully Dressed Use Case Add Video*

### View Video

| Use Case Section      | Comment                                                                                                                                                                                                                                                                                                                     |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case id           | UC-09                                                                                                                                                                                                                                                                                                                       |
| Use Case Name         | View Video                                                                                                                                                                                                                                                                                                                  |
| Scope                 | Registered User able to View Video                                                                                                                                                                                                                                                                                          |
| Level                 | User View Video                                                                                                                                                                                                                                                                                                             |
| Primary Actor         | Registered User                                                                                                                                                                                                                                                                                                             |
| Stake Holder          | Registered User                                                                                                                                                                                                                                                                                                             |
| Precondition          | Must be logged in and have a relevant video to an existing lecture/subject                                                                                                                                                                                                                                                  |
| Post Condition        | Successfully view video                                                                                                                                                                                                                                                                                                     |
| Main Success Scenario | <ul style="list-style-type: none"> <li>• This function starts when user want to view video of relevant topic.</li> <li>• User login in to the system and opens the relevant video of a relevant topic.</li> <li>• User adds the details of video and then click view</li> <li>• The user successfully view video</li> </ul> |
| Extensions            | <ul style="list-style-type: none"> <li>• If the user enters detail of a video</li> </ul>                                                                                                                                                                                                                                    |

|                     |                                                                                                                                                                                                                               |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     | <p>that doesn't exist then the system shows the message "video doesn't exist."</p> <ul style="list-style-type: none"> <li>User then make sure that the video he wants to view exists and re-enter all the details.</li> </ul> |
| Special Requirement | Efficiency                                                                                                                                                                                                                    |

*Table 19 Fully Dressed Use Case View Lecture*

### Send Notification

| Use Case Section      | Comment                                                                                                                                                                                                                                                                                                 |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case id           | UC-10                                                                                                                                                                                                                                                                                                   |
| Use Case Name         | Send Notifications                                                                                                                                                                                                                                                                                      |
| Scope                 | Registered User/Admin able to Send Notifications                                                                                                                                                                                                                                                        |
| Level                 | Admin/User send notification                                                                                                                                                                                                                                                                            |
| Primary Actor         | Admin/User                                                                                                                                                                                                                                                                                              |
| Stake Holder          | Admin/User                                                                                                                                                                                                                                                                                              |
| Precondition          | Must be logged in and can send notifications to particular students or everyone                                                                                                                                                                                                                         |
| Post Condition        | Successfully sends notifications                                                                                                                                                                                                                                                                        |
| Main Success Scenario | <ul style="list-style-type: none"> <li>User runs the notification function</li> <li>User enters the notification details like message particular contacts etc.</li> <li>After entering the details, the user press send button</li> <li>Notifications successfully sent to particular users.</li> </ul> |
| Extensions            | <ul style="list-style-type: none"> <li>If the admin/user enters the details wrong to which they want to send notifications the system show</li> </ul>                                                                                                                                                   |

|                     |                                                                                                                                                         |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     | <p>message “User wasn’t found.”</p> <ul style="list-style-type: none"> <li>Admin/User re-enter the details and again sends the notification.</li> </ul> |
| Special Requirement | Efficiency                                                                                                                                              |

*Table 20 Fully Dressed Use Case Send Notification*

## **Tutor**

| Use Case Section      | Comment                                                                                                                                                                                                                              |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case id           | UC-11                                                                                                                                                                                                                                |
| Use Case Name         | Tutor                                                                                                                                                                                                                                |
| Scope                 | Registered User able to register themselves as tutors                                                                                                                                                                                |
| Level                 | User register themselves as tutors                                                                                                                                                                                                   |
| Primary Actor         | Registered User                                                                                                                                                                                                                      |
| Stake Holder          | Registered User                                                                                                                                                                                                                      |
| Precondition          | Must be logged in and able to register themselves as tutors for a particular subject                                                                                                                                                 |
| Post Condition        | Successfully register themselves as tutors                                                                                                                                                                                           |
| Main Success Scenario | <ul style="list-style-type: none"> <li>User login to the system</li> <li>User register themselves as tutors for particular subjects</li> </ul>                                                                                       |
| Extensions            | <ul style="list-style-type: none"> <li>If a user skips some details about tutor registration then the system will not add user as a tutor.</li> <li>Tutor must have to enter all the details and details must be correct.</li> </ul> |
| Special Requirement   | Efficiency                                                                                                                                                                                                                           |

*Table 21 Fully Dressed Use Case Tutor*

## **View Tutor**

| Use Case Section | Comment |
|------------------|---------|
|------------------|---------|

|                       |                                                                                                                                                                                                         |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case id           | UC-12                                                                                                                                                                                                   |
| Use Case Name         | View Tutor                                                                                                                                                                                              |
| Scope                 | Admin, Registered User and Visitors able to View Tutors List                                                                                                                                            |
| Level                 | Admin/User/Visitors View Tutors                                                                                                                                                                         |
| Primary Actor         | Admin/User/Visitors                                                                                                                                                                                     |
| Stake Holder          | Admin/User/Visitors                                                                                                                                                                                     |
| Precondition          | Admin, Users and Visitors able to find tutor for a particular subject                                                                                                                                   |
| Post Condition        | Successfully view Tutors                                                                                                                                                                                |
| Main Success Scenario | <ul style="list-style-type: none"> <li>• Visitor/User can view tutor from tutor list by entering specific details</li> </ul>                                                                            |
| Extensions            | <ul style="list-style-type: none"> <li>• If User/Visitor views tutor for a specific subject and there is no tutor in the list then the message appears "Tutor wasn't found for this subject"</li> </ul> |
| Special Requirement   | Efficiency                                                                                                                                                                                              |

*Table 22 Fully Dressed Use Case View Tutor*

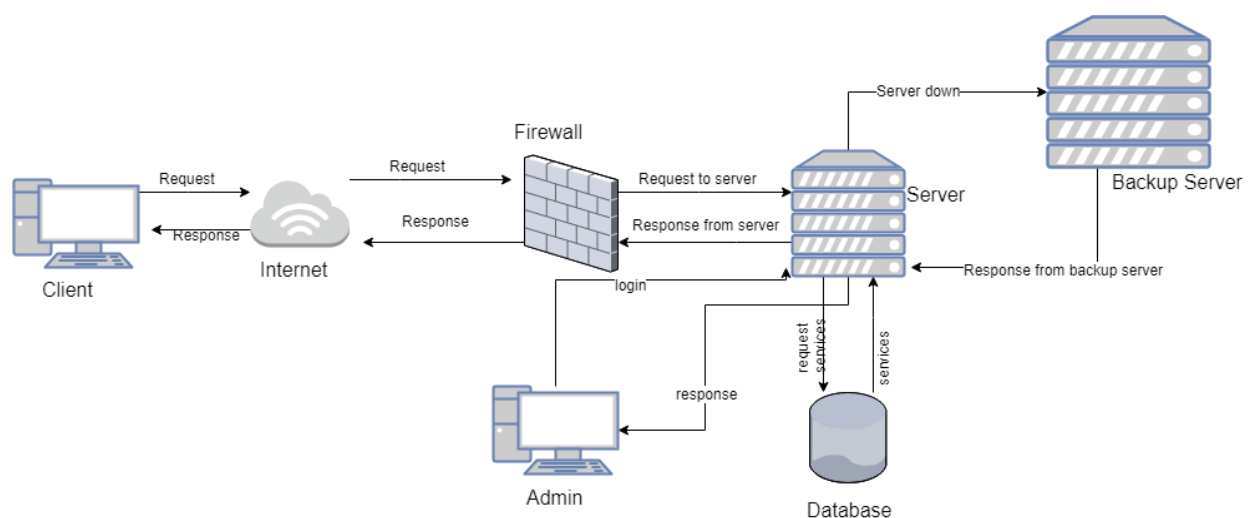
# Chapter 4

## System Design

## Chapter 4: System Design

In this chapter we have attached some UML diagrams like Architecture, Domain Model, Class Diagram, Activity Diagram, Component Diagram, Sequence Diagram, Deployment Diagram, ER Diagram etc. By the help of these diagrams we have explained about our whole system that how it works. The each and every function of our system has been explained in this chapter by the help of these diagrams.

### 4.1. Architecture Diagram



*Figure 7 Architecture Diagram*

### 4.2. Domain Model

A space express describes important abstract classes in an issue area. It is the most essential antiquity to make amid protest-situated study. A space display is generally utilized as a wellspring of motivation for planning programming objects.

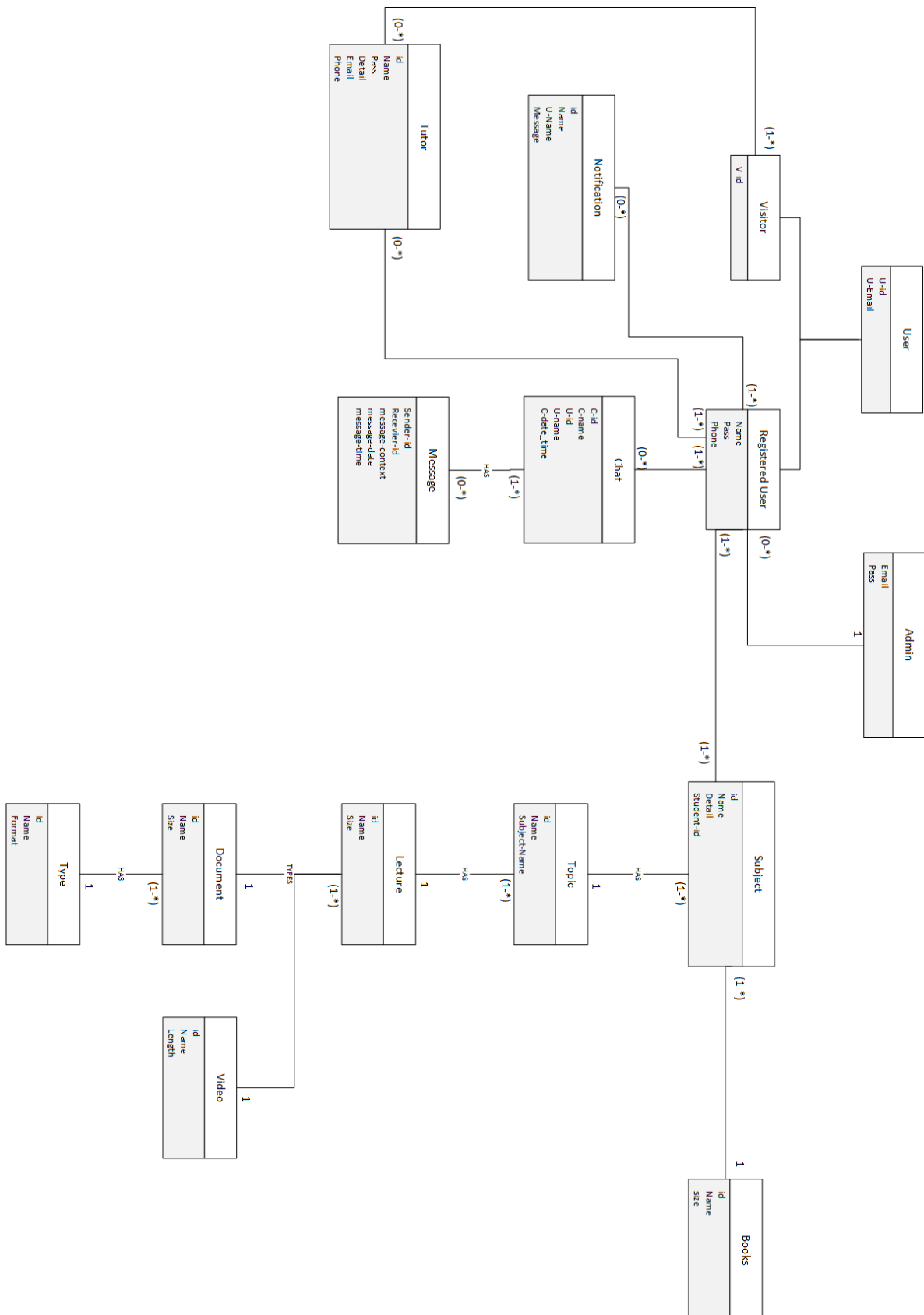


Figure 8 Domain Model

### 4.3. Entity Relationship Diagram with data dictionary

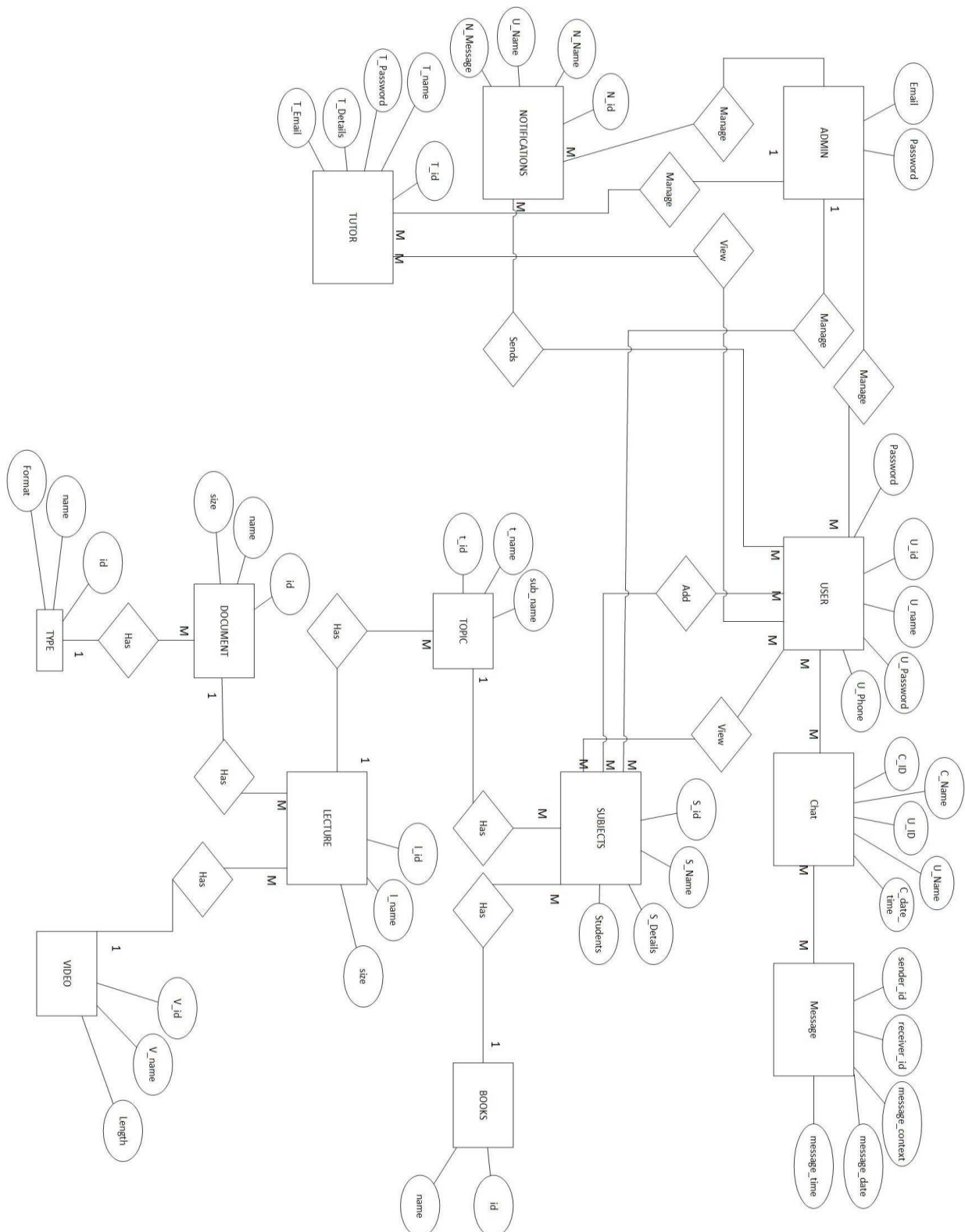


Figure 9 Entity Relationship diagram



## Data Dictionary

|                           |                                                     |                  |                               |                  |
|---------------------------|-----------------------------------------------------|------------------|-------------------------------|------------------|
| <b>Entity Name</b>        | User                                                |                  |                               |                  |
| <b>Entity Description</b> | It stores complete information of User account data |                  |                               |                  |
| <b>Column Name</b>        | <b>Column Description</b>                           | <b>Data Type</b> | <b>Constraints</b>            | <b>Reference</b> |
| U_Id                      | It stores id of the user                            | INT              | Primary Key, Unique, Not Null |                  |
| U_Name                    | It stores name of the user                          | VARCHAR          | Unique, Not Null              |                  |
| U_password                | It stores password of the user                      | VARCHAR          | Not Null                      |                  |
| U_Phone                   | It stores contact number of the user                | VARCHAR          | Not Null                      |                  |

*Table 23 Data Dictionary User*

|                           |                                                      |                  |                               |                  |
|---------------------------|------------------------------------------------------|------------------|-------------------------------|------------------|
| <b>Entity Name</b>        | Admin                                                |                  |                               |                  |
| <b>Entity Description</b> | It stores complete information of admin account data |                  |                               |                  |
| <b>Column Name</b>        | <b>Column Description</b>                            | <b>Data Type</b> | <b>Constraints</b>            | <b>Reference</b> |
| Email                     | It contains Email of the admin                       | VARCHAR          | Primary Key, Unique, Not Null |                  |
| Password                  | It contains Password of the admin                    | VARCHAR          | Unique, Not Null              |                  |

*Table 24 Data Dictionary Admin*

|                           |                                           |                  |                               |                  |
|---------------------------|-------------------------------------------|------------------|-------------------------------|------------------|
| <b>Entity Name</b>        | Subject                                   |                  |                               |                  |
| <b>Entity Description</b> | It stores complete information of Subject |                  |                               |                  |
| <b>Column Name</b>        | <b>Column Description</b>                 | <b>Data Type</b> | <b>Constraints</b>            | <b>Reference</b> |
| S_Id                      | It stores id of the Subject               | INT              | Primary Key, Unique, Not Null |                  |
| S_Name                    | It stores name of the subject             | VARCHAR          | Unique, Not Null              |                  |
| S_Details                 | It stores details of the subject          | VARCHAR          | Not Null                      |                  |
| Student_id                | It stores id of the students              | VARCHAR          | Foreign Key, Not Null         | User             |

*Table 25 Data Dictionary Subject*

|                           |                                |                  |                    |                  |
|---------------------------|--------------------------------|------------------|--------------------|------------------|
| <b>Entity Name</b>        | Tutor                          |                  |                    |                  |
| <b>Entity Description</b> | It stores the info of Tutor    |                  |                    |                  |
| <b>Column Name</b>        | <b>Column Description</b>      | <b>Data Type</b> | <b>Constraints</b> | <b>Reference</b> |
| T_id                      | It stores the id of tutor      | int              | Not Null           |                  |
| T_name                    | It stores the name of User     | VARCHAR          | Not Null           |                  |
| T_password                | It stores the password of user | VARCHAR          | Not Null           |                  |
| T_details                 | It stores the details of Tutor | VARCHAR          | Not Null           |                  |
| T_Email                   | It stores the email of Tutor   | VARCHAR          | Not Null           |                  |

Table 26 Data Dictionary Tutor

|                           |                                    |                  |                       |                  |
|---------------------------|------------------------------------|------------------|-----------------------|------------------|
| <b>Entity Name</b>        | Chatroom                           |                  |                       |                  |
| <b>Entity Description</b> | It manages the chatroom            |                  |                       |                  |
| <b>Column Name</b>        | <b>Column Description</b>          | <b>Data Type</b> | <b>Constraints</b>    | <b>Reference</b> |
| C_id                      | It stores the id of Chat           | int              | Primary Key, Not Null |                  |
| C_Name                    | It shows latitude of the location  | VARCHAR          | Not Null              |                  |
| U_id                      | It stores the id of chat creator   | int              | Foreign Key           | User             |
| U_Name                    | It stores the name of chat creator | VARCHAR          | Foreign Key           | User             |
| C_time_date               | It stores the chat time and date   | VARCHAR          | Not Null              |                  |

Table 27 Data Dictionary Chatroom

|                           |                                    |                  |                       |                  |
|---------------------------|------------------------------------|------------------|-----------------------|------------------|
| <b>Entity Name</b>        | Notification                       |                  |                       |                  |
| <b>Entity Description</b> | It Manages the Notification        |                  |                       |                  |
| <b>Column Name</b>        | <b>Column Description</b>          | <b>Data Type</b> | <b>Constraints</b>    | <b>Reference</b> |
| N_id                      | It stores the id of Notification   | Int              | Primary Key, Not Null |                  |
| N_Name                    | It stores the name of Notification | VARCHAR          | Not Null              |                  |

|           |                                    |         |                       |      |
|-----------|------------------------------------|---------|-----------------------|------|
| U_Name    | It stores the name of sender       | VARCHAR | Foreign Key, Not Null | User |
| N_Message | It stores the Notification Message | VARCHAR | Not Null              |      |

Table 28 Data Dictionary Notification

## 4.4. Class Diagram

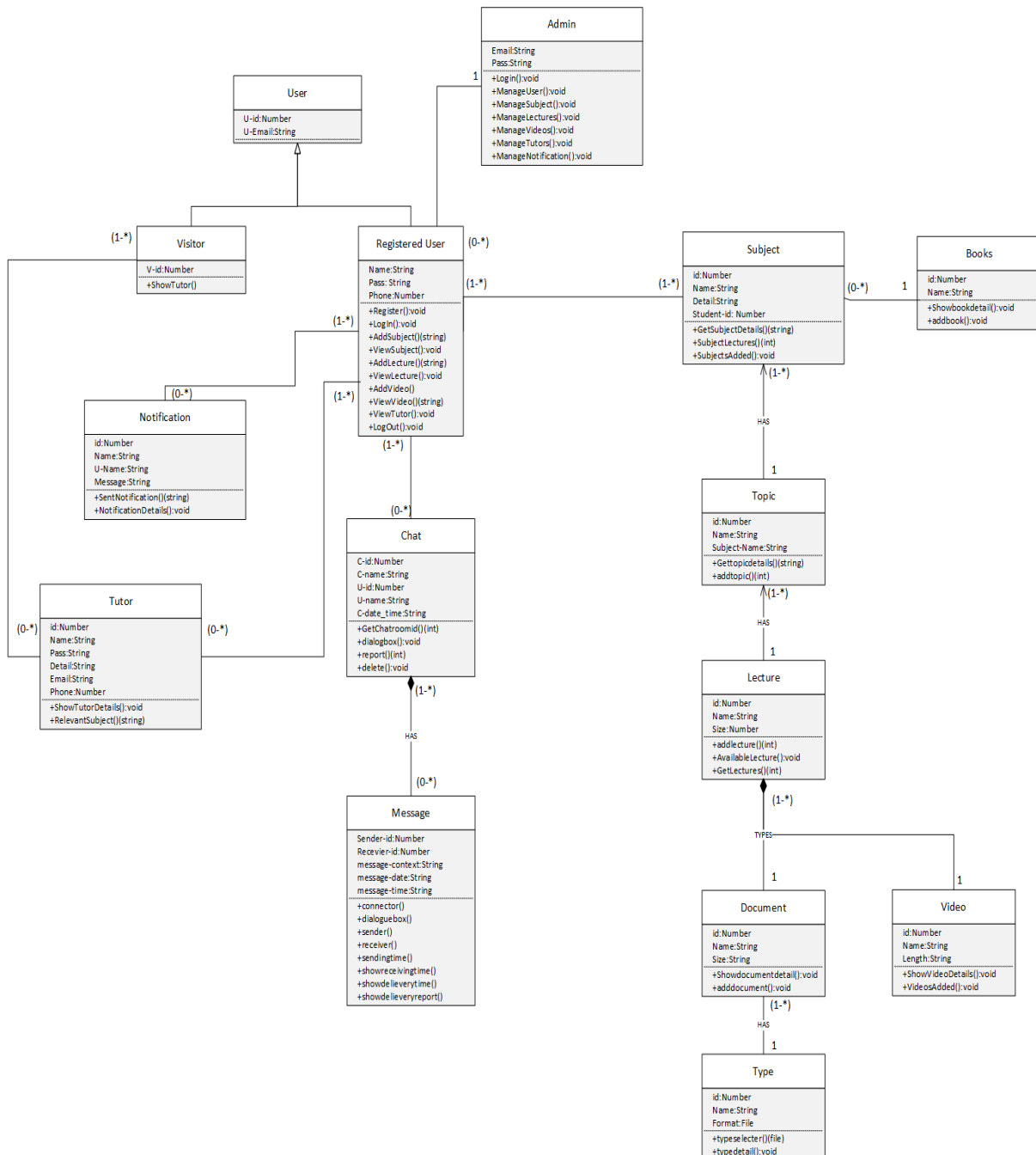


Figure 10 Class diagram

## 4.5. Sequence / Collaboration Diagram

### User Side

#### 1. Registration

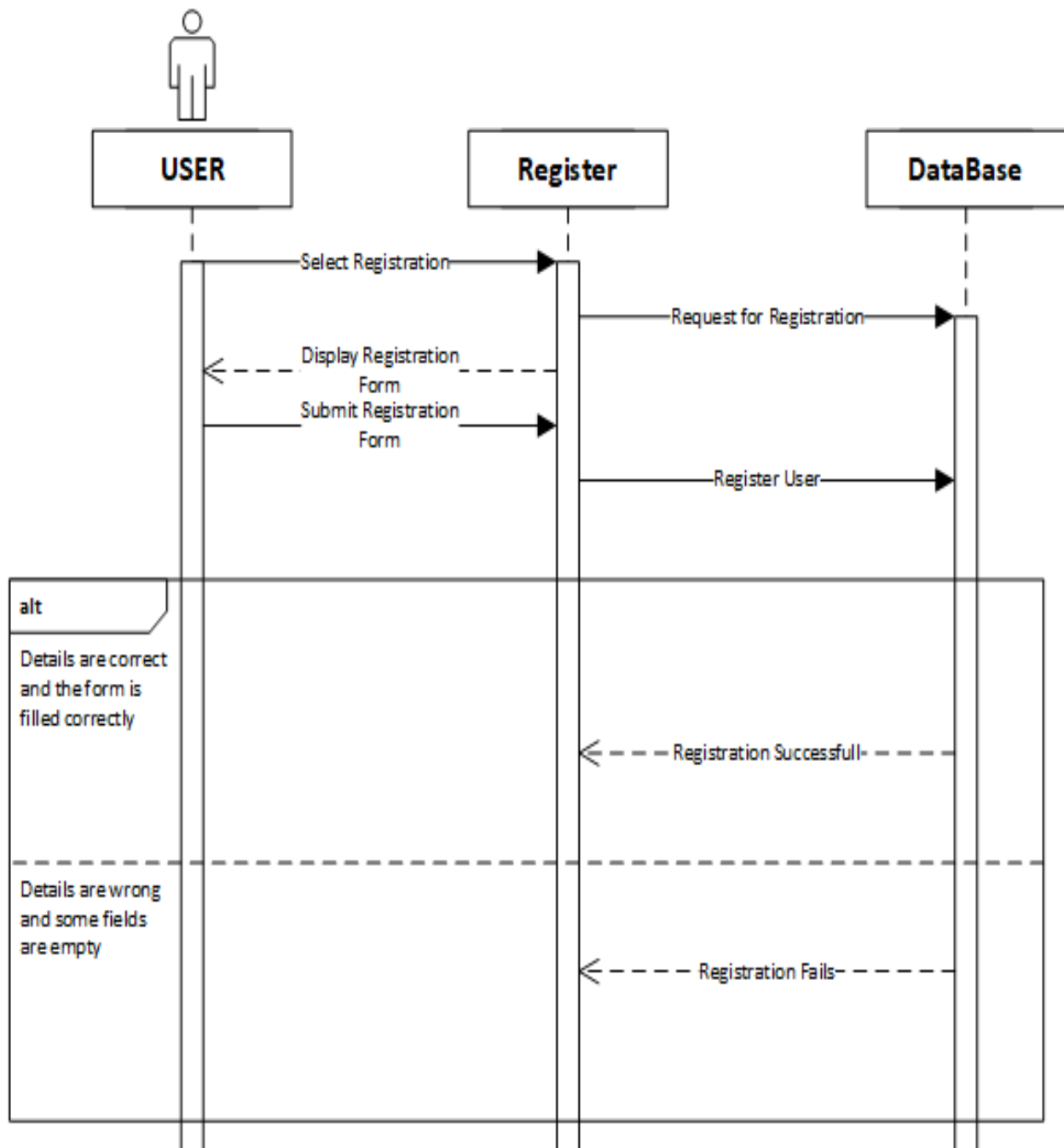


Figure 11 Sequence Diagram Registration

## 2. Login

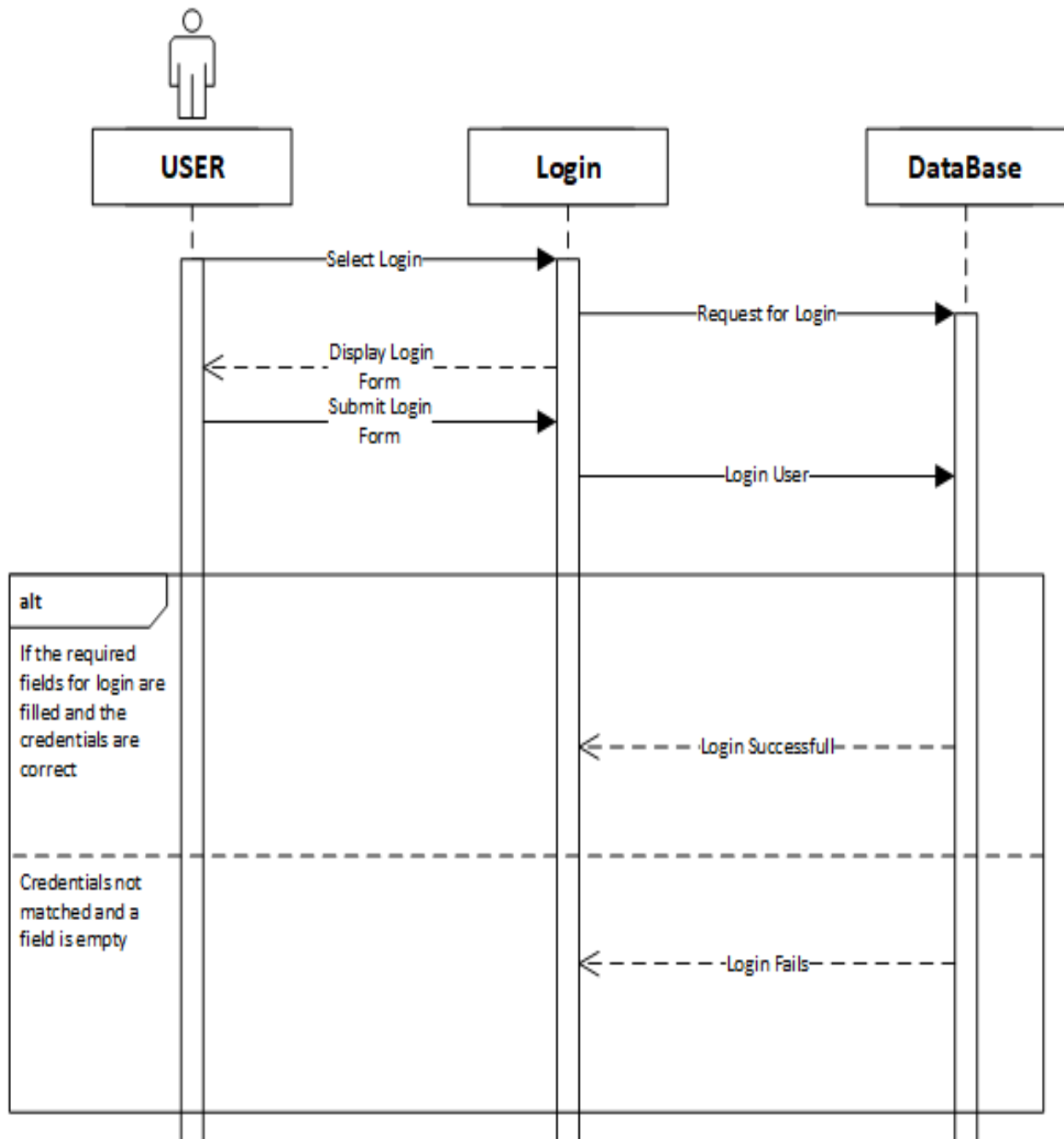


Figure 12 Sequence Diagram Login

### 3. Logout

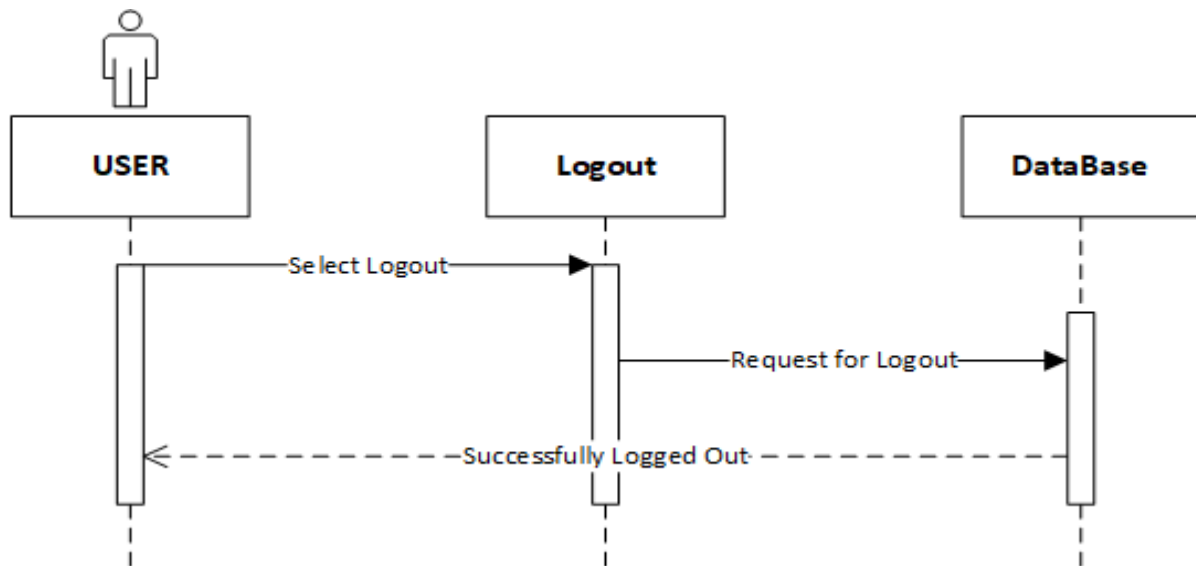


Figure 13 Sequence Diagram Logout

### 4. Add/View Subject

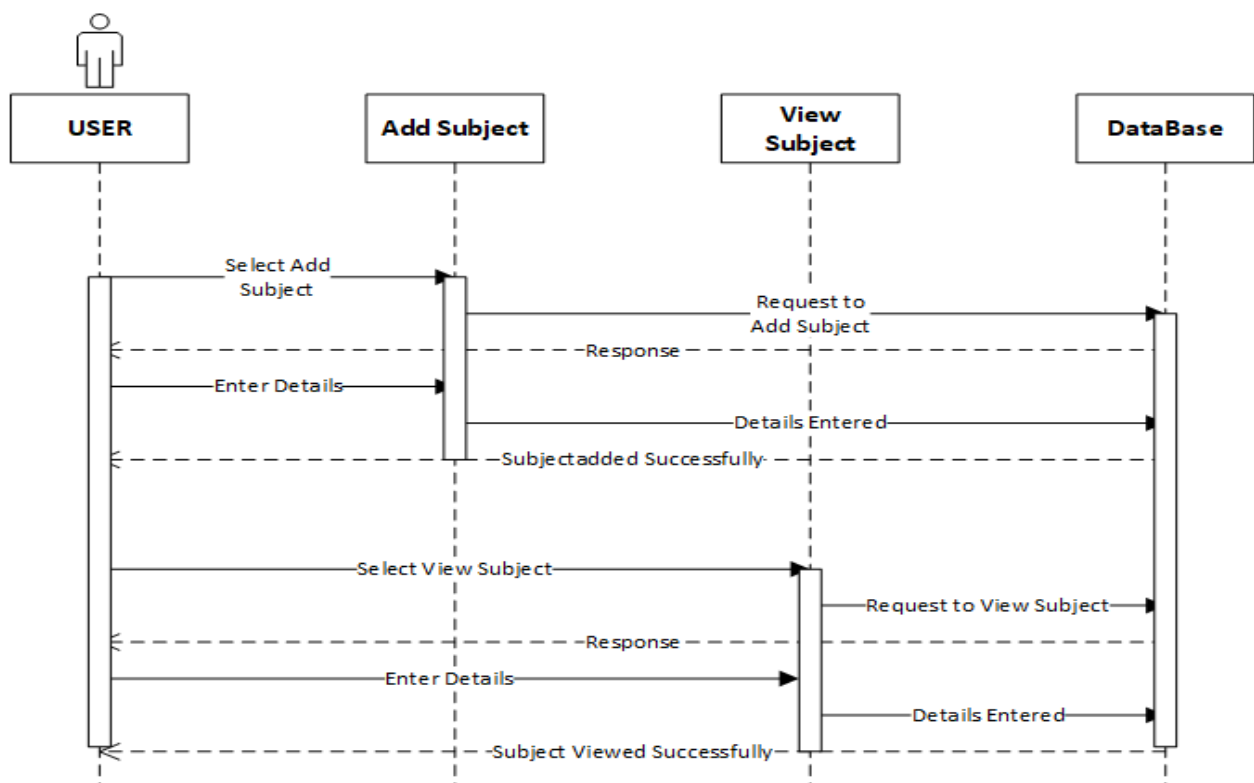


Figure 14 Sequence Diagram Add/View Subject

## 5. Add/View Lecture

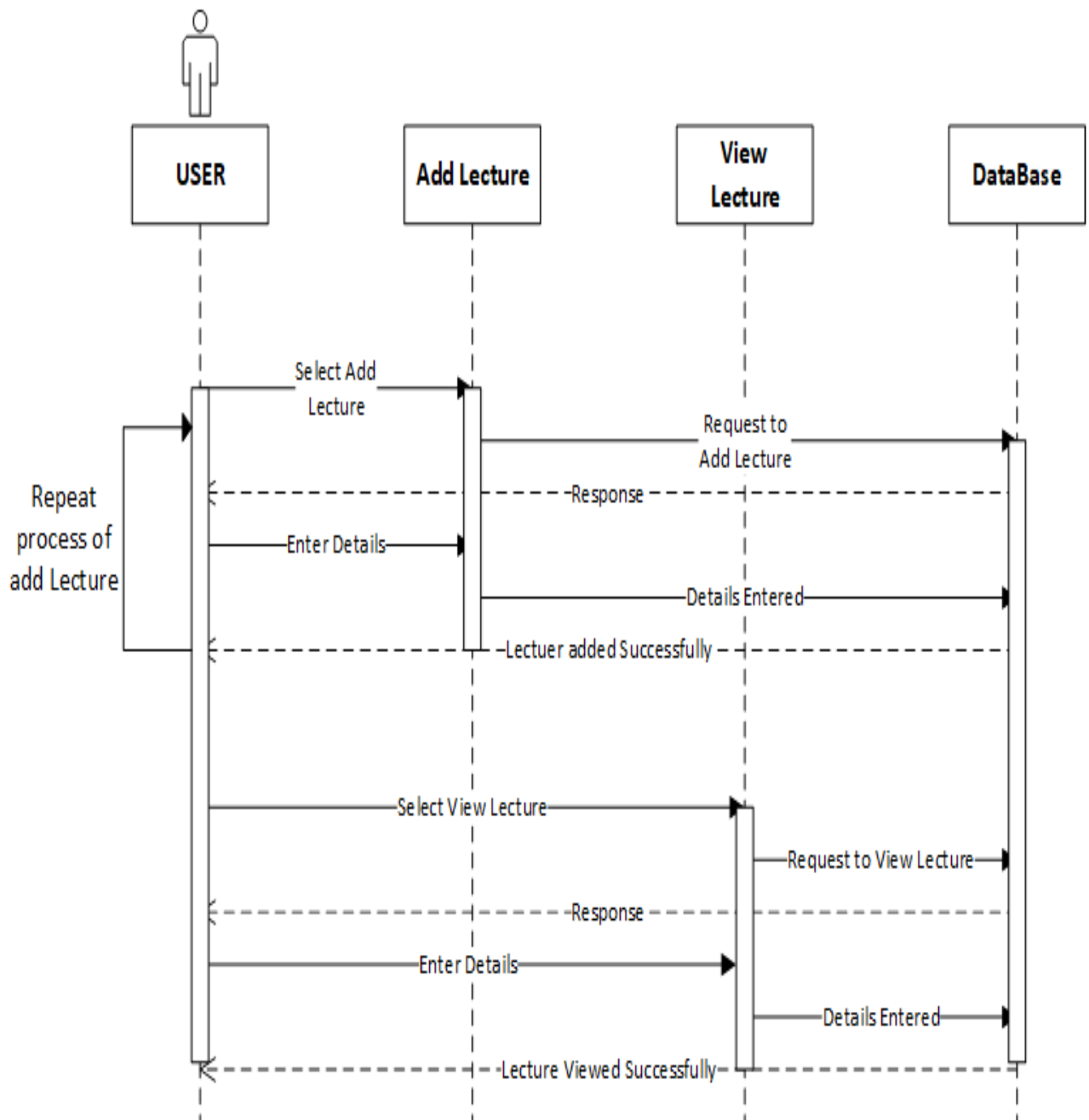


Figure 15 Sequence Diagram Add/View Lecture

## 6. Add/View Video

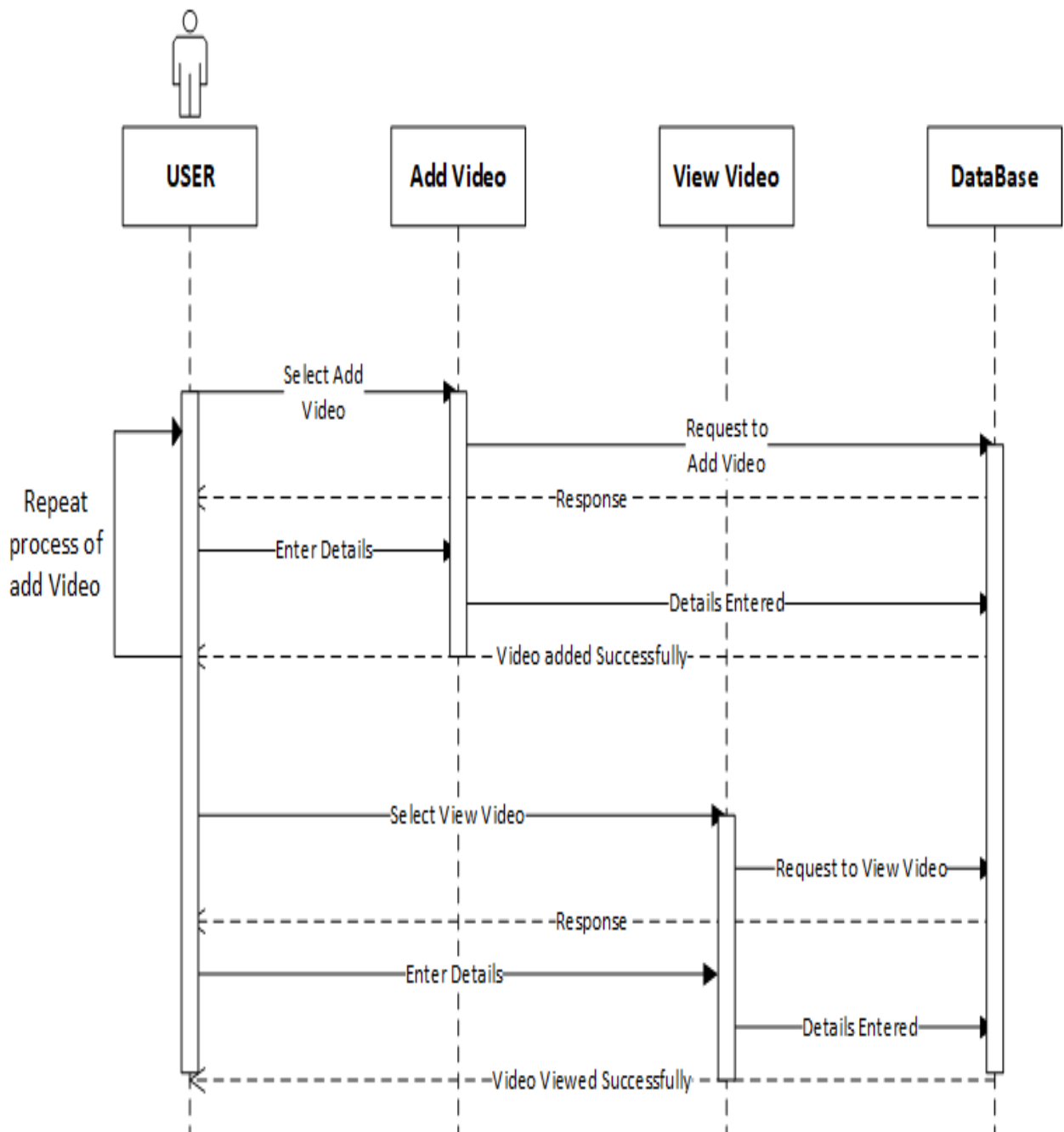


Figure 16 Sequence Diagram Add/View Video



## 7. View Tutor

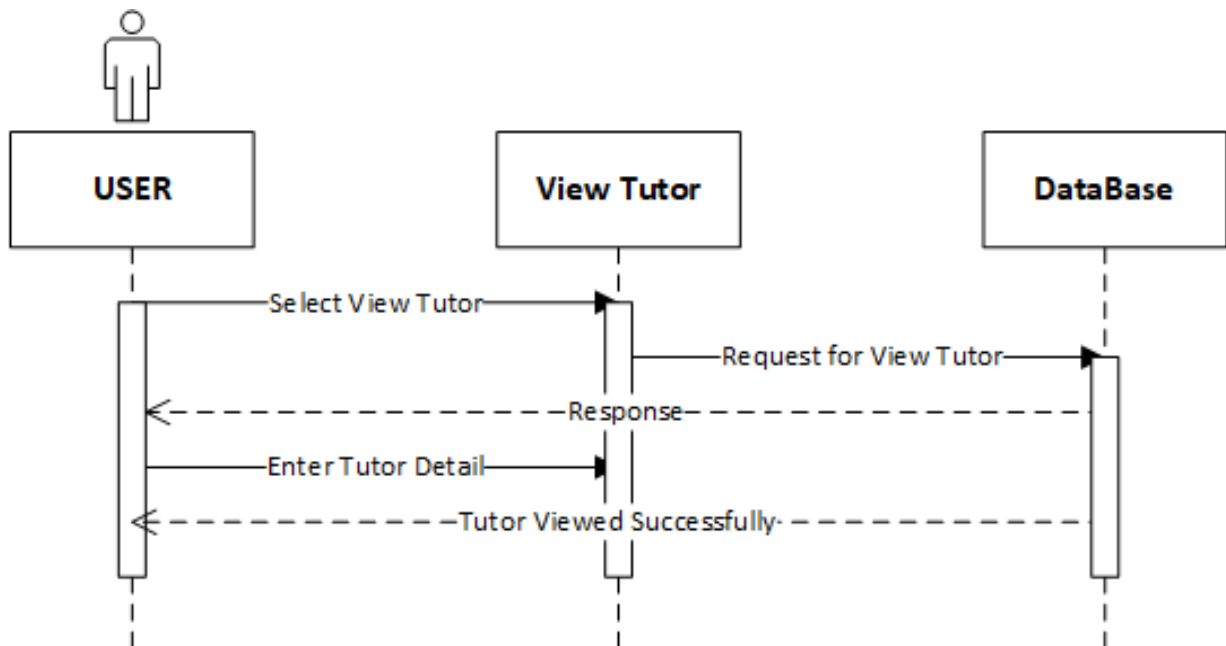


Figure 17 Sequence Diagram View Tutor

## 8. Send Notifications

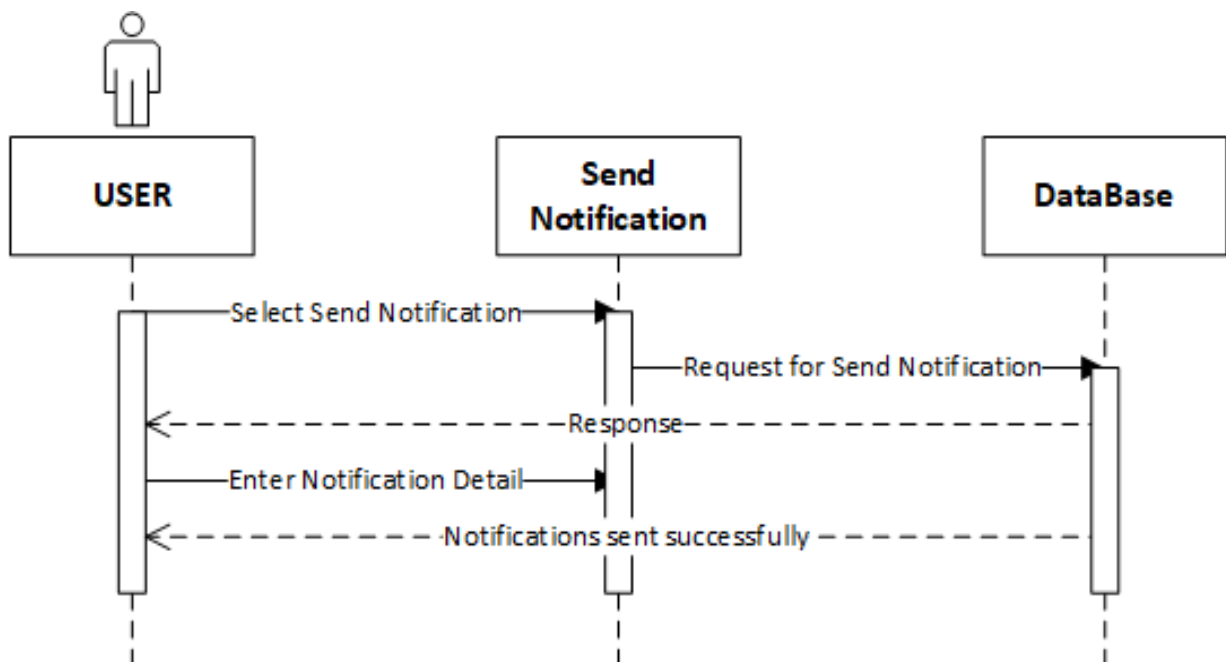


Figure 18 Sequence Diagram Send Notifications

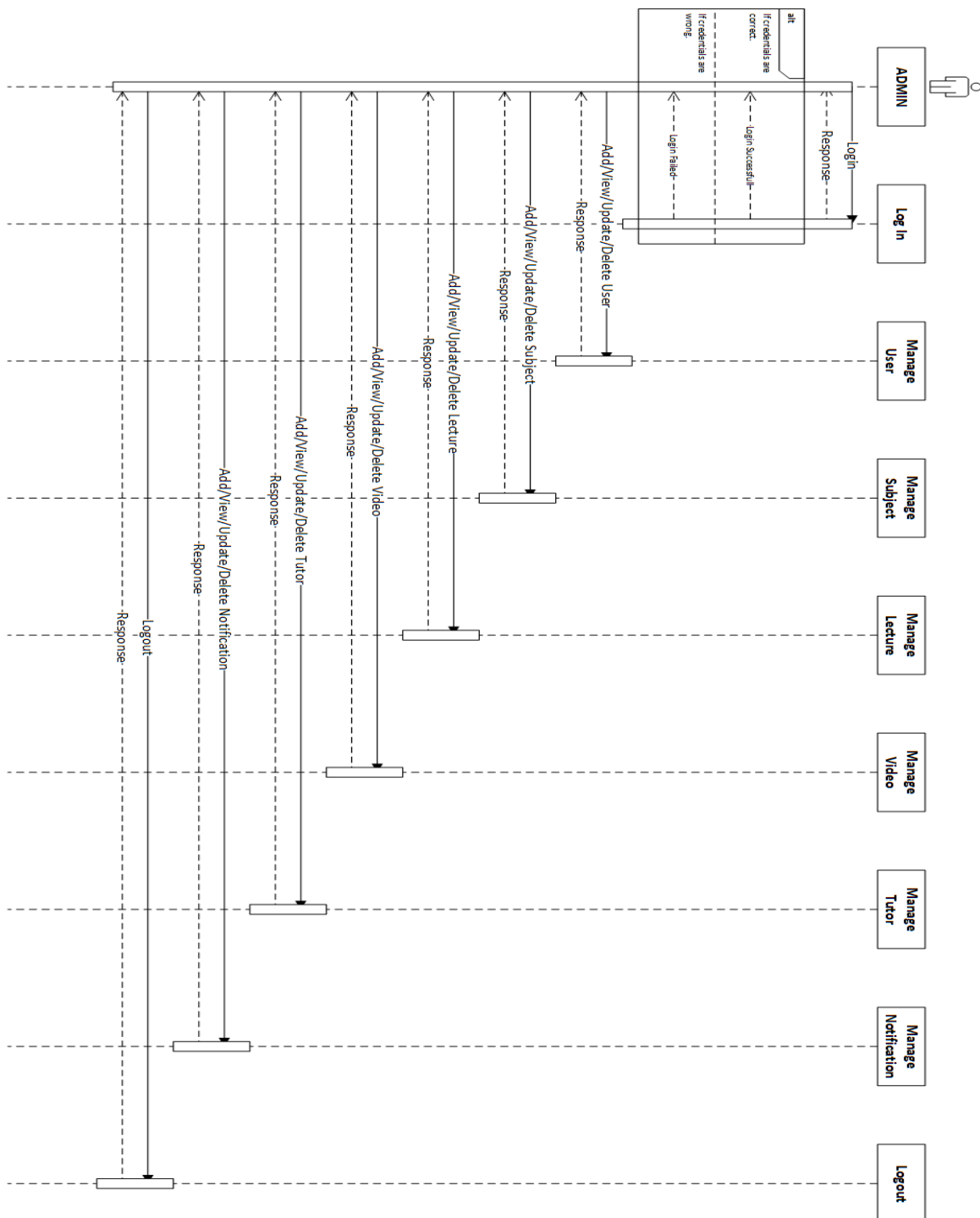
**Admin Side:**

Figure 19 Sequence Diagram Admin Side

## 4.6. Operation contracts

### 4.6.1 Operation Contract#1

**Name:** Register

**Responsibilities:** To Register the new user.

**Cross References:** Use Case: Register

**Exceptions:** None

**Preconditions:** Register interface must be opened and user must enter valid information

**Post conditions:**

- A new Registration Request instance reg\_Req was created
- reg\_Req was also associated with Admin
- A new account instance “n” was created.

### 4.6.2 Operation Contract#2

**Name:** Login

**Responsibilities:** A user is logged in.

**Cross References:** Use Case: Log In

**Exceptions:** Invalid Username & Password

**Preconditions:** User must be registered

**Post conditions:**

- A login variable “login” was created
- Instance “login” was associated with Current User
- Username and password were matched with database

### 4.6.3 Operation Contract#3

**Name:** Logout

**Responsibilities:** A user is logged out

**Cross References:** Use Case: Logout

**Exceptions:** None

**Preconditions:** User must be logged in

**Post conditions:**

- A logout variable “logout” was created
- Variable “logout” was associated with Current User
- User was successfully logged out

#### 4.6.4 Operation Contract#4

**Name:** Add Subject

**Responsibilities:** A user adds a subject.

**Cross References:** Use Case: Add Subject

**Exceptions:** Already Registered Subject

**Preconditions:** User must be logged in

**Post conditions:**

- A subject variable “add\_subject” was created
- Variable “add\_subject” was associated with Current User/Admin
- Subject was successfully added in database

#### 4.6.5 Operation Contract#5

**Name:** View Subject

**Responsibilities:** A user views a subject.

**Cross References:** Use Case: View Subject

**Exceptions:** Subject not registered

**Preconditions:** User must be logged in

**Post conditions:**

- A variable “view\_subject” was created
- Variable “view\_subject” was associated with Current User/Admin
- Subject was viewed successfully

### 4.6.6 Operation Contract#6

**Name:** Add Lecture

**Responsibilities:** A user adds a lecture.

**Cross References:** Use Case: Add Lecture

**Exceptions:** Subject not registered

**Preconditions:** User must be logged in and have a relevant subject

**Post conditions:**

- A lecture variable “add\_lecture” was created
- Variable “add\_lecture” was associated with Current User/Admin
- Lecture was successfully added in database

### 4.6.7 Operation Contract#7

**Name:** View Lecture

**Responsibilities:** A user views a lecture.

**Cross References:** Use Case: View Lecture

**Exceptions:** Lecture not found

**Preconditions:** User must be logged in and able to access lecture

**Post conditions:**

- A variable “view\_lecture” was created
- Variable “view\_lecture” was associated with Current User/Admin
- Lecture was viewed successfully

### 4.6.8 Operation Contract#8

**Name:** Add Video

**Responsibilities:** A user adds a video.

**Cross References:** Use Case: Add Video

**Exceptions:** Relevant subject was not found

**Preconditions:** User must be logged in and have an existing lecture

**Post conditions:**

- A lecture variable “add\_video” was created

- Variable “add\_video” was associated with Current User/Admin
- Video was successfully added in database

#### 4.6.9 Operation Contract#9

**Name:** View Video

**Responsibilities:** A user views a video.

**Cross References:** Use Case: View Video

**Exceptions:** Relevant video was not found

**Preconditions:** User must be logged in and able to access video

**Post conditions:**

- A variable “view\_video” was created
- Variable “view\_video” was associated with Current User/Admin
- Video was viewed successfully

#### 4.6.10 Operation Contract#10

**Name:** Send Notification

**Responsibilities:** A user/admin able to send notifications

**Cross References:** Use Case: Send Notification

**Exceptions:** Particular student wasn't found

**Preconditions:** User must be logged in

**Post conditions:**

- A notification variable “notif” was created
- Variable “notif” was associated with Current User/Admin
- Notification was sent successfully

#### 4.6.11 Operation Contract#11

**Name:** Tutor

**Responsibilities:** A user able to apply as tutor.

**Cross References:** Use Case: Tutor

**Exceptions:** Relevant subject wasn't found

**Preconditions:** User must be logged in

**Post conditions:**

- A tutor variable "tutor" was created
- Variable "tutor" was associated with Current User
- User had successfully applied as tutor

## 4.6.12 Operation Contract#12

**Name:** View Tutor

**Responsibilities:** A user/Admin/visitor able to view tutors

**Cross References:** Use Case: View Tutor

**Exceptions:** Relevant tutor wasn't found

**Preconditions:** User must be logged in

**Post conditions:**

- A variable "view\_tutor" was created
- Variable "view\_tutor" was associated with Current User/Admin/Visitor
- Tutor was viewed successfully

## 4.7. Activity Diagram

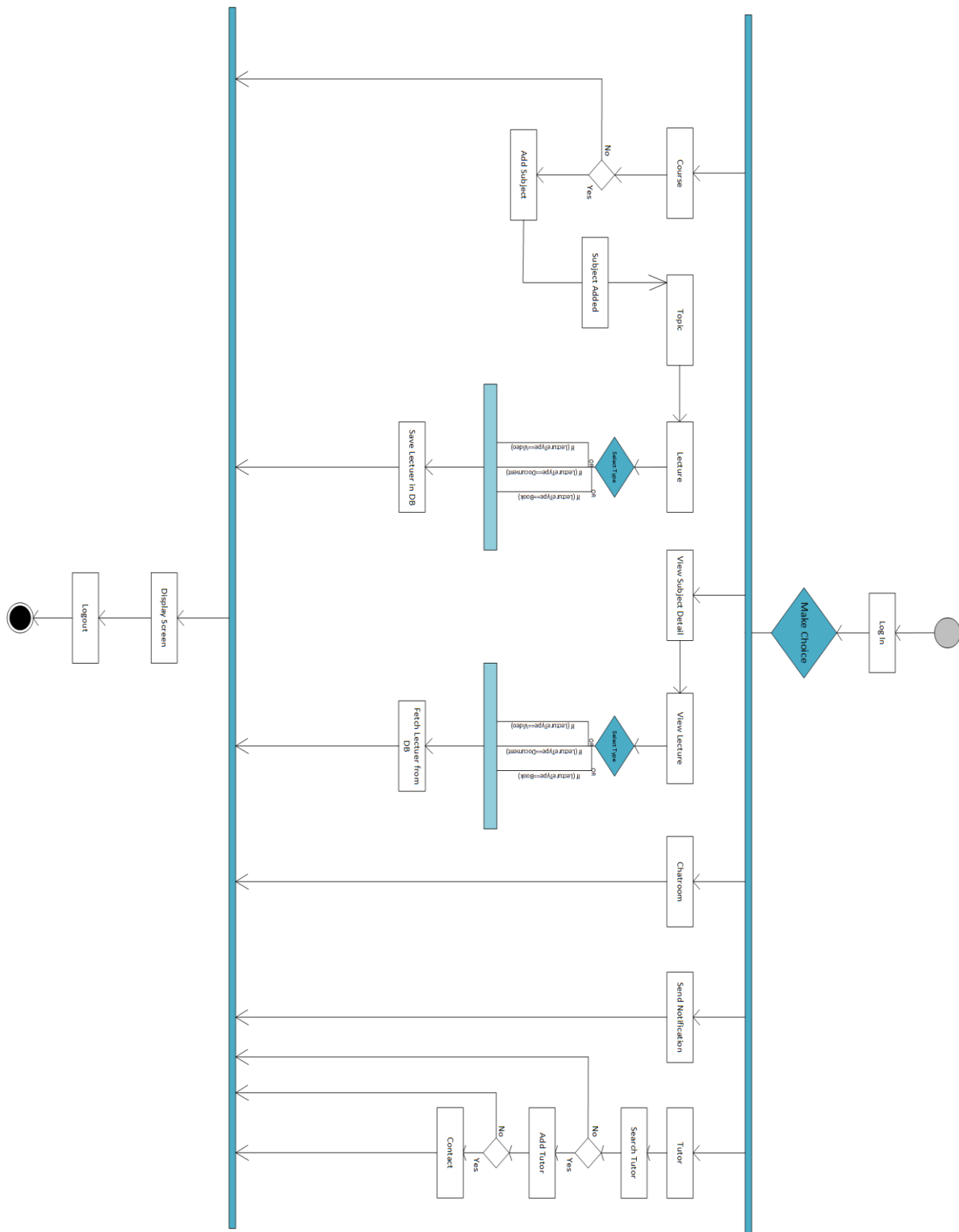


Figure 20 Activity Diagram



## 4.8. State Transition Diagram

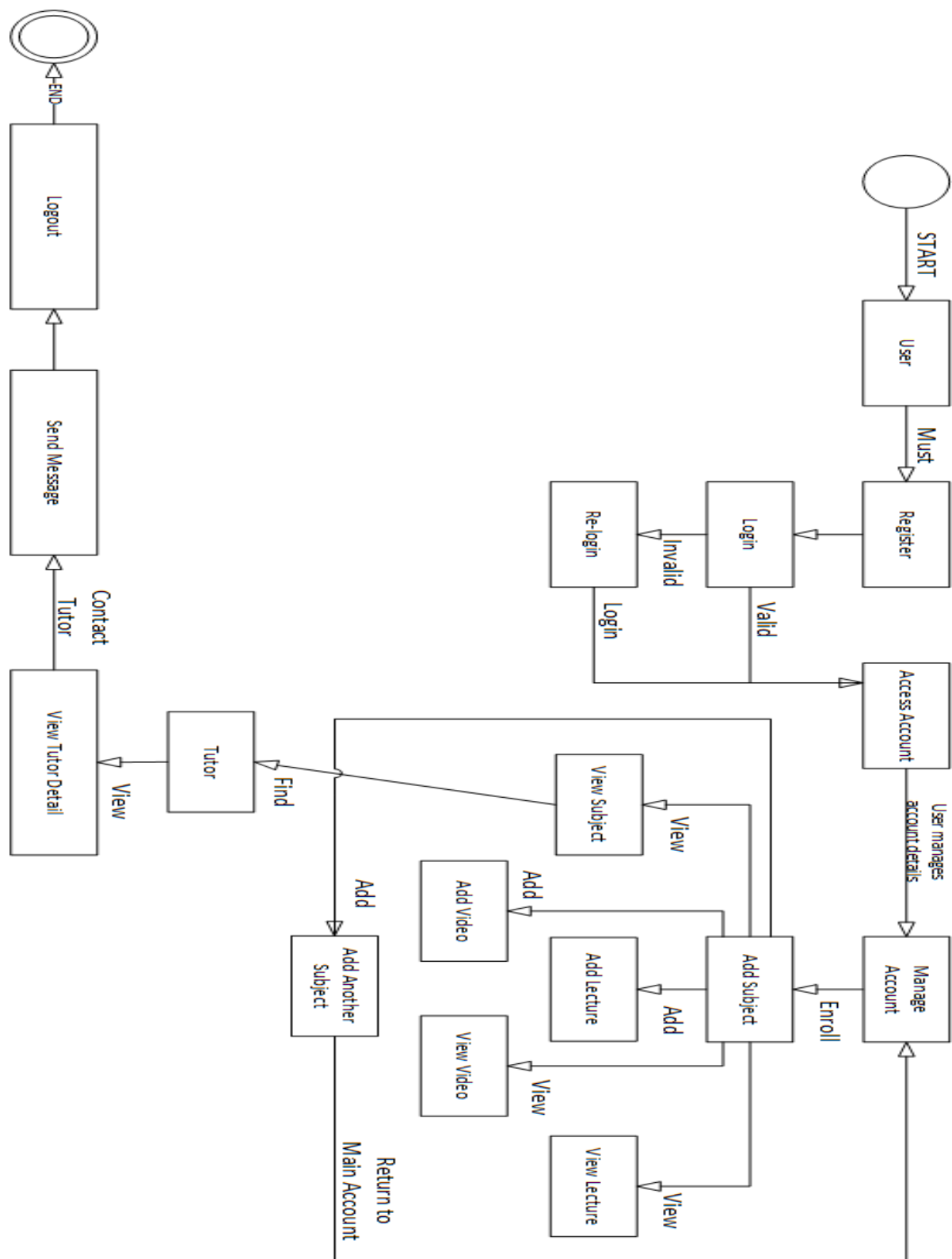


Figure 21 State transition diagram

## 4.9. Component Diagram

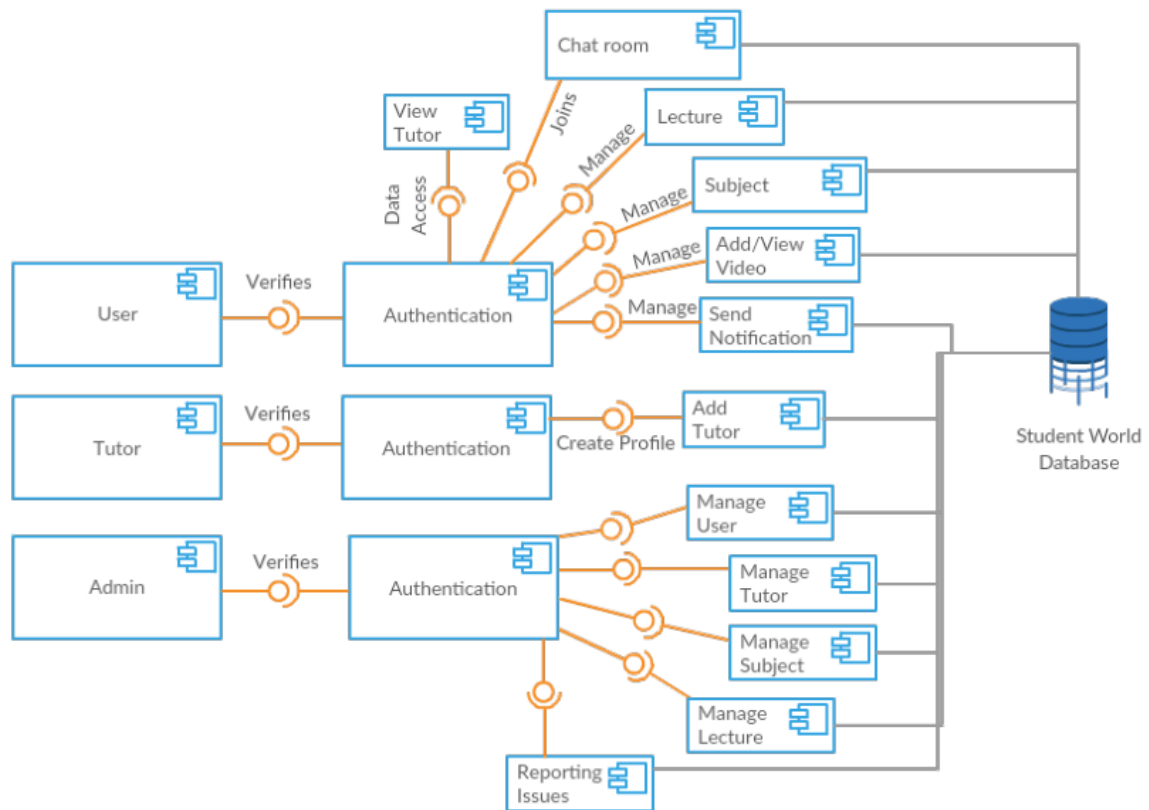
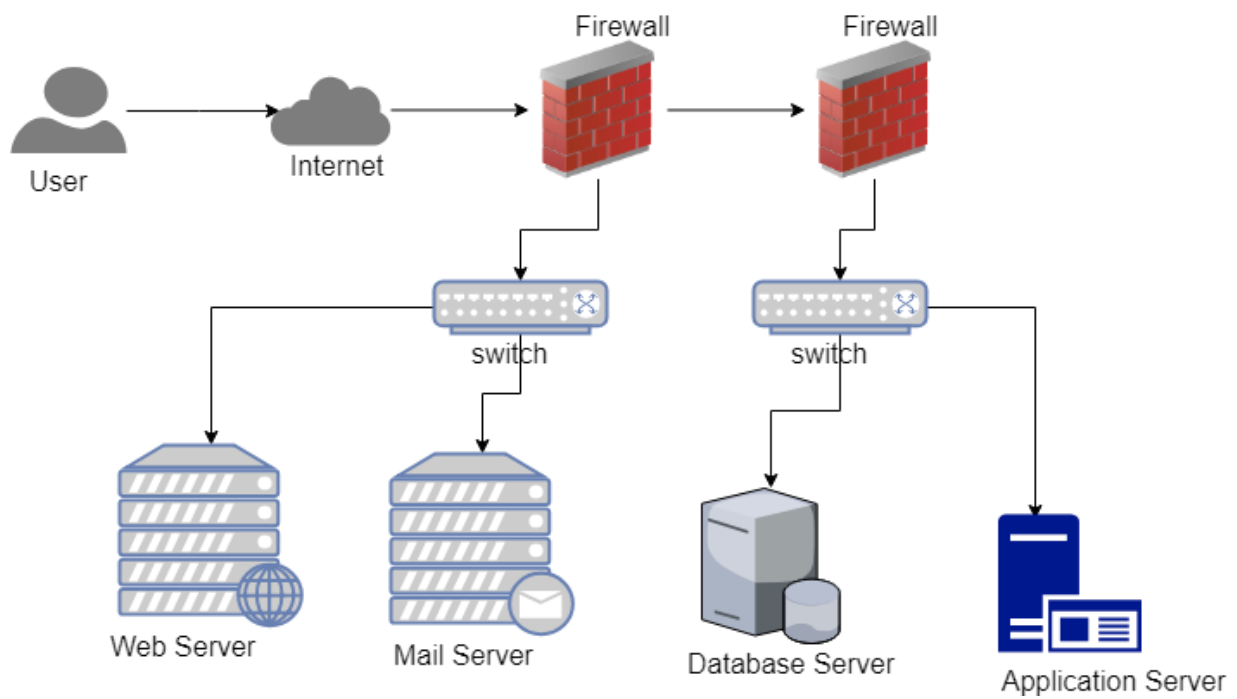


Figure 22 Component diagram

## 4.10. Deployment Diagram



*Figure 23 deployment diagram*

# Chapter 5

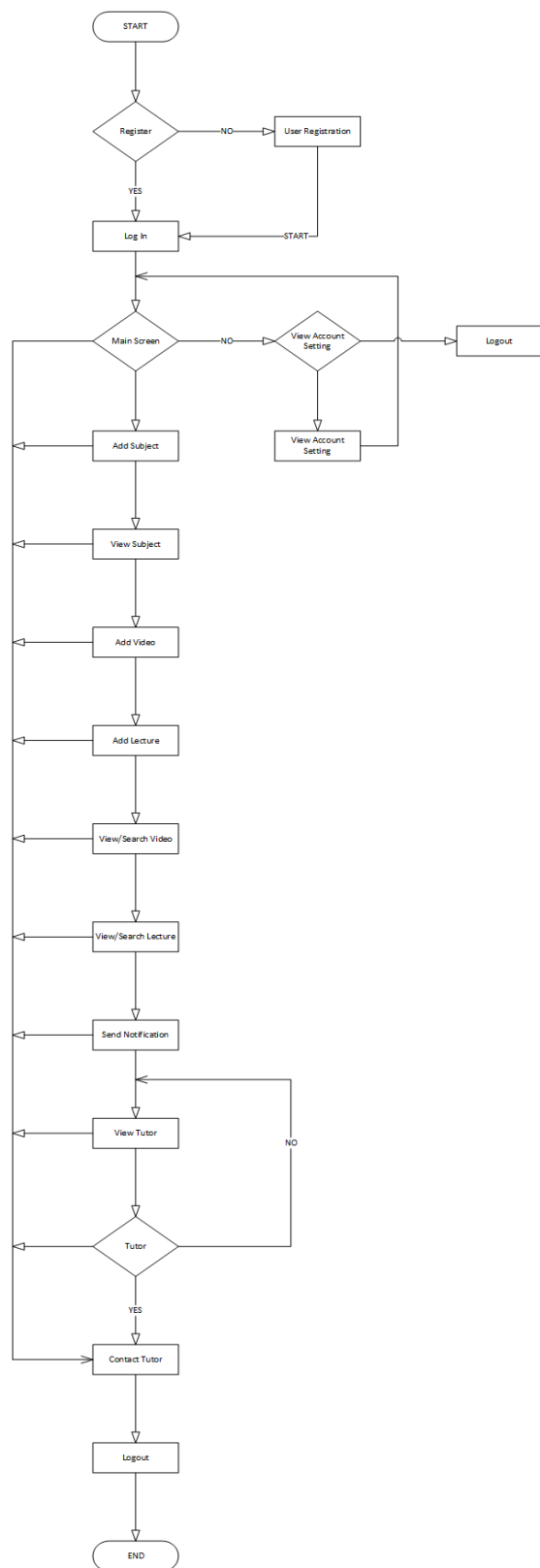
## Implementation

## Chapter 5: Implementation

Our main goal is to provide each and every student with the help they need in their study including the lectures, video tutorials, PPT slides, complete courses etc. The scope of our project is that students can manage their study material easily and they can have chatrooms available for each course to communicate with other students. They can also find most accurate and relevant data. A Particular Student can upload their own lectures and will also be able to help other students and the benefit they get is that they can be hired by anyone as a tutor.

This project has well accurately and perfectly designed database so that no data can be misplaced or lost. It has well designed front end that will attract the viewers and looks cool. It is a friendly user interface.

## 5.1. Important Flow Control/Pseudo codes



## 5.2. Components, Libraries, Web Services and stubs

### Components:

#### 1. Properly Structured Website

Student World is a platform that is easy to use. If a student wants any kind of lectures and tutorials they can find it from here. Our platform has a simple structure and it has friendly user interface with the most important lectures, video lectures and tutors for your help. You can find any type of study related material and tutors for yourself and for others.

#### 2. Mobile Friendly experience

Today, the world of technology is evolving quickly; more and more people have been using their mobile devices to study online. My Student world website is mobile friendly. This website mobile navigation is easy to use and Student World having friendly environment.

### Web Services

- Users straightforwardly add/view any subject, Lectures and videos from our platform without any problem.
- 24-hr access
- Relevant Lectures
- Tutors for help
- Video Tutorials

## 5.3. Deployment Environment

Here we need a domain on which the Student World system will exists and through the domain several people will access the application. And smart phone users can open the responsive website from any browser on either Apple phone or Android phone through which they can easily access the service and can easily receive documents. It required one administrator who will be in charge of the approval and other supervision duty.

## 5.4. Tools and Techniques

Applications and tools that would be used during this project preparation are as follows:

**ASP.Net/C#**

These are the languages, which are supported in the .NET framework/Laravel Framework. It provides latest Functionalities regarding desktop and mobile based application development.

**Microsoft Word**

Microsoft word is the word processor and would be used for documentation purpose Throughout the whole project.

**Microsoft Excel**

Microsoft excel is spreadsheet and would be used for data analysis throughout the Whole project.

**Microsoft Project**

Microsoft Projects provides extensive features on project management with a Prominent feature of project scheduling. Gantt charts, resource allocations. This would be Used as a project management tool throughout the whole project.

**Project Modeling**

Rational Rose and Microsoft Visio are to be used to draw use cases with different Perspectives as they support the language of Rational Unified Process (RUP). Use case Diagram, Data Flow Diagram (DFD), domain model and all other OOAD analysis and Design diagrams are supported.

**Server**

Server is required where whole data is stored.

**5.5. Best Practices / Coding Standards**

- No coding repetitions.
- Function and classes are always commented showing expected input and output.



# Chapter 6

## Testing and Evaluation

## Chapter 6: Testing and Evaluation

We have developed an online education system in which students can manage their subjects and lectures according to their need. Students can login to their accounts and add courses or join already registered course according to the categories that have been listed there. After registering or joining a course student can add lectures, tutorials, documents etc. of that particular course. A test strategy is methodology that describes the various steps that need to be performed during testing and the time and effort required for performing them. The following strategies are used for testing:

- Use Case Testing
- Equivalence Partitioning
- Boundary Value Analysis
- Data Flow Testing
- Unit Testing
- Integration Testing
- Performance Testing
- Stress Testing

### 6.1. Use Case Testing

#### Register

| Main Success Scenario | Step | Description                                                                                                                                                   |
|-----------------------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| A=Actor               | 1    | A: Fill Form: name, email, password, dob, phone.                                                                                                              |
| S=System              | 2    | S: For name field:<br>Jack,<br>For email field: john123@gmail.com<br>For password field:<br>123xyz!#<br>For dob:<br>Dd/mm/yyyy<br>For phone:<br>+911255236624 |

|           |   |                      |
|-----------|---|----------------------|
|           | 3 | Allow Account Access |
| Extension |   | Null                 |

Table 29 Use Case Testing Register

Figure 24 Use Case Testing Register

## Login

| Main Success Scenario | Step | Description                                          |
|-----------------------|------|------------------------------------------------------|
| A=Actor               | 1    | A: Enter Email and password                          |
| S=System              | 2    | S: Valid Password                                    |
|                       | 3    | Allow Account Access                                 |
| Extension             | 2a   | S: Invalid Password<br>Message Display on the Screen |

|  |    |                                           |
|--|----|-------------------------------------------|
|  |    | asks for Re-enter password.               |
|  | 2b | Password not valid.<br>Application close. |

Table 30 Use Case Testing Login

Figure 25 Use Case Testing Login

### Logout

| Main Success Scenario | Step | Description                                           |
|-----------------------|------|-------------------------------------------------------|
| A=Actor               | 1    | A: Enter Logout                                       |
| S=System              | 2    | S: System Show Message to User                        |
| Extension             | 2a   | S: User must login<br>System Take user to home screen |

Table 31 Use Case Testing Logout

**Add Course:**

| Main Success Scenario | Step | Description                 |
|-----------------------|------|-----------------------------|
| A=Actor               | 1    | A: Fill Form and add course |
| S=System              | 2    | S: Course added to database |
| Extension             | 2a   | S: User must login.         |

*Table 32 Use Case Testing Add Course*
*Figure 26 Use Case Testing Add Course***Add Lecture:**

| Main Success Scenario | Step | Description                  |
|-----------------------|------|------------------------------|
| A=Actor               | 1    | A: Fill Form and add lecture |
| S=System              | 2    | S: Lecture added to database |
| Extension             | 2a   | S: User must login.          |

*Table 33 Use Case Testing Add Lecture*

Figure 27 Use Case Testing Add Lecture

### View Lecture

| Main Success Scenario | Step | Description                              |
|-----------------------|------|------------------------------------------|
| A=Actor               | 1    | A: View Detail of the Lectures.          |
| S=System              | 2    | S: System Show the list of the Lectures. |
| Extension             | 2a   | S: User must login                       |

Table 34 Use Case Testing View Lecture

| Lecture Name | Course Name | File                            |
|--------------|-------------|---------------------------------|
| Invoice      | Computer    | ~/Files/Invoice.pdf             |
| SQL FILE     | Computer    | ~/Files/neoschool.sql           |
| Invoice      | Computer    | ~/Files/Furqan Malik.docx       |
| Invoice      | Computer    | ~/Files/1.jpg                   |
| Image        | Computer    | ~/Files/IMG-20190421-WA0008.jpg |

Figure 28 Use Case Testing View Lecture

**Add Video:**

| Main Success Scenario | Step | Description                |
|-----------------------|------|----------------------------|
| A=Actor               | 1    | A: Fill Form and add video |
| S=System              | 2    | S: Video added to database |
| Extension             | 2a   | S: User must login.        |

*Table 35 Use Case Testing Add Video*
*Figure 29 Use Case Testing Add Video***View Video**

| Main Success Scenario | Step | Description                             |
|-----------------------|------|-----------------------------------------|
| A=Actor               | 1    | A: View Video of the Course.            |
| S=System              | 2    | S: System Shows the list of the Videos. |
| Extension             | 2a   | S: User must login                      |

*Table 36 Use Case Testing View Video*

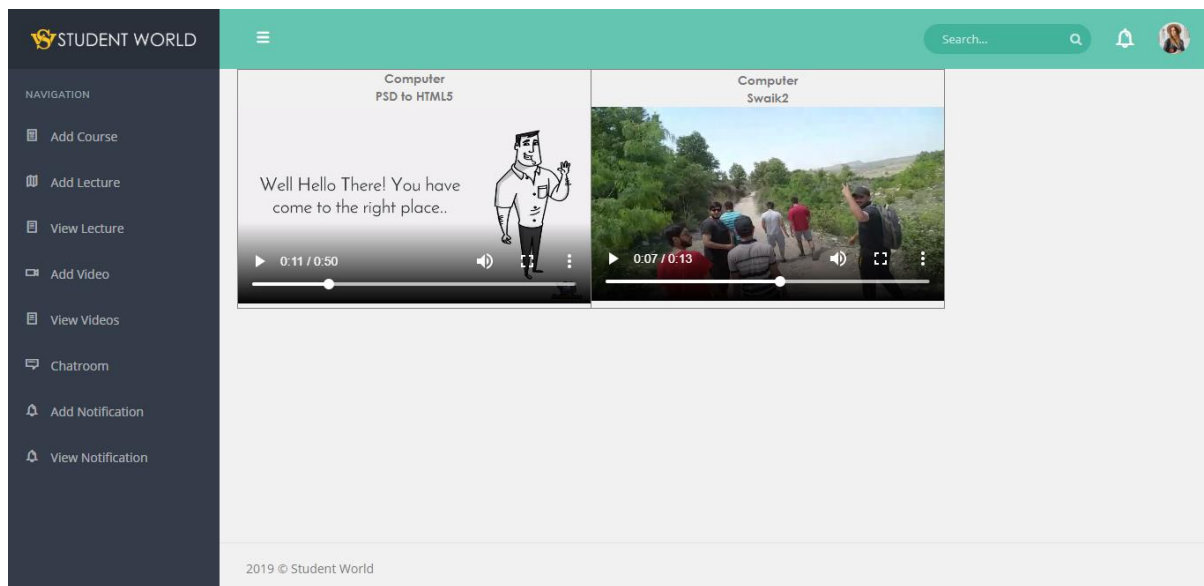


Figure 30 Use Case Testing View Video

### Chatroom:

| Main Success Scenario | Step | Description                              |
|-----------------------|------|------------------------------------------|
| A=Actor               | 1    | A: User enters name and start group chat |
| S=System              | 2    | S: Chat started                          |
| Extension             | 2a   | S: User must login.                      |

Table 37 Use Case Testing Chatroom

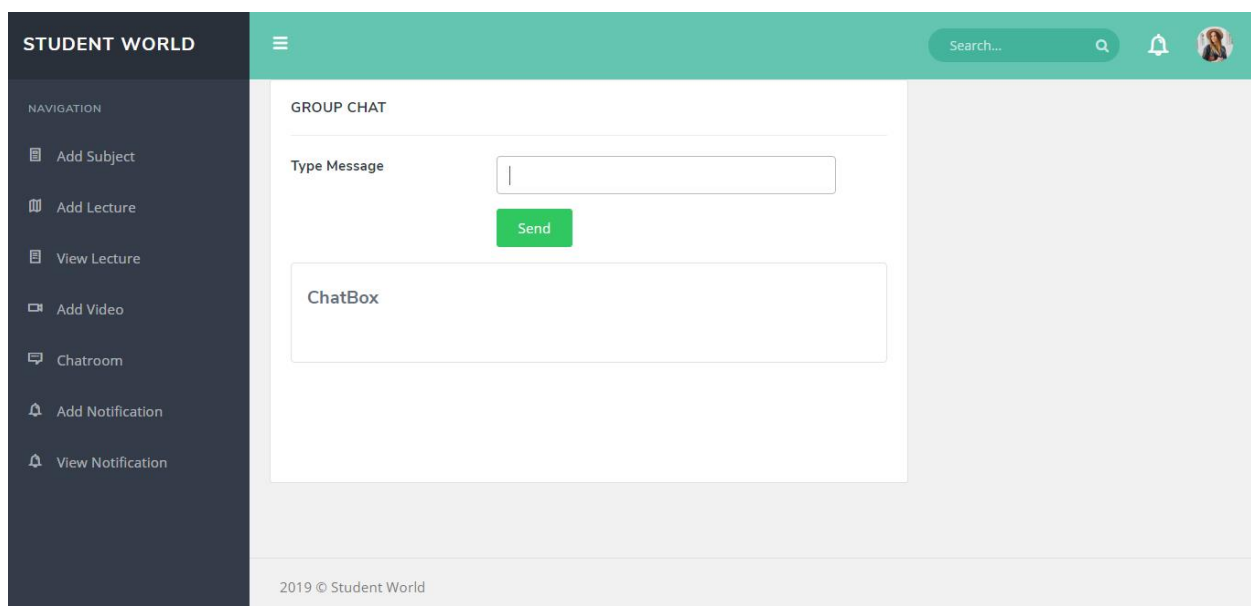


Figure 31 Use Case Testing Chatroom



### Add Notification:

| Main Success Scenario | Step | Description                            |
|-----------------------|------|----------------------------------------|
| A=Actor               | 1    | A: Enter notification message and send |
| S=System              | 2    | S: Video added to database             |
| Extension             | 2a   | S: User must login.                    |

*Table 38 Use Case Testing Add Notification*

*Figure 32 Use Case Testing Add Notification*

### View Notification

| Main Success Scenario | Step | Description                             |
|-----------------------|------|-----------------------------------------|
| A=Actor               | 1    | A: View Notification of the Course.     |
| S=System              | 2    | S: System Shows the list of the Videos. |
| Extension             | 2a   | S: User must login                      |

*Table 39 Use Case Testing View Notification*

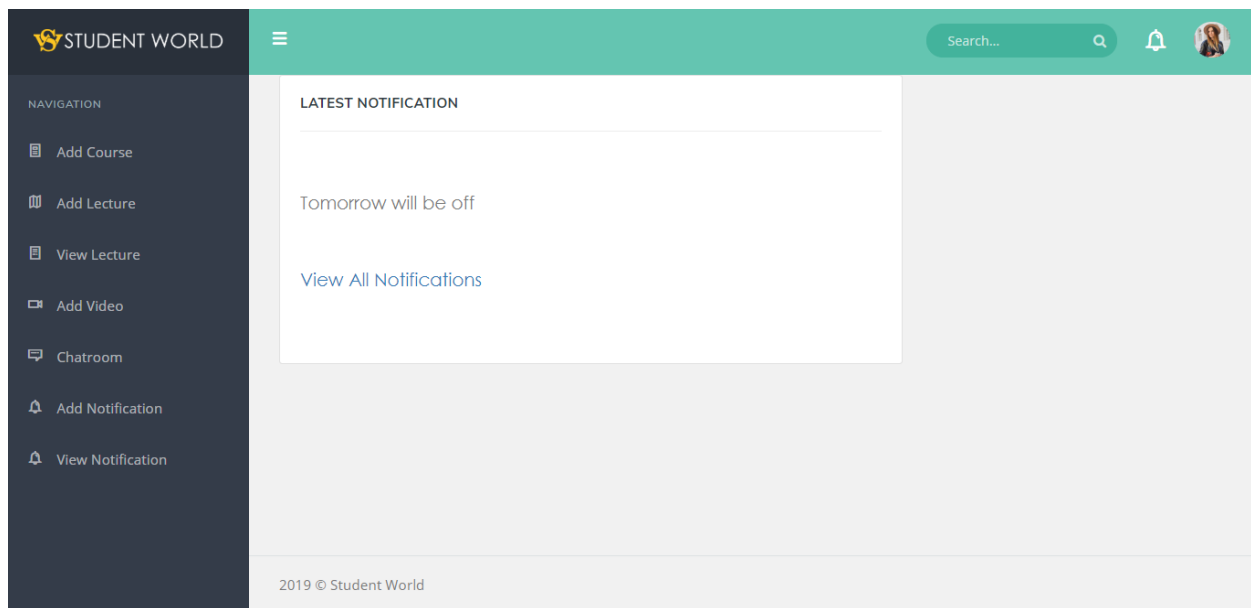


Figure 33 Use Case Testing View Notification

### View Tutors

| Main Success Scenario | Step | Description                             |
|-----------------------|------|-----------------------------------------|
| A=Actor               | 1    | A: View Notification of the Course.     |
| S=System              | 2    | S: System Shows the list of the Videos. |
| Extension             | 2a   | S: User must login                      |

Table 40 Use Case Testing View Tutors

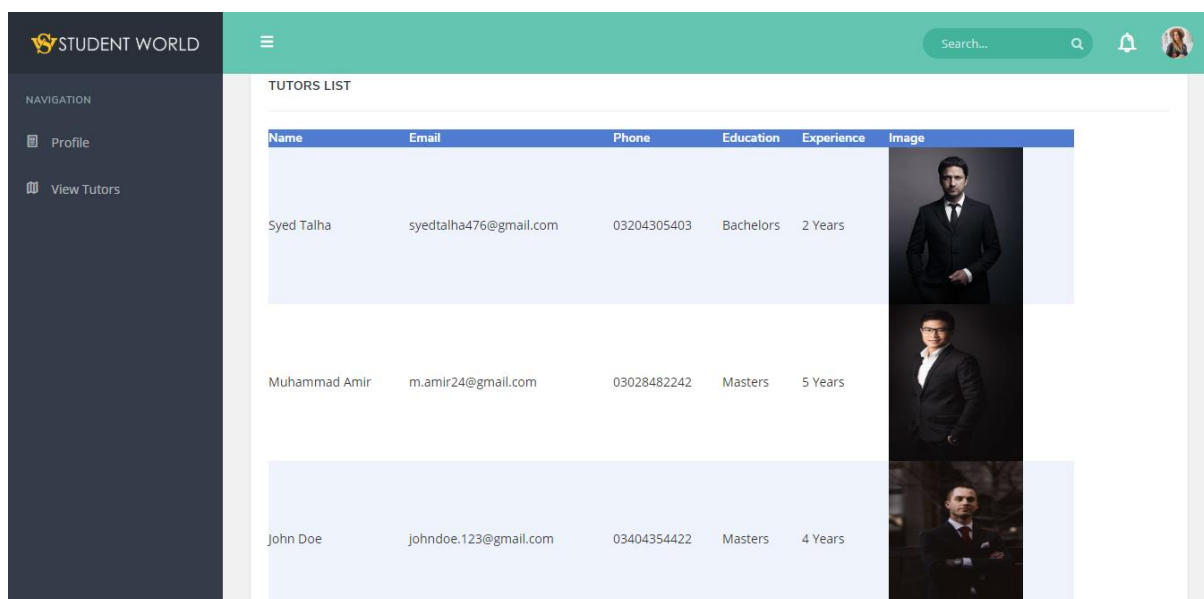


Figure 34 Use Case Testing View Tutors

## 6.2. Equivalence partitioning

### Registration

| Input    | Invalid Equivalence class Partitioning | Valid Equivalence class Partitioning                                                                                         |
|----------|----------------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| Username | Less than 3 not allow                  | Between 3 to 50 Greater than 50 not allowed                                                                                  |
| Password | Less than 8 not allow                  | Between 8 to 20 Greater than 20 not allowed                                                                                  |
| Email    | Admin123 Format is not allowed         | <a href="mailto:Admin123@outlook.com">Admin123@outlook.com</a><br><a href="mailto:Admin123@gmail.com">Admin123@gmail.com</a> |
| DOB      | 12 january 1999 is not allow           | Format will be dd/mm/yyyy                                                                                                    |
| Phone    | It must not be in alphabets<br>ABc1213 | It must be in digits<br>123456789                                                                                            |

*Table 41 Equivalence Partitioning Registration*

### Login

| Input    | Invalid Equivalence class Partitioning | Valid Equivalence class Partitioning                                                                                         |
|----------|----------------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| Email    | Admin123 Format is not allow           | <a href="mailto:Admin123@outlook.com">Admin123@outlook.com</a><br><a href="mailto:Admin123@gmail.com">Admin123@gmail.com</a> |
| Password | Less than 8 not allow                  | Between 8 to 20 Greater than 20 not allowed                                                                                  |

*Table 42 Equivalence Partitioning Login*

## 6.3. Boundary value analysis

### Register/Login

#### Email

|         |       |         |
|---------|-------|---------|
| Invalid | Valid | Invalid |
|---------|-------|---------|

| <b>(min-1)</b> | <b>(min, +max, -min ,max)</b> | <b>(max+1)</b> |
|----------------|-------------------------------|----------------|
| 3              | 3 to 50                       | 50             |

*Table 43 Boundary Value Analysis Email***Password**

| <b>Invalid<br/>(min-1)</b> | <b>Valid<br/>(min,+max,-min,max)</b> | <b>Invalid<br/>(max+1)</b> |
|----------------------------|--------------------------------------|----------------------------|
| 8                          | 8 to 25                              | 25                         |

*Table 44 Boundary Value Analysis Password*

## 6.4. Data flow testing.

### Login

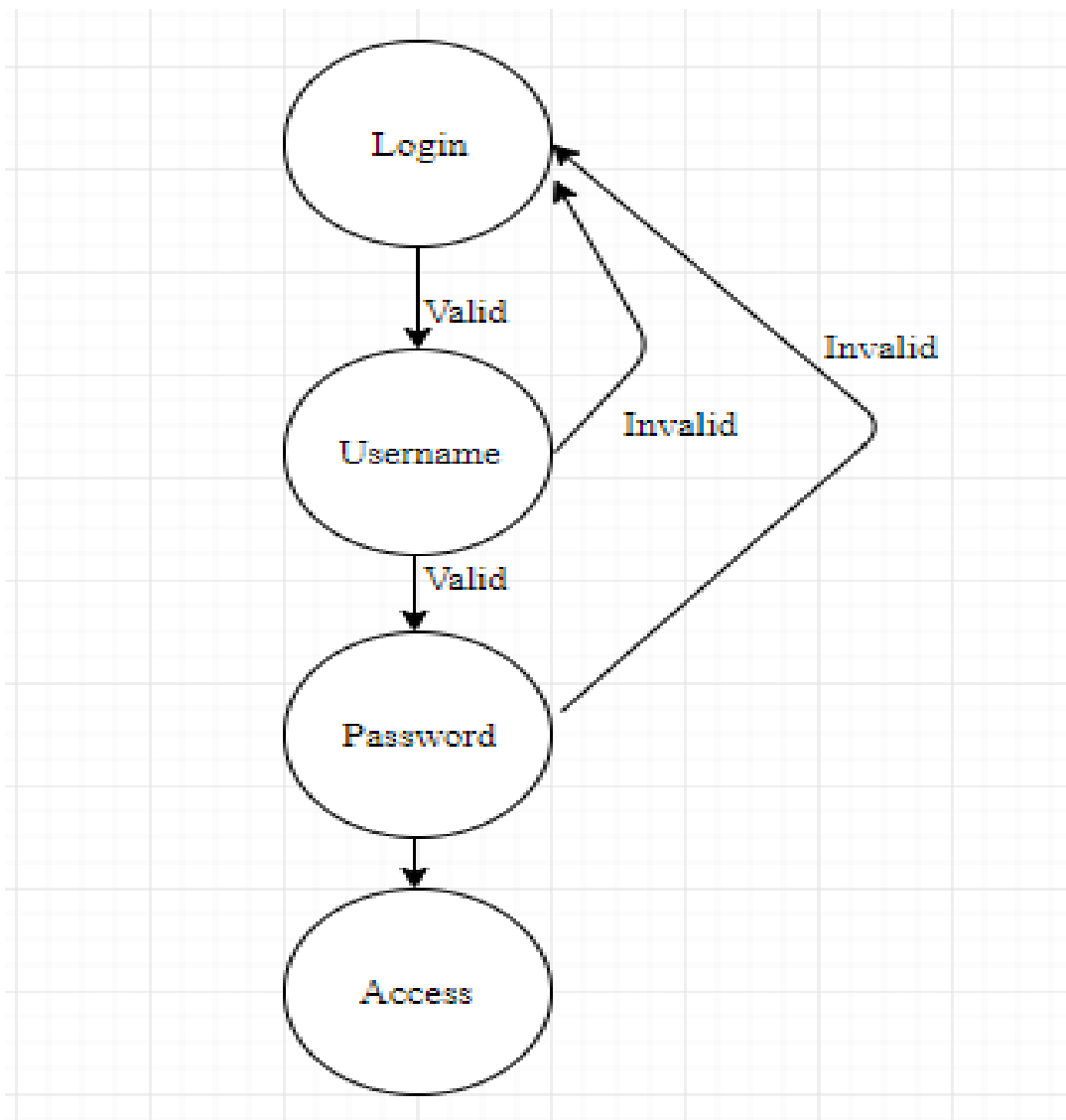
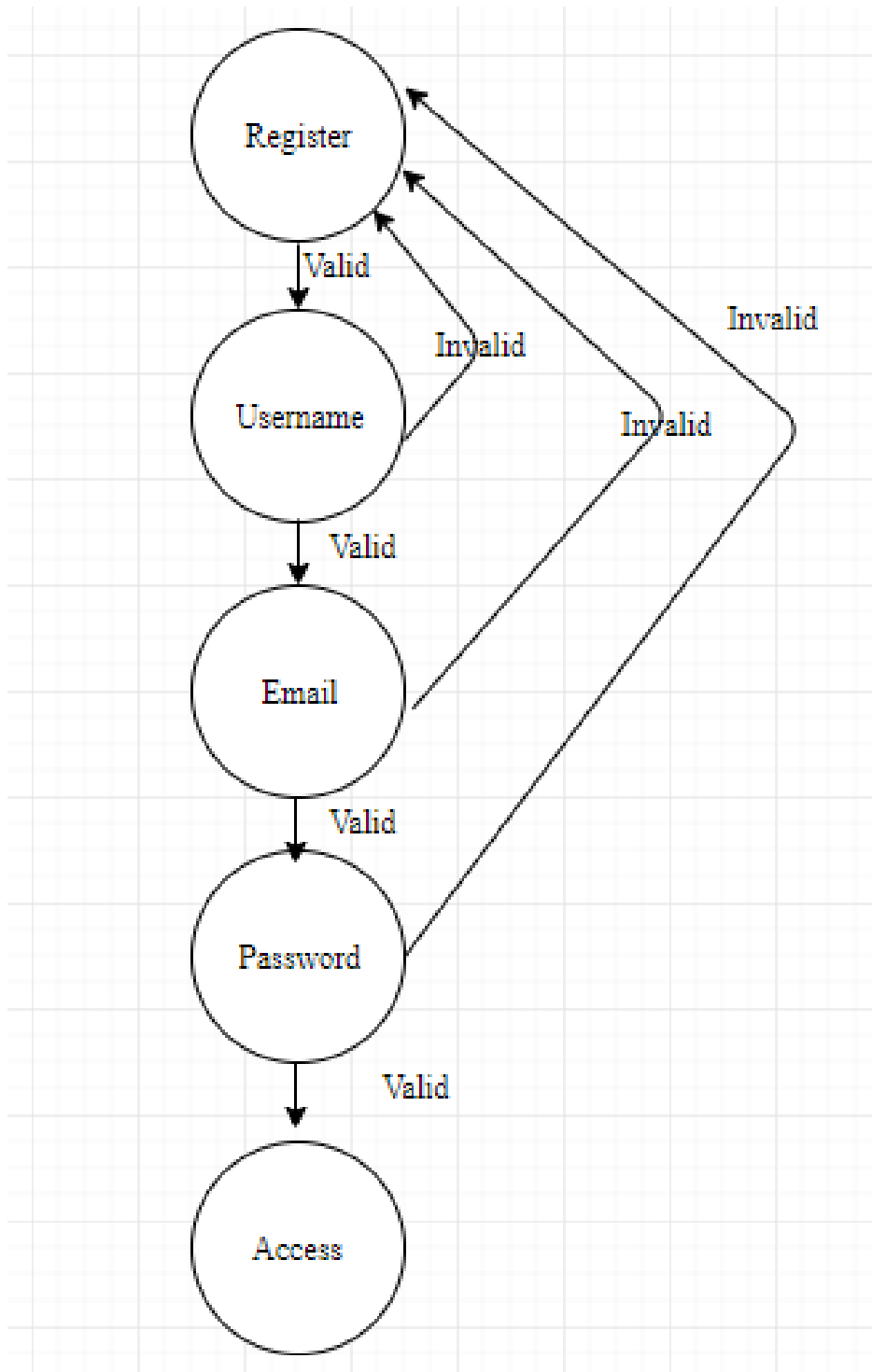


Figure 35 Data Flow Testing Login

**Register***Figure 36 Data Flow Testing Register*

## 6.5. Unit testing

Unit testing is a white-box testing technique. The main consideration in this test is verification of all modules of the software system. Each module is unit tested, as follows:

- Unit testing for the module User Management is tested by valid user or invalid user.
- Unit testing for the module User Account management is provided by testing Login, Logout and creating new or deleting updating existing user.
- Unit testing for the module Theme Management is performed by entering new theme information.
- Unit testing for the module Detail of courses/lectures etc. give detail accurately.
- Unit testing for the module of Notification System is performed by sending notifications to the students in registered course.
- Unit testing for the module Add/update/delete course, lectures and students' detail.

### 6.5.1 UNIT TESTING

**TEST CASES FOR UNIT TESTING OF MODULES**

| Test Case No | Identifier | Test item    | Input Specifications                 | Output Specifications                             | Pass / Fail | Description                                       |
|--------------|------------|--------------|--------------------------------------|---------------------------------------------------|-------------|---------------------------------------------------|
| 1            | UTS-LOG-01 | Login        | Email, Password                      | Error message, Welcome Message                    | Pass        | To verify login mechanism and user authentication |
| 2            | UTS-UR-01  | Registration | Name, Password, Email, DOB, Phone_No | Validation Error, Successful Registration Message | Pass        | To Register a New User                            |

|   |           |               |                                         |                                         |      |                       |
|---|-----------|---------------|-----------------------------------------|-----------------------------------------|------|-----------------------|
| 3 | UTS-AC-01 | Add Course    | Course Category, Course Name            | Validation Error, Successful Add Course | Pass | To Add a new Course   |
| 4 | UTS-AL-01 | Add Lecture   | Course Name, Lecture Name, Lecture File | Validation Error, Lecture added         | Pass | To add a Lecture      |
| 5 | UTS-AV-01 | Add Video     | Course Name, Video Name, Video File     | Validation Error, Video added           | Pass | To add a Video        |
| 6 | UTS-VL-01 | View Lectures | File Name, File Size, File Type,        | Validation Error, Lecture Viewed        | Pass | To View Lectures      |
| 7 | UTS-VV-01 | View Videos   | Video Name, Video Size, Video Type      | Validation Error, Video Viewed          | Pass | To View Videos        |
| 8 | UTS-SN-01 | Notification  | Notification Message                    | Validation Error, Notification Sent     | Pass | To send notifications |
| 9 | UTS-UL-01 | Logout        | Must login                              | Validation Person                       | Pass | Logout                |

Table 45 Test Cases for Unit Testing

**TEST CASES FOR BLACK BOX TESTING OF DOCUMENTATION / TEXT**

| Test Case No | Identifier | Test item                                     | Input Specifications                    | Output Specifications                  | Pass / Fail | Description                                                    |
|--------------|------------|-----------------------------------------------|-----------------------------------------|----------------------------------------|-------------|----------------------------------------------------------------|
| 1            | BBT-TXT-01 | Syntax Grammatical Correctness Of the Content | Text written on each page of the system | Grammatically correct content or error | Pass        | To check the content for grammatical & syntactical correctness |
| 2            | BBT-TXT-   | Semantics                                     | Text written on                         | Semantics correctness                  | Pass        | To check the text for the use of improper                      |



|   |            |                                 |                                                                             |                                 |      |                                                                         |
|---|------------|---------------------------------|-----------------------------------------------------------------------------|---------------------------------|------|-------------------------------------------------------------------------|
|   | 02         |                                 | each page of the system                                                     | or Bugs / errors in text        |      | phrases terminologies and semantic correctness                          |
| 3 | BBT-TXT-03 | Validity of contact information | All contact information provided in Contact Us or other pages of the system | Correct / incorrect information | Pass | To check all the contact information for validity                       |
| 4 | BBT-TXT-04 | Text in title bar               | Text in browser's title bar in each page                                    | Bugs / errors in text           | Pass | To check the text in title bar of each page for relevance & correctness |

Table 46 Test Cases for Black Box Testing of Document

### TEST CASES FOR BLACK BOX TESTING OF HYPERLINKS

| Test Case No | Identifier | Test item               | Input Specifications                                           | Output Specifications                | Pass / Fail | Description            |
|--------------|------------|-------------------------|----------------------------------------------------------------|--------------------------------------|-------------|------------------------|
| 5            | BBT-HPL-01 | Hyperlinks connectivity | All text and image hyperlinks and their sub-links on each page | Working<br>Broken<br>Incorrect Links | Pass        | To check all the links |

Table 47 Test Cases for Black Box Testing of Hyperlinks

### TEST CASES FOR BLACK BOX TESTING OF GRAPHICS

| Test Case No | Identifier | Test item      | Input Specifications      | Output Specifications                                | Pass / Fail | Description                                                 |
|--------------|------------|----------------|---------------------------|------------------------------------------------------|-------------|-------------------------------------------------------------|
| 7            | BBT-GRA-01 | Load & Display | All graphics on each page | Displayed properly/bug s, errors or delay in loading | Pass        | To check that all the graphics display and load properly    |
| 8            | BBT-GRA-02 | Balance        | All graphics on each page | All images placed properly/ bugs errors              | Pass        | To check that all the graphics for their size and placement |

Table 48 Test Cases for Black Box Testing of Graphics

### TEST CASES FOR BLACK BOX TESTING OF FORMS

| Test Case No | Identifier | Test item       | Input Specifications                       | Output Specifications                    | Pass / Fail | Description                                                |
|--------------|------------|-----------------|--------------------------------------------|------------------------------------------|-------------|------------------------------------------------------------|
| 9            | BBT-FRM-01 | Text Boxes      | Text and numeric inputs in all text fields | Validation checks applied / no bug found | Pass        | To check that all the fields store only correct input data |
| 10           | BBT-FRM-01 | Drop down lists | Input test data according to the field     | -do-                                     | Pass        | -do-                                                       |
| 11           | BBT-FRM-01 | Buttons         | Click, double click                        | -do-                                     | Pass        | -do-                                                       |

*Table 49 Test Cases for Black Box Testing of Forms*

## 6.5.2 USABILITY TESTING

### TEST CASES

| Test Case No | Identifier | Feature to be Tested                            | Expected Result                                        | Actual Result | Pass / Fail | Description                                                                                       |
|--------------|------------|-------------------------------------------------|--------------------------------------------------------|---------------|-------------|---------------------------------------------------------------------------------------------------|
| 1            | UST-GRA-01 | Aesthetic Appropriate tenses                    | All graphics must present a consistent and good design | Ok            | Pass        | To check the color scheme, balance emphasis and consistency in all graphical elements of the site |
| 2            | UST-GRA-02 | Scrolling text, marquees and constantly running | Moving images and text should not be used increasingly | -do-          | Pass        | To check the moving elements specially text is readable                                           |
| 3            | UST-GRA-03 | Outdated information                            | Content provided must be up to date                    | -do-          | Pass        | To check the content specially news for outdated information                                      |
| 4            | UST-GRA-04 | Overly long download time                       | Pages must load instantaneously and properly           | -do-          | Pass        | To check the loading time of all the web pages                                                    |

*Table 50 Usability Testing*

### 6.5.3 COMPATIBILITY TESTING

#### TEST CASES

| Test Case No | Identifier | Browser Software & version | Display & Load Properly | Different text sizes supported Yes/No | Downloading of web pages Supported/ Not supported |
|--------------|------------|----------------------------|-------------------------|---------------------------------------|---------------------------------------------------|
| 1            | CMT-BRW-01 | Internet explorer          | Yes                     | Yes                                   | Supported                                         |
| 2            | CMT-BRW-02 | Mozilla Firefox            | Yes                     | Yes                                   | Supported                                         |
| 3            | CMT-BRW-03 | Google Chrome              | Yes                     | Yes                                   | Supported                                         |

*Table 51 Compatibility Testing*

### 6.6. Integration testing

Integration testing is the technique for testing the interfaces of software components. Each software components in unit tested, and all the components are integrated to perform together. The tests are conducted to ensure that the components are working properly after interfacing.

All modules are integrated by an incremental approach, and integration testing of the system is performed as follows:

- Integrate login module and test that the software is properly connected to the database.
- Integrate all modules and test by inserting updating, and deleting records.

#### 6.6.1 SYSTEM TESTING

System testing involves the set of tests that ensures that the entire system performs according to specifications.

#### 6.6.2 SECURITY TESTING

Security testing is protection testing that test security methods in the system to avoid invalid intrusions. Several security checks have been included in the system. The system requires a logon ID and password to initiate. For allowing only authenticated users to log-in.

## **6.7. Performance testing**

Performance testing uses criteria to check that the system functions according to the specifications. All unit modules have been tested successfully. The integration of these unit modules produces reports in the required format.

## **6.8. Stress Testing**

A system evaluated under normal conditions in all previous testing techniques, and no conditions where system can fail are tested. Stress testing evaluates the system under abnormal conditions. The system is tested for various costs and number to check accuracy and to ensure that the system performs accurately under all input conditions.

# Chapter 7

## **Summary, Conclusion and Future Enhancements**

## **Chapter 7: Summary, Conclusion & Future Enhancements**

### **7.1. Project Summary**

This site helps the students to effectively register and login and then add courses according to the given categories and of their need that will be useful for other students and themselves. This framework encourages the students to study more efficiently and more focused by allowing them to join registered courses and upload their own lectures according to those courses and also by uploading different related material according to the course in which they have registered. This framework enables the students to add videos or video lectures in the joined course and also allowing them to chat with everyone in the registered course. Students in the registered course are also allowed to send notification openly so that everyone in the group can keep themselves up to date by the current activities that have been going on. The framework additionally enables users to register themselves as tutors so that they can be hired by any student for tutoring them.

### **7.2. Achievements and Improvements**

We have created this platform for the betterment of students and the things we improved were that after the implementation of lecture system we thought that we should add a search option because a student will find it easy with search option to view a specific lecture in the list of thousands of lectures. During the working on this project we got an idea of notification system that a student can send a notification to every student in the particular class so that it will be helpful for everyone to stay up-to-date with the current situation.

### **7.3. Lessons Learnt**

Stay aware of patterns and advancements in online education system. Subscribe to trustworthy platforms and also watch out for the opposition and receive new innovation. Create quality system and keep it reliable. Improve user benefit, build up a decent user relationship administration. In spite of the fact that different platforms may interfere to

criticize you so remind them at each progression that there is a human on the opposite end. Focus on keep advancing and upgrading the system for better response get its name out and grow a piece of overall system.

#### **7.4. Future Enhancements/Recommendations**

The project here is simply to guarantee that the students will be able to study more effectively and efficiently in today's difficult testing world. As of now, framework works for a predetermined number of users to work. In not so distant future it will reach out for some sorts of protection strategies with the goal that proficiency can be made strides.

# Appendices



## Appendix A: Information / Promotional Material

We have created a platform which will help all the student in their study. We have promoted our project through digital marketing.

### A.1. Broacher (Bi-Fold)

#### Inside

**EXECUTIVE SUMMARY**

We are developing a source that works on Website. This source totally depends on facilitating students in Study and other Educational Material. In this project we provide students with a specific source so that he/she can gather their study material from a single source and don't have to use other sources like Google Drive, Google Classroom and Dropbox etc. It provides feature in which every student of a particular class can communicate with each other in a single chat room. It provides students with extra materials of their course like video tutorials, lectures and books etc. A Particular student can provide his/her own lectures if he wants too and he also can help other students in study. A main benefit for this student is that anybody can hire him/her as a tutor. The main feature is to notify every student on their mobile numbers other than email whenever someone uploads the data.

**GOALS & OBJECTIVE**

**GOAL**

Our main goal is to provide each and every student with the help they need in their study including the lectures, video tutorials, PPT slides, complete courses etc.

**TIME & ENERGY EFFICIENCY**

It is time saving and energy efficient. Students without any physical effort can login to their accounts and can have access to any type of study material they need. They don't have to visit someone else for the lectures.

**ACCESSABILITY**

Students can have access to their lectures, tutorials and many other things while sitting anywhere.

**SCOPE**

The scope of our project is that students can manage their study material easily and they can have chatrooms available for each course to communicate with other students. They can also find most accurate and relevant data. A Particular Student can upload their own lectures and will also be able to help other students and the benefit they get is that they can be hired by anyone as a tutor.

This project has well accurately and perfectly designed database so that no data can be misplaced or lost. It has well designed front end that will attract the viewers and looks cool. It is a friendly user interface.

**WHY US ?**

- ✓ The best designed user interface with well organized database.
- ✓ Student world is running 24/7 with full backup power.
- ✓ Student world provide their users the best security and reliability so that their personal data cannot be used somewhere else.
- ✓ Every student can also choose their own tutor from the registered tutors.

## Outside



### About Us :

Student World is a platform on which students and tutors can register themselves and find it useful. Students can create their own classrooms and related to the courses and then further add lecture files, video tutorials and create chatrooms according to the course. Other registered students can join those classrooms. Tutors can be selected by the students for teaching.

### Contact Us :

Address: 6km Raiwind Rd, OPF Society, Lahore  
Phone: +92 3152333543  
Email: [info@studentworld.com](mailto:info@studentworld.com)  
Web: [www.studentworld.com](http://www.studentworld.com)



**LEARNING, EXPERIENCE  
AND CONFIDENCE**

WHERE **LEARNING** BEGINS



## A.2 Visiting Card

### Front



### Back



### A.3 Standee

STUDENT WORLD



Build Your Career



Register Now and start your educational career.



<https://www.studentworld.com>



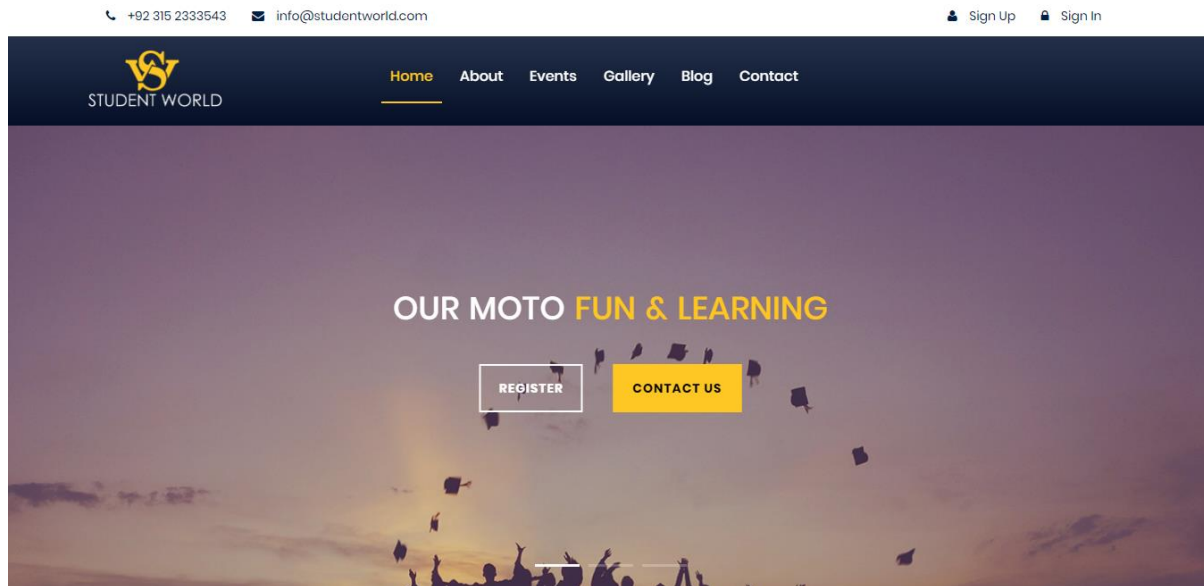
6km Raiwind Rd, OPF Society,  
Lahore



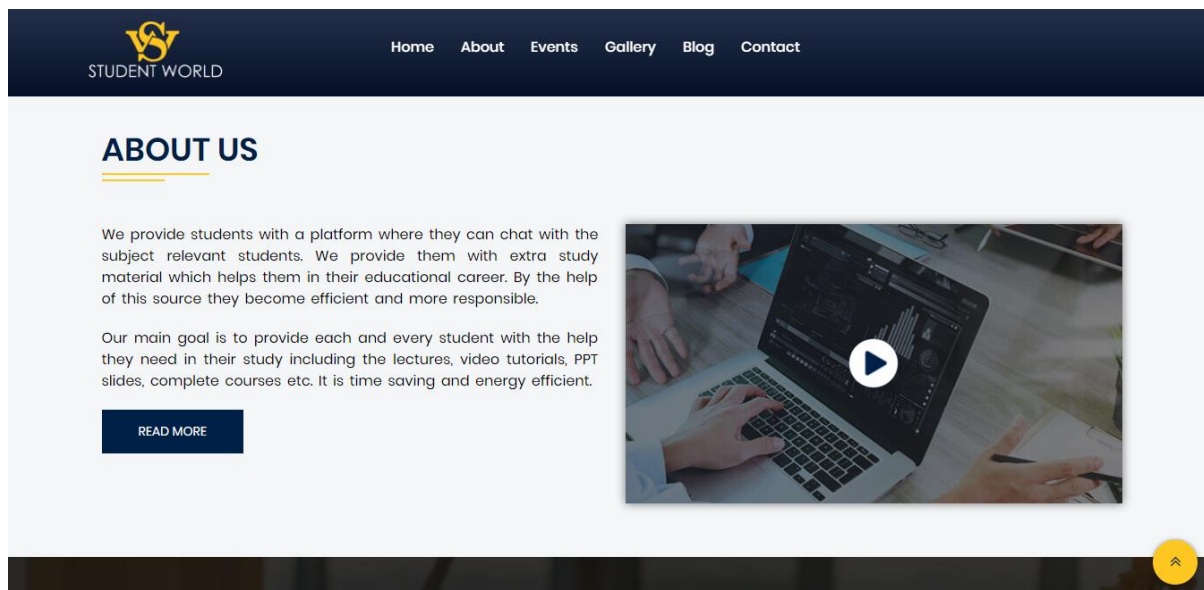
+92 315 2333543

## Website Screen Shots

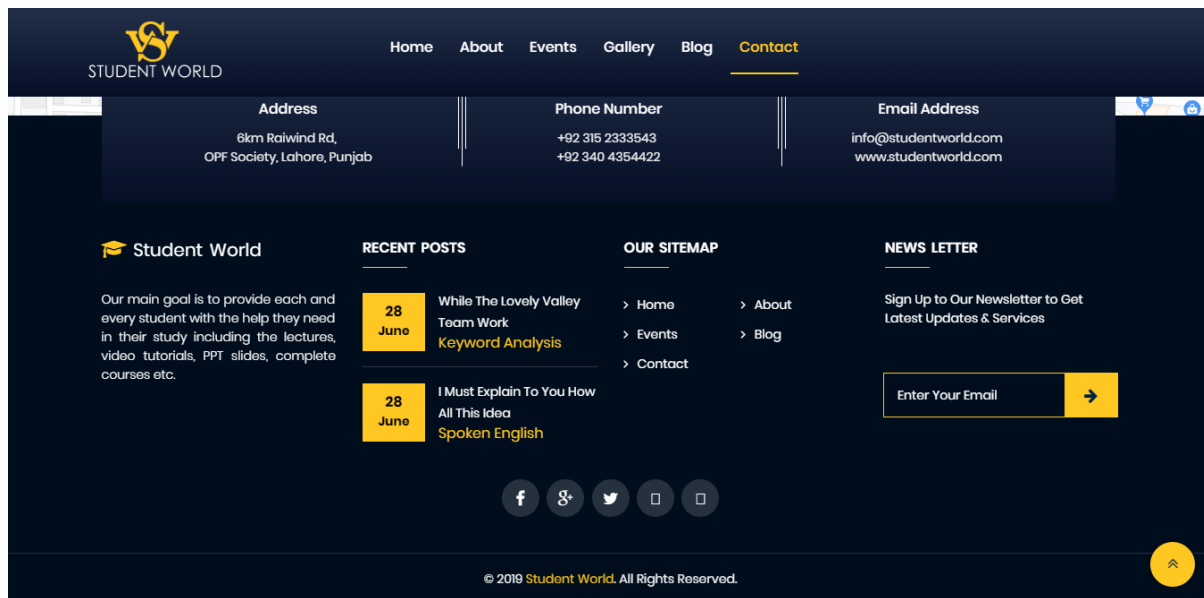
### Home1



### Home2



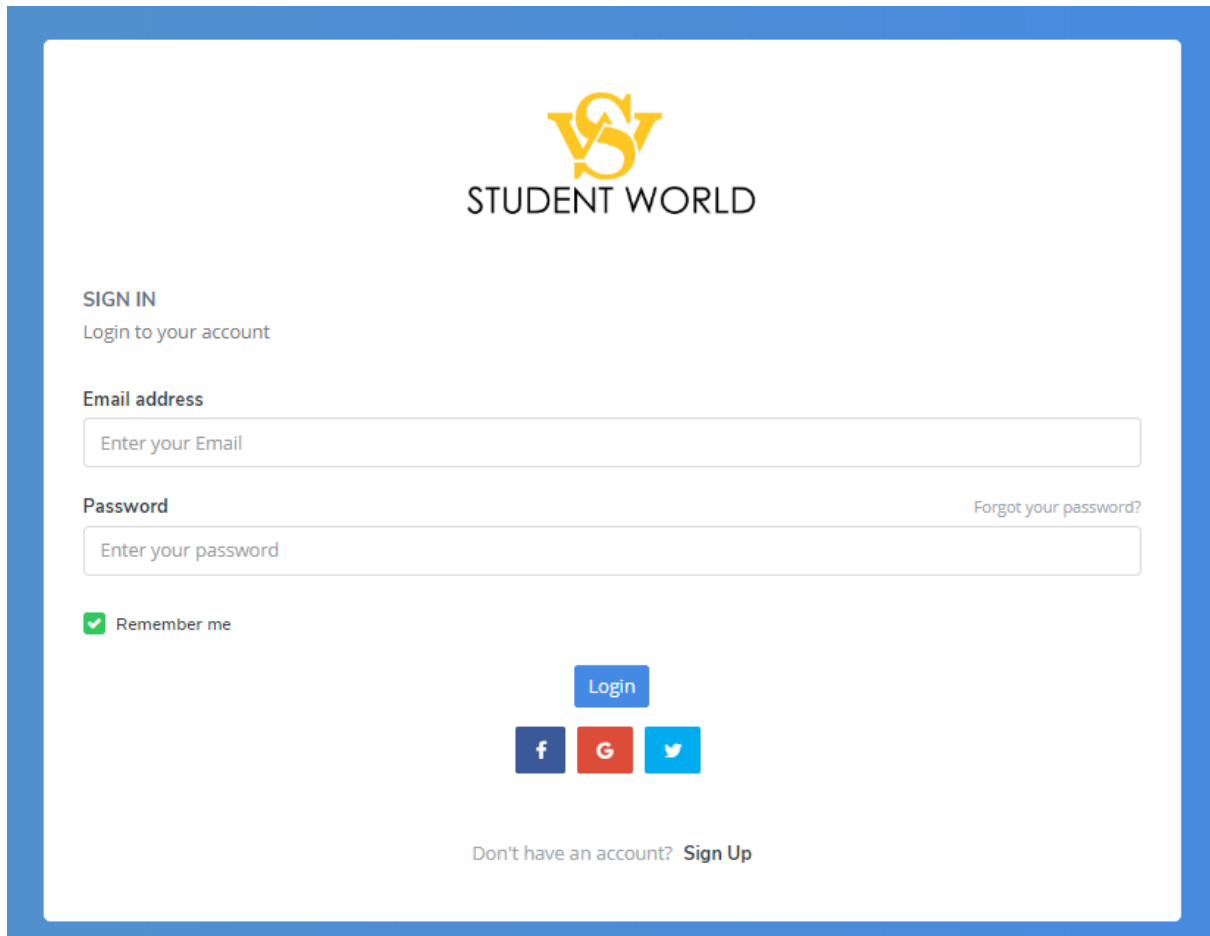
## Home3



## Sign Up

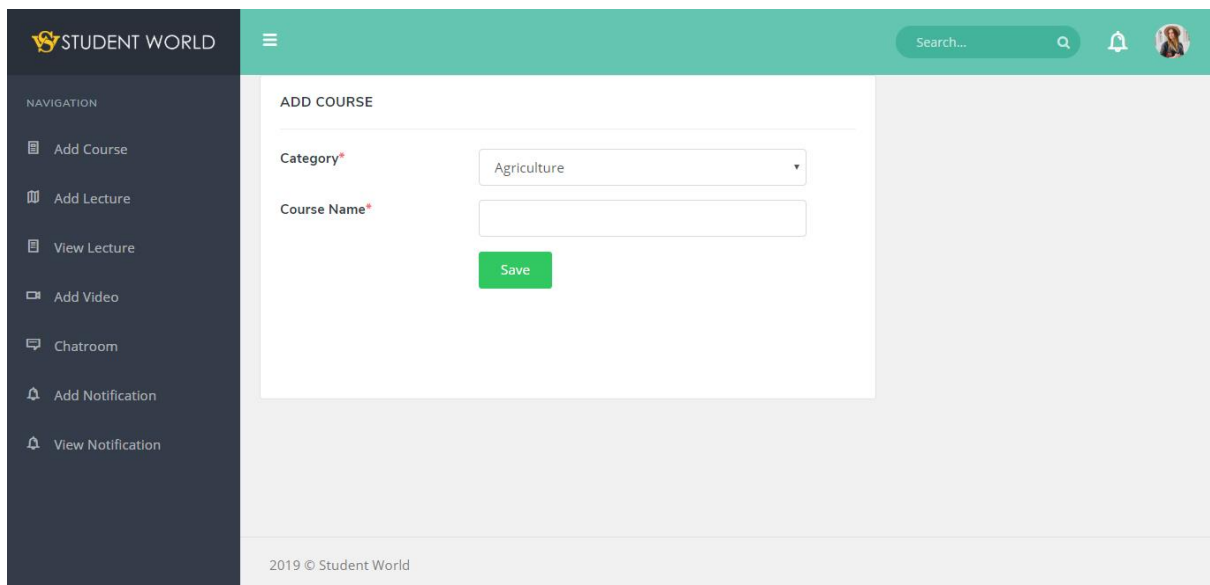


## Log In



The login page features the Student World logo at the top center. Below the logo, the text "SIGN IN" is followed by "Login to your account". There are two input fields: "Email address" with the placeholder "Enter your Email" and "Password" with the placeholder "Enter your password". A link "Forgot your password?" is located to the right of the password field. A checkbox labeled "Remember me" is checked. Below the inputs is a blue "Login" button. Underneath the button are three social media icons for Facebook, Google+, and Twitter. At the bottom, a link "Don't have an account? Sign Up" is displayed.

## Add Course



The "Add Course" page is part of the Student World application. It has a dark sidebar on the left with a "NAVIGATION" menu containing links for "Add Course", "Add Lecture", "View Lecture", "Add Video", "Chatroom", "Add Notification", and "View Notification". The main content area has a teal header with the Student World logo, a search bar, and a user profile icon. Below the header, the "ADD COURSE" form is visible. It includes a "Category" dropdown menu currently set to "Agriculture", a "Course Name" text input field, and a green "Save" button. The footer of the page shows the copyright "2019 © Student World".

## Add Lecture

The screenshot shows the 'Add Lecture' form in the Student World application. The form is titled 'LECTURE DETAILS' and contains three input fields: 'Lecture Name\*', 'File\*', and 'Course Name\*'. The 'File\*' field has a 'Choose file' button and the text 'No file chosen'. The 'Course Name\*' field is a dropdown menu with 'Computer' selected. A green 'Save' button is at the bottom of the form. The left sidebar shows navigation options: Add Course, Add Lecture, View Lecture, Add Video, Chatroom, Add Notification, and View Notification. The top header includes the Student World logo, a search bar, and a user profile icon. The footer shows '2019 © Student World'.

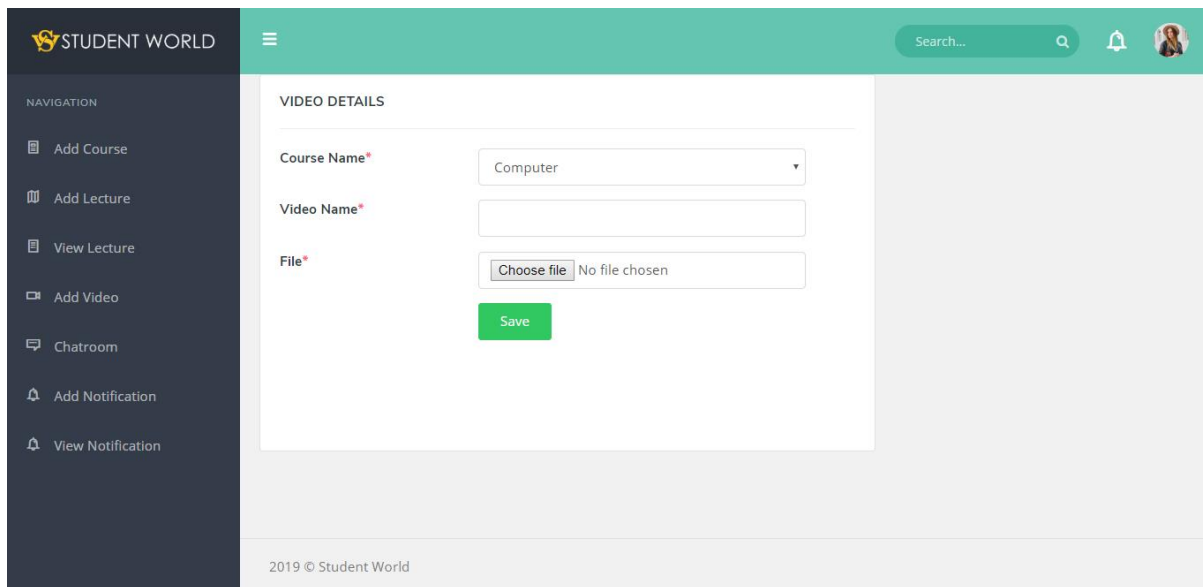
## View Lecture

The screenshot shows the 'View Lecture' page in the Student World application. The page is titled 'LECTURE FILES' and includes a search bar with the text 'SEARCH LECTURE FILES!'. Below the search bar is a table with three columns: 'Lecture Name', 'Course Name', and 'File'. The table contains five rows of data. The left sidebar shows navigation options: Add Course, Add Lecture, View Lecture, Add Video, View Videos, Chatroom, Add Notification, and View Notification. The top header includes the Student World logo, a search bar, and a user profile icon. The footer shows '2019 © Student World'.

| Lecture Name | Course Name | File                            |
|--------------|-------------|---------------------------------|
| Invoice      | Computer    | ~/Files/invoice.pdf             |
| SQL FILE     | Computer    | ~/Files/neoschool.sql           |
| Invoice      | Computer    | ~/Files/Furqan Malik.docx       |
| Invoice      | Computer    | ~/Files/1.jpg                   |
| Image        | Computer    | ~/Files/IMG-20190421-WA0008.jpg |



## Add Video

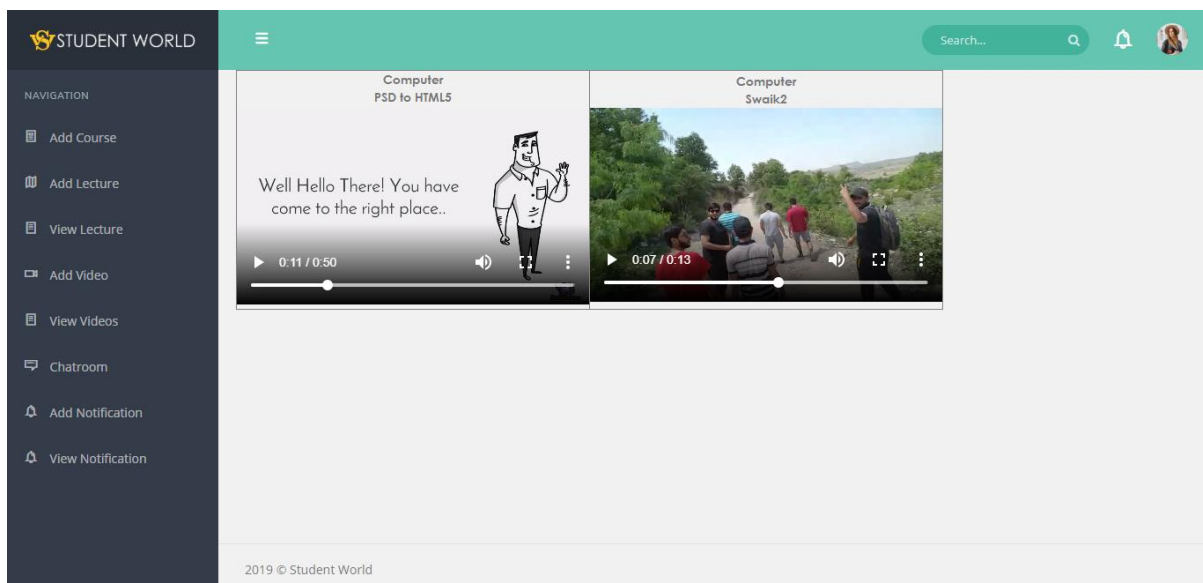


The screenshot shows the 'Add Video' form in the Student World application. The form is titled 'VIDEO DETAILS' and is located in the main content area. It has a dark sidebar on the left with navigation options: Add Course, Add Lecture, View Lecture, Add Video, Chatroom, Add Notification, and View Notification. The top header is teal with a search bar, a bell icon, and a user profile icon. The form itself has a white background and contains the following fields:

- Course Name\***: A dropdown menu with 'Computer' selected.
- Video Name\***: A text input field.
- File\***: A file upload button labeled 'Choose file' and a status 'No file chosen'.
- Save**: A green button to submit the form.

At the bottom of the page, there is a footer that reads '2019 © Student World'.

## View Video



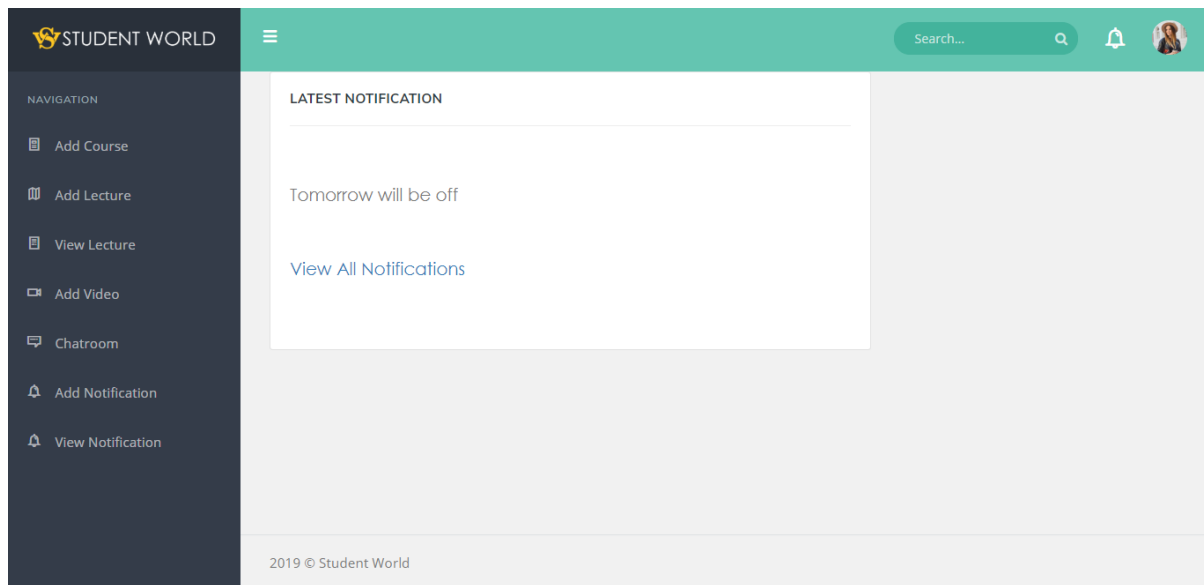
## Chatroom

The screenshot shows the 'Chatroom' page of the 'STUDENT WORLD' application. The interface features a dark blue sidebar on the left with a 'NAVIGATION' menu containing options: Add Subject, Add Lecture, View Lecture, Add Video, Chatroom (selected), Add Notification, and View Notification. The main content area has a teal header with a search bar, a notification bell, and a user profile icon. Below the header, the 'GROUP CHAT' section includes a 'Type Message' input field, a green 'Send' button, and a 'ChatBox' placeholder. The footer displays '2019 © Student World'.

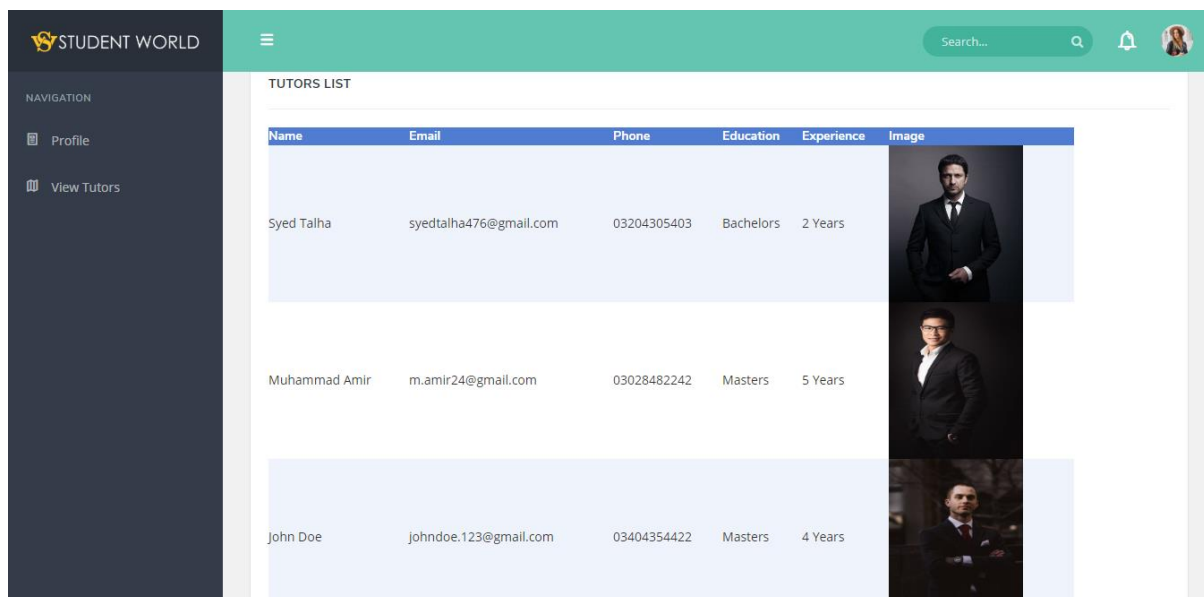
## Add Notification

The screenshot shows the 'Add Notification' page of the 'STUDENT WORLD' application. The sidebar is identical to the previous page, with 'Add Notification' selected in the 'NAVIGATION' menu. The main content area has a teal header with the same search bar, notification bell, and user profile icon. Below the header, the 'SEND NOTIFICATIONS' section includes a 'Your Message' input field, a green 'Send' button, and a placeholder for the notification content. The footer displays '2019 © Student World'.

## View Notification



## View Tutors



### A.4 First Level heading [16 pt, Calibri, Bold, Left aligned]

The first heading's height is 15%, font size is 40px, and it is bold, aligned center. The text color is white. The background color is CMYK. Font family is century gothic. Two more headings are used that are also bold, aligned center, color is same to the first heading font family is Century Gothic height is 14%.

**A.4.1 Second level heading [14 pt, Calibri, Bold, Left aligned]**

The second level heading has the Visiting Card which has 317px height, font size is 40px, and it is bold, aligned center. The text color is black. The background color is CMYK. Font family is Century Gothic. Second heading in Card is on the back. Font size is 40px, color black, and font family is Century Gothic.

**A.4.1.1 Third level heading [12 pt, Calibri, Bold, Left aligned]**

In third level heading standee height is 17392px we kept, it is bold, font family is Century Gothic the font size is 15%, text aligned in center, text color is black, background color is CMYK.

# Reference and Bibliography

## Reference and Bibliography

- [1] M. Sher, M. Rehman, "*Title of the Paper*" Conference name/Journal Name, Edition, Volume, Issue, ISBN/ISSN, PP, Publisher/City-Country, Year.