

**Department of Computer Science &**

**Information Technology**

**University of Sargodha Lyallpur campus Faisalabad**

**“Educational Information and Recommendation System”**

1 Chapter 1 INTRODUCTION 4

1.1) Introduction 4

1.2) Project Scope 5

1.3) Existing System 5

1.4) Identifying Entities 5

1.5) Specific Object of the System 5

Admin 5

Institute 6

Student 6

1.6) Goals and Objectives 6

1.7) Problem Statement 6

1.8) Gantt Chart 6

1.9) Product Features 7

2 CHAPTER 2 REQUIREMENT 8

2.1) Functional Requirements 8

2.2) Non-Functional Requirements 8

2.2.1) **Usability** 8

2.2.2) **Efficiency** 8

2.2.3) **Response Time** 8

2.2.4) **Error Handling** 8

2.3) Technical Requirements 8

Tools 9

Development Tools 9

Documentation Tools 9

Technology 9

Tools and Technologies in Detail 9

Tools 9

XAMPP 9

MySQL 10

Php Storm 10

MS Project 10

MS Visio 10

Technology. 10

HTML5 11

CSS 11

Bootstrap 11

JavaScript 11

Ajax 11

JQuery 12

Php 12

2.4) User Requirements 12

2.5) Capture “Shall” Requirements 12

2.6) Allocate Requirements 13

2.7) System Use Case Diagram 14

2.8) Admin Use Case Diagram 16

2.8.1) Admin Login Use Case Diagram 16

2.8.2) Admin Change Password 17

2.8.3) Admin Add Information 18

2.8.4) Admin Manage Information 19

2.8.5) Admin Manage Registrations 20

2.8.6) Admin Manage Users 21

2.9) Student Use Case Diagrams 22

2.9.1) Student View Information 22

2.9.2) Student Create Account 23

2.9.3) Student Login 24

2.9.4) Student Change/Update Password 25

2.9.5) Student Update Profile 26

2.9.6) Student Get Guidelines 27

2.10) Institute Use Case Diagrams 28

2.10.1) Institute Create Account 28

2.10.2) Institute Login 29

2.10.3) Institute Change/Update Password 30

2.10.4) Institute Add Information 31

2.10.5) Institute Update Information 32

2.11) Description of Use Cases 33

2.11.1) Admin Use Case 33

2.11.1.1) Admin Login 33

2.11.1.2) Admin Change/Update Password 34

2.11.1.3) Admin Add Information 35

2.11.1.4) Admin Manage Information 36

2.11.1.5) Admin Manage Registrations 37

2.11.1.6) Admin Manage Users 38

2.11.2) Student Use Case 39

2.11.2.1) Student View Information 39

2.11.2.2) Student Registration 39

2.11.2.3) Student Login 40

2.11.2.4) Student Change/Update Password 42

2.11.2.5) Student Update Profile 42

**2.11.3) Institute Use Case** 43

2.11.3.1) Institute Registration 43

2.11.3.2) Institute Login 44

2.11.3.3) Institute Change/Update Password 45

2.11.3.4) Institute Add Information 47

2.11.3.5) Institute Update Information 47

CHAPTER 3: DESIGNING 48

3.1) System Collaboration Diagram 49

3.1.1) Admin Collaboration Diagram 49

3.1.2) Student Collaboration Diagram 50

3.1.3) Institute Collaboration Diagram 51

3.2) Activity Diagram 52

3.2.1) Admin Activity Diagram 52

3.2.2) Student Activity Diagram 53

3.2.3) Institute Activity Diagram 54

3.3) Data Flow Diagram 55

3.3.1) Context Level Diagram 55

3.3.2) Admin Side Data Flow Diagram - Level 1 56

3.3.3) Student Side Data Flow Diagram - Level 1 57

3.3.4) Institute Side Data Flow Diagram - Level 1 57

CHAPTER 4: DELIVERABLE (User Interface Design) 58

4.1) Introduction 58

4.2) Interface Detail 58

# Chapter 1 INTRODUCTION

## Introduction

Educational Information and Recommendation System is web-based system. The purpose of this system is to guide the students in choosing a proper career. When students complete their intermediate, they are in search of the institutes which are better for them. At that time, they need a proper guideline about different institutes, programs and their ratings. This system is that platform that can guide the students about the institutes regarding their merit, levels, fees, scholarships and other aspects. This system is also time saving for students as they can find all information about institutes at a single platform instead of visiting different sites. Institutes are categorized by cities, programs, public sectors and private sectors. It is a dynamic site having admin panel as well as client.

This system provides a facility to education institutes to advertise here. The educational institutes (universities and colleges) must registered first and after approval from admin, they can login with their email address and passwords and can insert information about them. Moreover, these users can view information by using the internet.

Thus, this system is very much convenient for the students as they can get information about different institutes just visiting a single platform.

## Project Scope

This system can provide all information about educational institutes (Universities & colleges) to the students that they needed for their proper career. This system provides all information about universities regarding their merit, levels, fees, scholarships and other aspects. This system categorize the institutes according to their cities, programs and public and private sectors.

This system guide the students about their previous education, interest, and aggregate of marks.

This system provides the facility to the institutes to get register and then update data and advertise about their institutes.

## Existing System

The existing system provides information only and do not guide the students according to their interest, previous education and marks. The aim of our project is to guide the students completely so he/she can choose such institute that is proper for building his/her future

## Identifying Entities

On the basis of abstract, one might identify the following identities.

* Admin
* Student
* Institute

## Specific Object of the System

### Admin

* Admin can Login.
* Admin can add new information or update existing information
* Admin can manage details by edition or deletion.
* Admin can approve or reject the new registration.
* Admin can view the information of the registered users.
* Admin can email any information to the registered users.
* Admin can read queries of the users and answer them.

### Institute

* Institutes can register.
* Institutes can advertise here.
* Institutes can add, edit or update their information.

### Student

* Students can review information.
* Students can get guidelines by giving some data.
* Students can send their messages to admin.
* Students can create an account to get updates.

## Goals and Objectives

Providing a platform to students that guide can guide them about educational institutes so that they can choose a better institute regarding their ambition. Providing all details about educational institutes at a single platform so that we can save the time of students.

## Problem Statement

When students complete their intermediate, they need information about different universities to take admission and build their career. So, there should be a platform that can give all necessary information time to students without wasting their time.

Thus, this system will provide all necessary information to students that they need. This system will also guide them according to their interest, previous education and marks.

## Product Features

Main features of this project are as follows:

* User friendly interface
* Ease of use
* Easy use of language and no specific psychological term is used so users do not feel any problem in using the system.
* 99.99 % accurate information is given.
* Fast and efficient
* Accurate result generation
* Reduce the time and effort of human resources.
* Maintain the record of each applicant that will register.

# CHAPTER 2 REQUIREMENT

## 2.1) Functional Requirements

Functional Requirements are those requirements that are used to illustrate the internal working nature of the system, the description of the system, and explanation of each sub-system. It involves information about what are the tasks performed by the system, what are the system processes, what information will be shown to the user etc.

The functional requirements for this system are,

* The system should allow the users to review any information that is available on the site.
* The system should allow the new users to register here.
* Admin can accept or reject the new registrations and other proposals.
* The system allow the organization to update their information.

## 2.2) Non-Functional Requirements

Non-functional requirements are those requirements that effect the working of the system. These requirements are need to fulfill to make sure the quality of the system.

### ­ ­­ 2.2.1) **Usability**

The system is user-friendly. The interface and working is easily understandable.

### 2.2.2) **Efficiency**

The system will work efficiently without facing any error.

### 2.2.3) **Response Time**

The system will provide good response time. It will show output within 5 seconds.

### 2.2.4) **Error Handling**

If there is any error face by the user, the user can inform the admin with a message. Thus, the error will be removed in the minimum time.

## 2.3) Technical Requirements

### Tools

#### Development Tools

* Php Storm (Text Editor)
* XAMP Server
* MySQL database

#### Documentation Tools

* Microsoft Word 2013
* Microsoft Visio 2013
* Microsoft Project

### Technology

We are using languages:

* HTML, CSS, Bootstrap, JQuery, Ajax and JavaScript for front-end.
* Php and MySQL for back-end.

## Tools and Technologies in Detail

### Tools

#### XAMPP

XAMPP is a free and open source cross-platform web server solution stack package, consisting mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perl programming languages.

• XAMPP's name is an acronym for:

• X (to be read as "cross", meaning cross platform)

• Apache HTTP Server

• MySQL

• PHP

• Perl

#### MySQL

It is the world's most popular open source database. It is a Relational Database Management System (RDBMS) - data and its relationships are stored in the form of tables that can be accessed by the use of MySQL queries in almost any format that the user wants.

• MySQL is a database system used on the web server.

• MySQL is ideal for both small and large applications.

• MySQL is very fast, reliable, and easy to use.

• MySQL is free to download and use.

• MySQL is developed, distributed, and supported by Oracle Corporation.

#### Php Storm

**Php Storm** is a proprietary cross-platform source code editor. It natively supports many programming languages and markup languages, and functions can be added by users with plugins, typically community-built and maintained under free-software licenses.

#### MS Project

It is used for making Gantt chart, task dependency table, task and member assignment table, project costing and critical path finding so we will use it for such kind of activities that are mentioned here.

#### MS Visio

We will use them for UML diagrams because these tools provide the best environment to design all sorts of diagrams like class, use case, sequence, activity and network diagrams.

### Technology.

#### HTML5

* **HTML**stands for **Hypertext Markup Language**, is the predominant markup language for web pages.
* HTML provides it means to create a structured document such as headings, forms, paragraphs, lists, links, quotes, and other so many items.
* Support Images and objects (audio and video) to be embedded and create a interactive Web-pages.

#### CSS

* + - CSS stand for “Cascading Style Sheets”.
    - **C**ascading **S**tyle **S**heets is the language for describing the presentation of Web pages, including colors, layout, and fonts.
    - It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers.

#### Bootstrap

* Bootstrap is the world's most popular framework for building responsive, mobile-first sites and applications.
* Inside you'll find high quality HTML, CSS, and JavaScript to make any project easier than ever.

#### JavaScript

* JavaScript is a cross-platform, object-oriented scripting language.
* It is a small and lightweight language.
* JavaScript contains a standard library of objects, such as Array, Date, and Math, and a core set of language elements such as operators, control structures, and statements.

#### Ajax

* Asynchronous JavaScript and XML.
* AJAX is a technique for creating fast and dynamic web pages.
* AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.
* Classic web pages, (which do not use AJAX) must reload the entire page if the content should change.

#### JQuery

* JQuery is a lightweight, "write less, do more", JavaScript library.
* The purpose of JQuery is to make it much easier to use JavaScript on your website.
* JQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.

#### Php

* PHP is a recursive acronym for "PHP: Hypertext Pre-processor".
* PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
* It is integrated with a number of popular databases, including MySQL, Oracle, Sybase, Informix, and Microsoft SQL Server.

## 2.4) User Requirements

Users need following requirements to access this system.

* Operating System
* Web Browser
* Internet Access

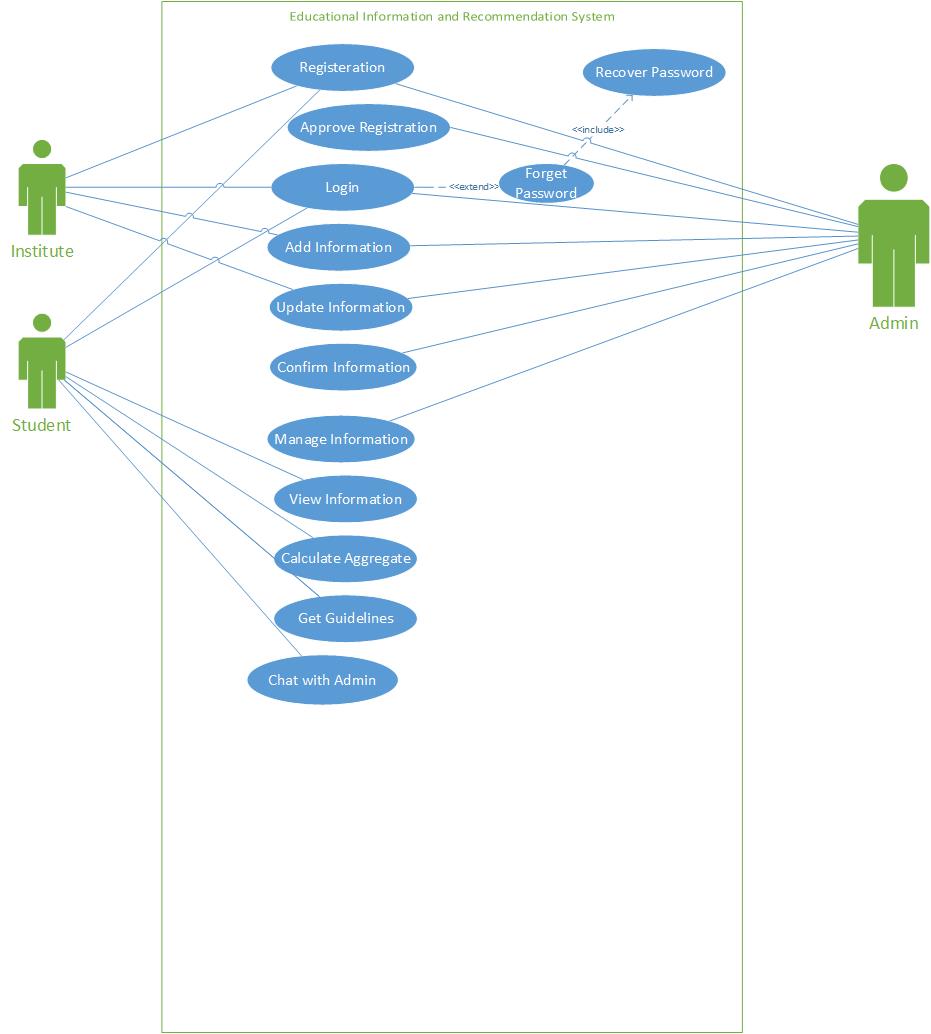
## 2.5) Capture “Shall” Requirements ­­

|  |  |
| --- | --- |
| **#** | **Initial Requirements** |
| **1** | The system will be managed by admin. |
| **2** | The system will provide the functionality to the admin to manage the information, manage the registrations and users, and reply to the user’s massages. |
| **3** | The system will provide the functionality to the students to view the information. |
| **4** | The system will also provide functionality to the students to create an account to get update notifications. |
| **5** | The system will provide the functionality to the students to get guidelines according to his given details. |
| **6** | The system will provide the functionality to the institutes to advertise here. The institutes can add and update their information. |

## 2.6) Allocate Requirements

|  |  |  |
| --- | --- | --- |
| **#** | **Initial Requirements** | **Use Case Diagram** |
| 1 | The system will be managed by admin after login. | UC\_2.8.1\_Admin\_Login |
| 2 | The admin can change password after login. | UC\_2.8.2\_Admin\_Change\_Password |
| 3 | The admin can add or update any information. | UC\_2.8.3\_Admin\_Add\_Update\_Information |
| 4 | The system facilitate the admin to manage all information. | UC\_2.8.4\_Admin\_Manage\_Information |
| 5 | The admin can approve the registrations. | UC\_2.8.5\_Admin\_Approve\_Registrations |
| 6 | The admin can manage all users. | UC\_2.8.6\_Admin\_Manage\_Users |
| 7 | The students can view the information available on the site. | UC\_2.9.1\_Student\_View\_Information |
| 8 | The students can create an account to get update notifications. | UC\_2.9.2\_Student\_Create\_Account |
| 9 | The students can change or update their password. | UC\_2.9.3\_Student\_change\_recover\_Password |
| 10 | Students can login to update their profiles and check notifications. | UC\_2.9.4\_Student\_login |
| 11 | Students can update their profiles. | UC\_2.9.5\_Student\_update\_profile |
| 12 | The students can get guidelines according to given information. | UC\_2.9.6\_Student\_Get\_Guidlines |
| 13 | Institutes can create their account. | UC\_2.10.1\_Institute\_Create\_Account |
| 14 | Institutes should login to add their details. | UC\_2.10.2\_Institute-login |
| 15 | Institutes can change and recover password. | UC\_2.10.3\_Institute\_change\_recover\_password |
| 16 | Institutes can add their information after login. | UC\_2.10.4\_Institute\_Add\_Information |
| 17 | Institutes can update their information. | UC\_2.10.5\_Institute\_Update\_Information |

## 2.7) System Use Case Diagram



## 2.8) Admin Use Case Diagram

### 2.8.1) Admin Login Use Case Diagram

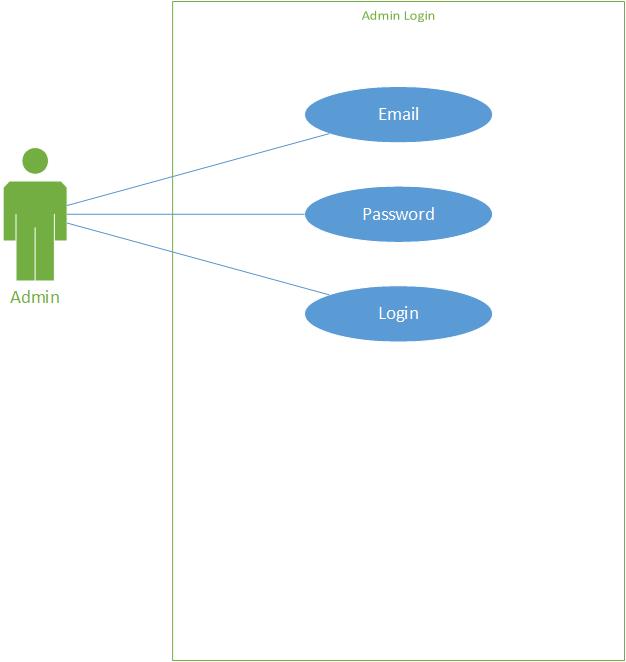
­­

Fig.2.8.1

### 2.8.2) Admin Change Password

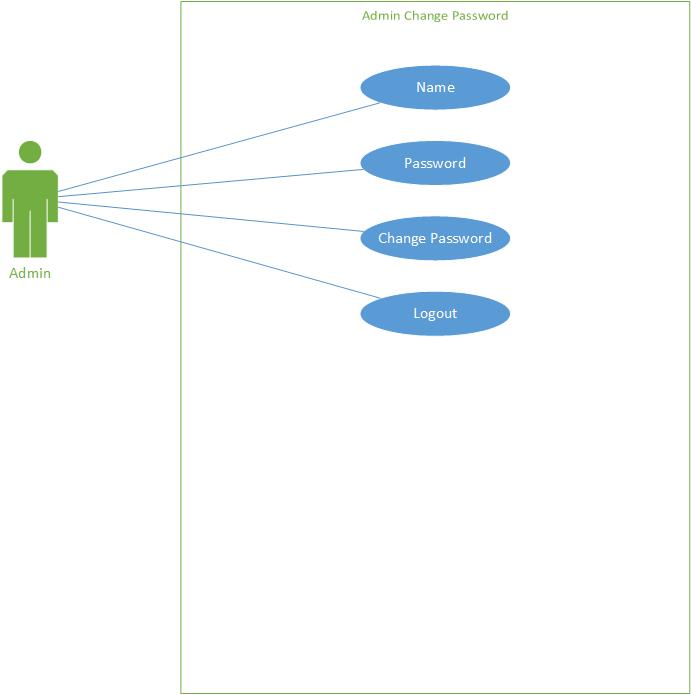
****

Fig.2.8.2

### 2.8.3) Admin Add Information

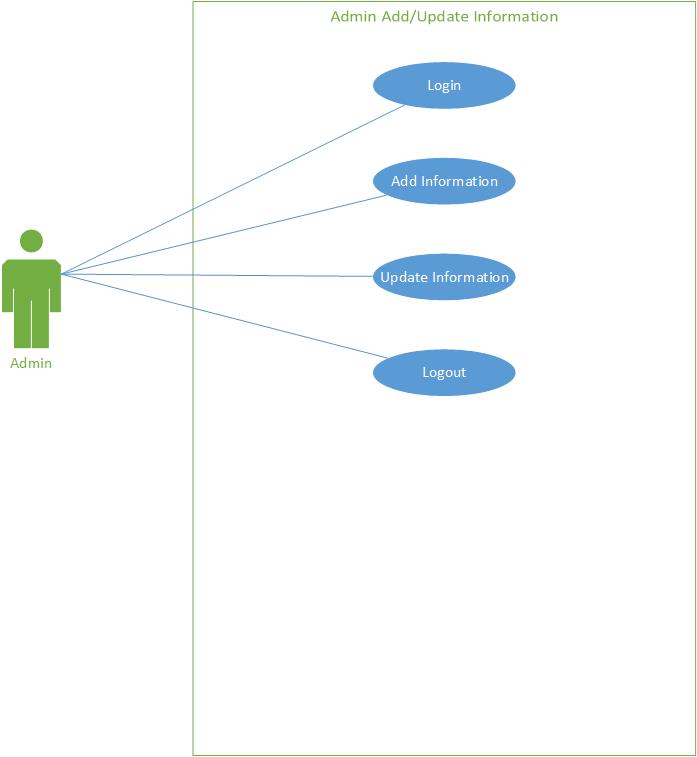
****

Fig.2.8.3

### 2.8.4) Admin Manage Information

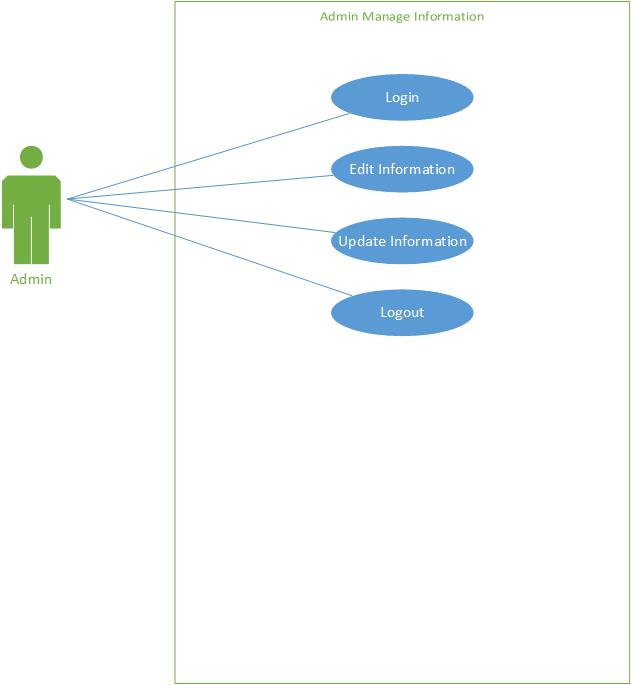
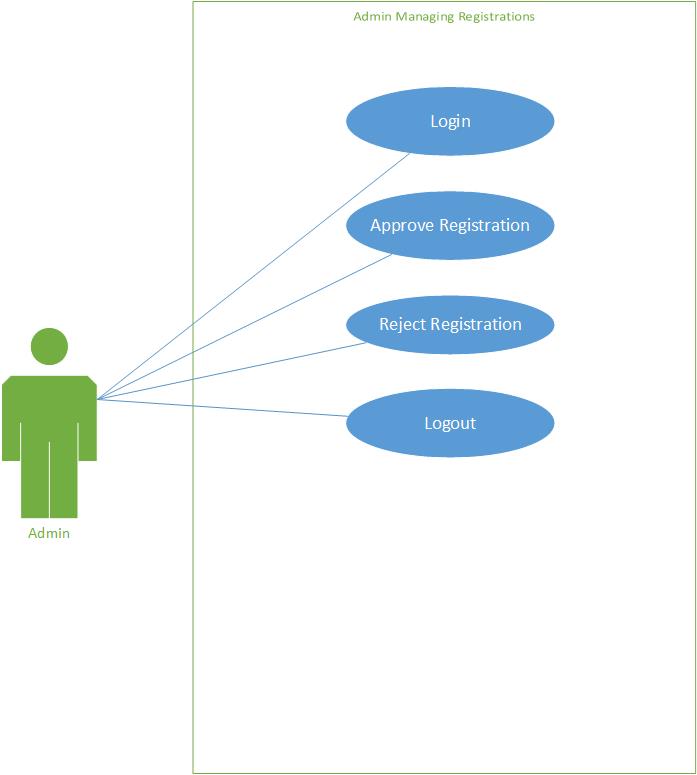
****

Fig.2.8.4

### 2.8.5) Admin Manage Registrations

****

2.8.5

### 2.8.6) Admin Manage Users

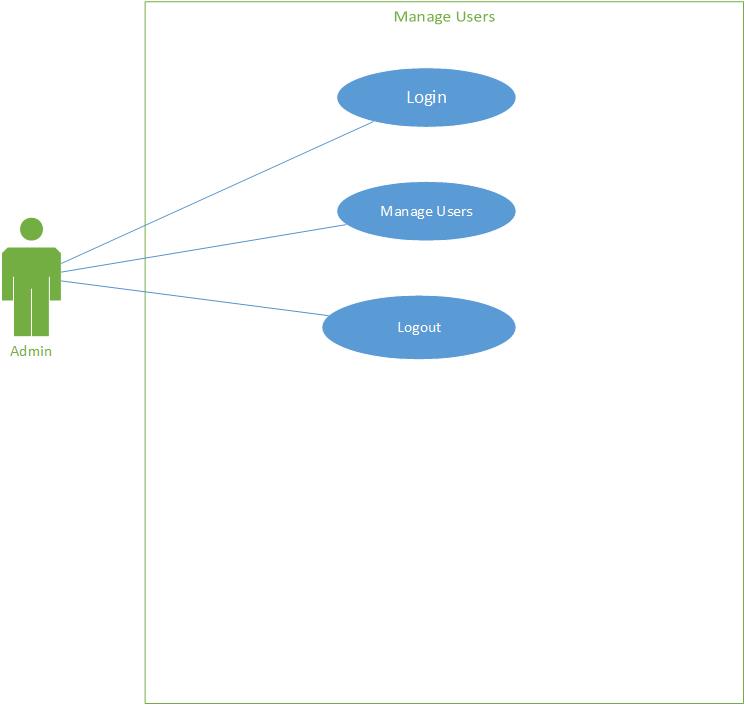
****

Fig.2.8.6

## 2.9) Student Use Case Diagrams

### 2.9.1) Student View Information

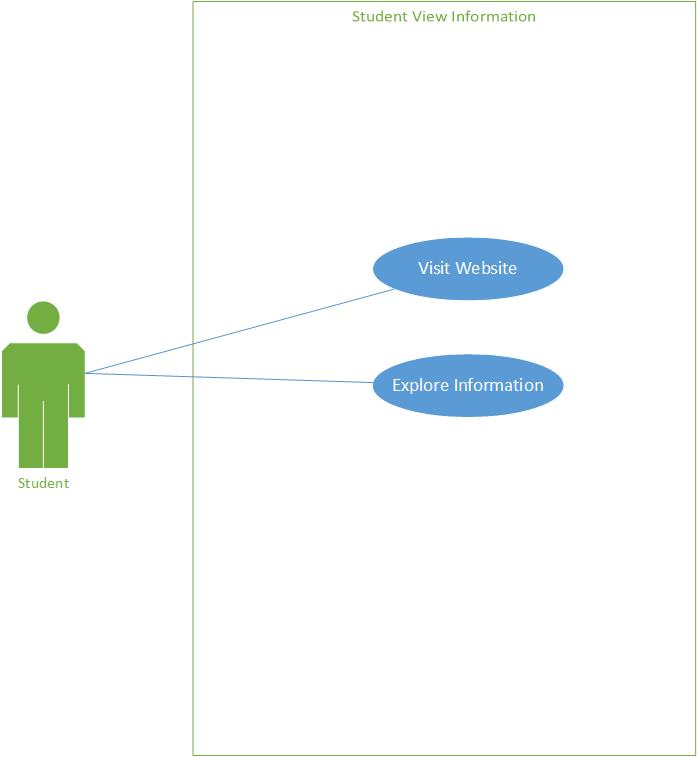
****

Fig.2.9.1

### 2.9.2) Student Create Account

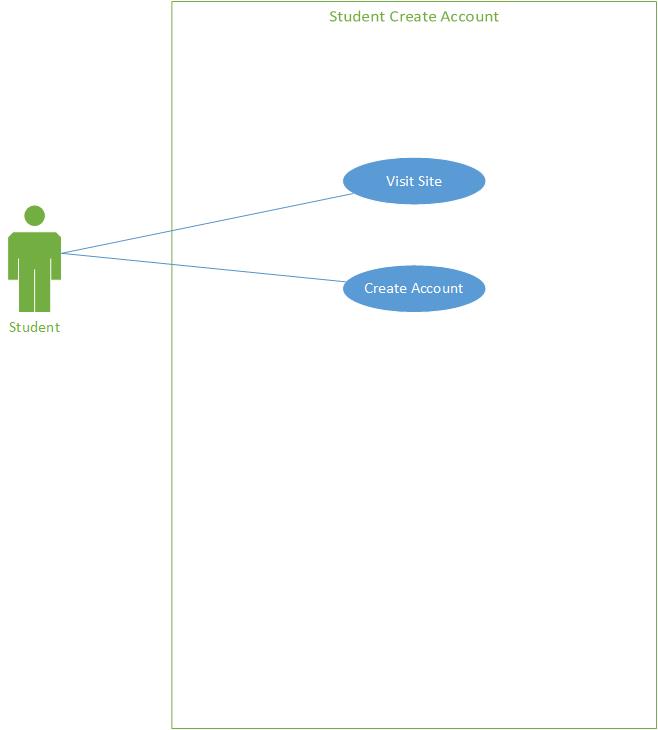
****

Fig.2.9.2

### 2.9.3) Student Login

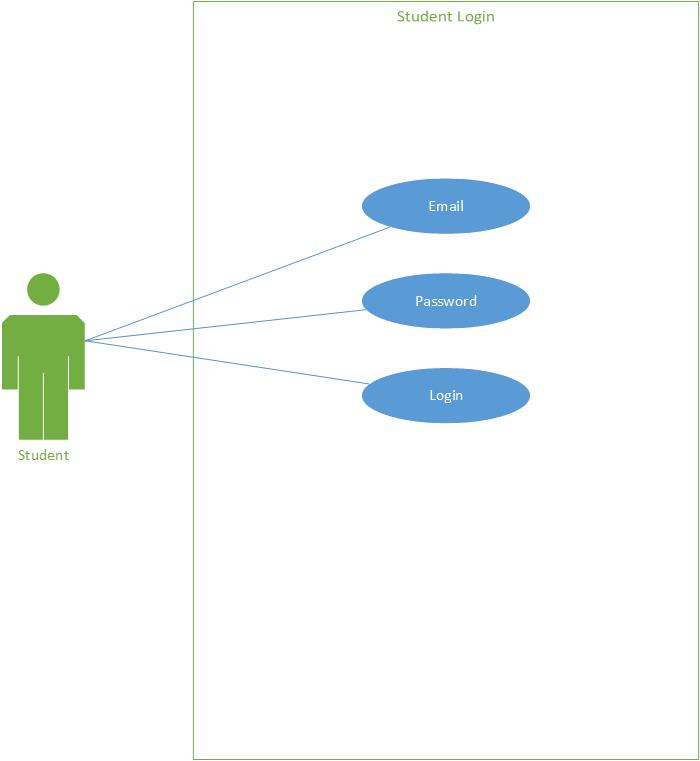
****

Fig.2.9.3

### 2.9.4) Student Change/Update Password

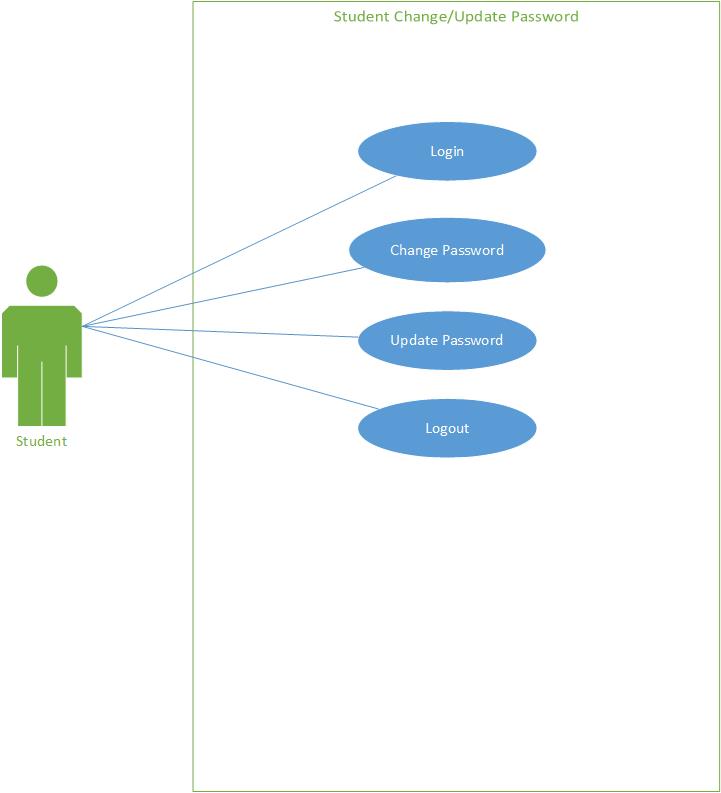
****

Fig.2.9.4

### 2.9.5) Student Update Profile

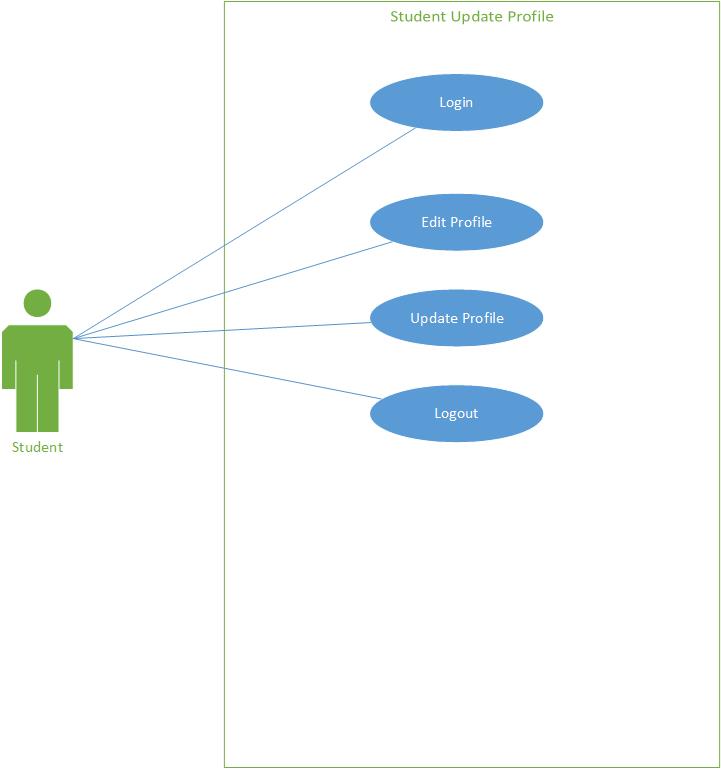
****

Fig.2.9.5

### 2.9.6) Student Get Guidelines

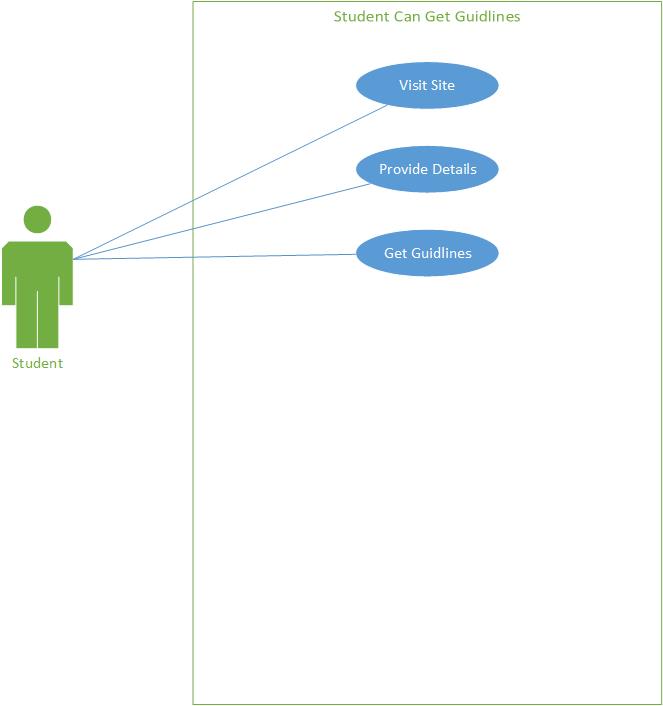
****

Fig.2.9.6

## 2.10) Institute Use Case Diagrams

### 2.10.1) Institute Create Account

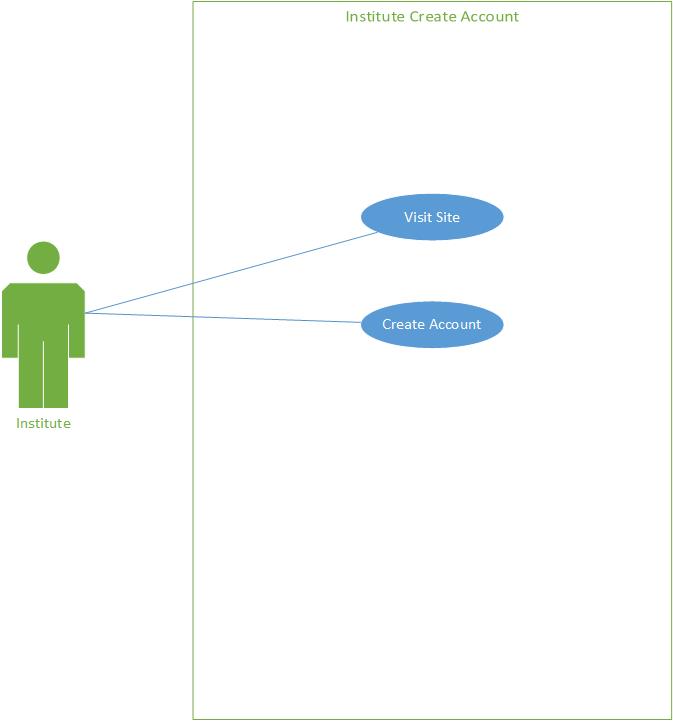


Fig.2.10.1

### 2.10.2) Institute Login

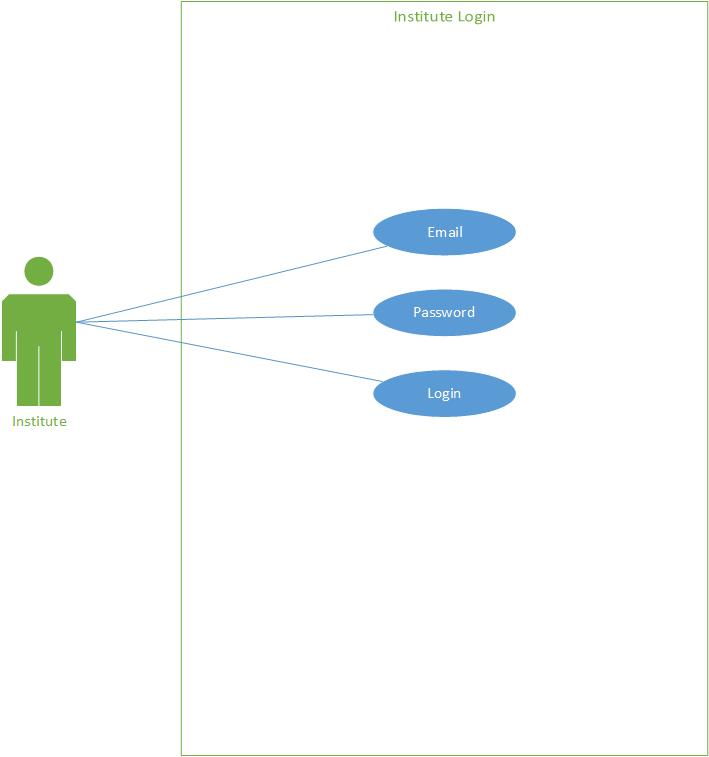


Fig.2.10.2

### 2.10.3) Institute Change/Update Password

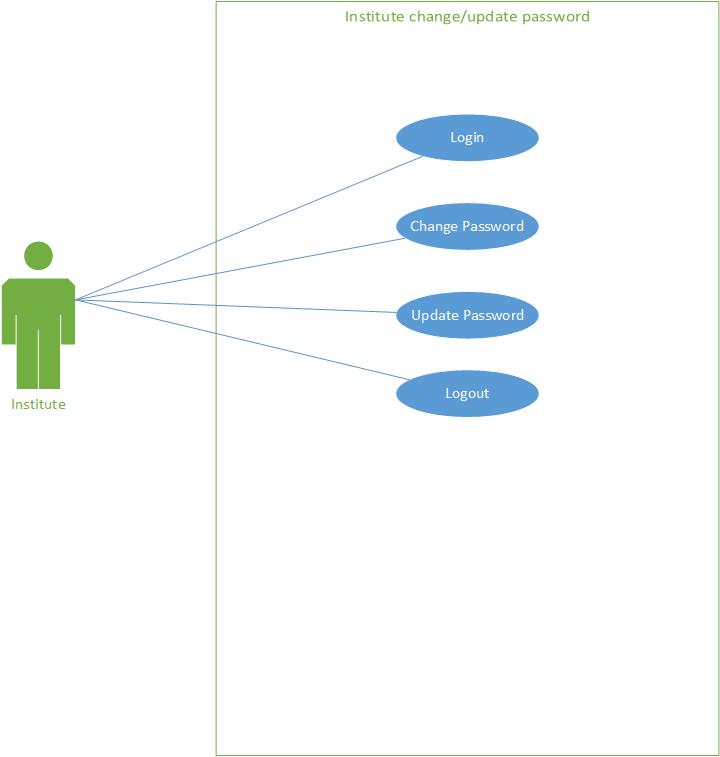


Fig.2.10.3

### 2.10.4) Institute Add Information

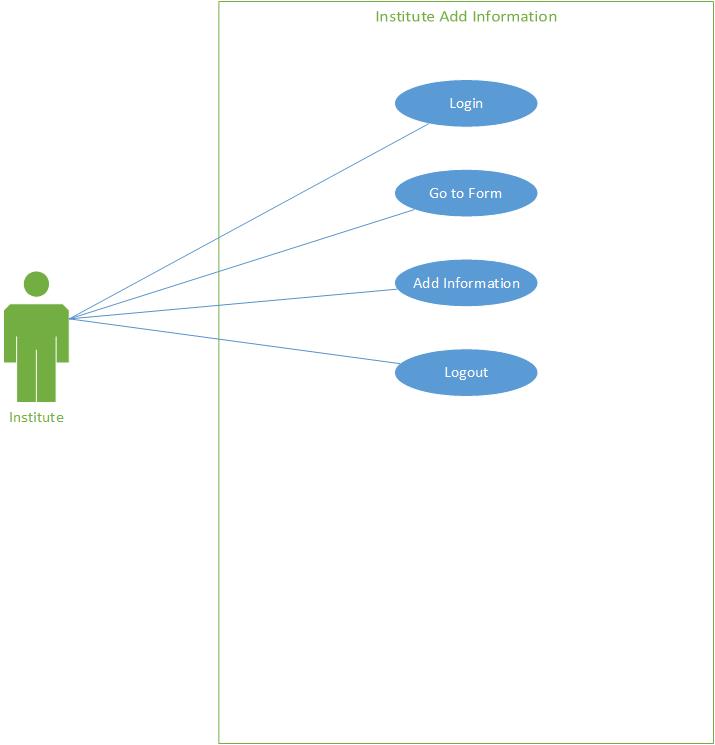


Fig.2.10.4

### 2.10.5) Institute Update Information

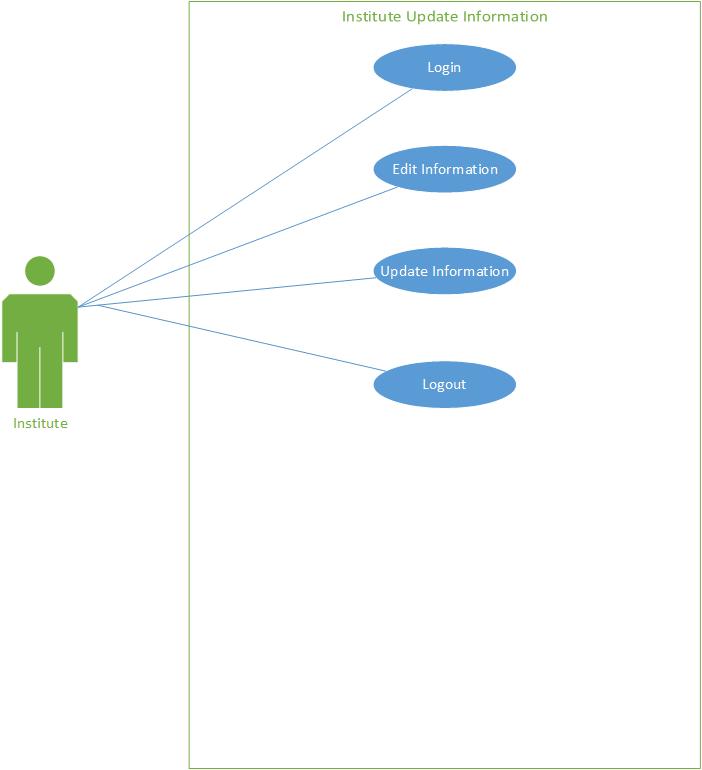
****

Fig.2.10.5

## 2.11) Description of Use Cases

### 2.11.1) Admin Use Case

#### 2.11.1.1) Admin Login

|  |  |  |  |
| --- | --- | --- | --- |
| **Admin** | | | |
| **Actor:** Admin | | | |
| **Feature:** Log In | | | |
| **Use case Id:** | | UC-2.8.1 | |
| **Precondition:** | | None | |
| **Scenarios** | | | |
| **Step#** | **Action** | | **Software Reaction** |
| **1.** | Admin Login | | Admin can login through email and password. |
| **Alternate Scenarios:** If the admin forgets login password then he will not login. | | | |
| **Post Conditions** | | | |
| **Step#** | **Description** | | |
| **1** | Successful login | | |
| **2** | Show Main GUI Dashboard | | |

#### 2.11.1.2) Admin Change/Update Password

|  |  |  |  |
| --- | --- | --- | --- |
| **Admin** | | | |
| **Actor:** Admin  . | | | |
| **Feature:** Update Password | | | |
| **Use case Id:** | | UC-2.8.2 | |
| **Pre-condition:** | | Admin should be log in. | |
| **Scenarios** | | | |
| **Step#** | **Action** | | **Software Reaction** |
| **1.** | Log In | | System login him/her using GUI |
| **2.** | Change Password | | After log in the admin can update his/her password |
| **Alternate Scenarios: Null** | | | |
| **Post Conditions** | | | |
| **Step#** | **Description** | | |
| **1** | Successful login | | |
| **2** | Show Main GUI | | |
| **3** | Update Password | | |

#### 2.11.1.3) Admin Add Information

|  |  |  |  |
| --- | --- | --- | --- |
| **Admin** | | | |
| **Actor:** Admin  . | | | |
| **Feature:** Add Information | | | |
| **Use case Id:** | | UC-2.8.3 | |
| **Pre-condition:** | | Admin should be log in. | |
| **Scenarios** | | | |
| **Step#** | **Action** | | **Software Reaction** |
| **1.** | Admin Log In | | System login him/her using GUI |
| **2.** | Use System | | Add Information |
| **Alternate Scenarios: Null** | | | |
| **Post Conditions** | | | |
| **Step#** | **Description** | | |
| **1** | Successful login | | |
| **2** | Show Main GUI | | |
| **3** | Information Added | | |

#### 2.11.1.4) Admin Manage Information

|  |  |  |  |
| --- | --- | --- | --- |
| **Admin** | | | |
| **Actor:** Admin  . | | | |
| **Feature:** Manage Existing Information | | | |
| **Use case Id:** | | UC-2.8.4 | |
| **Pre-condition:** | | Admin should be log in. | |
| **Scenarios** | | | |
| **Step#** | **Action** | | **Software Reaction** |
| **1.** | Admin Log In | | System login him/her using GUI. |
| **2.** | Use System | | Manage existing Information. |
| **Alternate Scenarios: Null** | | | |
| **Post Conditions** | | | |
| **Step#** | **Description** | | |
| **1** | Success full login | | |
| **2** | Show Main GUI | | |
| **3** | Manage All Available Information | | |

#### 2.11.1.5) Admin Manage Registrations

|  |  |  |  |
| --- | --- | --- | --- |
| **Admin** | | | |
| **Actor:** Admin  . | | | |
| **Feature:** Manage the Registrations. | | | |
| **Use case Id:** | | UC-2.8.5 | |
| **Precondition:** | | The admin should be logged in. | |
| **Scenarios** | | | |
| **Step#** | **Action** | | **Software Reaction** |
| **1.** | User Login | | System login him/her using GUI. |
| **2.** | Use System | | Check the total registration requests. |
| **3.** | Use System | | Approve or Reject the registration. |
| **Alternate Scenarios: Null** | | | |
| **Post Conditions** | | | |
| **Step#** | **Description** | | |
| **1** | Success full login | | |
| **2** | Show Main GUI | | |
| **3** | Approve or reject the registration requests. | | |

#### 2.11.1.6) Admin Manage Users

|  |  |  |  |
| --- | --- | --- | --- |
| **Admin** | | | |
| **Actor:** Admin  . | | | |
| **Feature:** Checking the total registered users. | | | |
| **Use case Id:** | | UC-2.8.6 | |
| **Precondition:** | | The admin should be logged in. | |
| **Scenarios** | | | |
| **Step#** | **Action** | | **Software Reaction** |
| **1.** | User Login | | System login him/her using GUI. |
| **2.** | Use System | | Check the total registered users. |
|  | Use System | | Manage all the registered users and their details. |
| **Alternate Scenarios: Null** | | | |
| **Post Conditions** | | | |
| **Step#** | **Description** | | |
| **1** | Success full login | | |
| **2** | Show Main GUI | | |
| **3** | Checking the total registered customers and manage all of their details. | | |

## 2.11.2) Student Use Case

#### 2.11.2.1) Student View Information

|  |  |  |  |
| --- | --- | --- | --- |
| **Student** | | | |
| **Actor:**  Student  . | | | |
| **Feature:** Find Details About Institutes and Admissions | | | |
| **Use case Id:** | | UC-2.9.1 | |
| **Pre-condition:** | | Student can view details with Log in/without login. | |
| **Scenarios** | | | |
| **Step#** | **Action** | | **Software Reaction** |
| **1.** | Show GUI | | Student can find all information with login as well as without login. |
| **Alternate Scenarios: Null** | | | |
| **Post Conditions** | | | |
| **Step#** | **Description** | | |
| **1** | Without log in | | |
| **2** | With log in | | |

#### 2.11.2.2) Student Registration

|  |  |  |  |
| --- | --- | --- | --- |
| **Student** | | | |
| **Actor:**  Student  . | | | |
| **Feature:** Student can view information without registration. But if he/she get registered, then system can inform them about new updates. | | | |
| **Use case Id:** | | UC-2.9.2 | |
| **Pre-condition:** | | Student should be registered. | |
| **Scenarios** | | | |
| **Step#** | **Action** | | **Software Reaction** |
| **1.** | Student sign up | | The student will enter required information for registration. |
| **Alternate Scenarios:** If student will not sign up, then he/she can only view the available information. | | | |
| **Post Conditions** | | | |
| **Step#** | **Description** | | |
| **1** | Successful registration | | |
| **2** | Show Main GUI | | |

#### 2.11.2.3) Student Login

|  |  |  |  |
| --- | --- | --- | --- |
| **Student** | | | |
| **Actor:**  Student  . | | | |
| **Feature:** Log In | | | |
| **Use case Id:** | | UC-2.9.3 | |
| **Pre-condition:** | | Student should be Registered. | |
| **Scenarios** | | | |
| **Step#** | **Action** | | **Software Reaction** |
| **1.** | Student registration | | The student can log in after sing up. |
| **Alternate Scenarios:** If password is wrong then he/she can’t log in | | | |
| **Post Conditions** | | | |
| **Step#** | **Description** | | |
| **1** | Successful registration. | | |
| **2** | Log in | | |
| **3** | Show Main GUI | | |

#### 2.11.2.4) Student Change/Update Password

|  |  |  |  |
| --- | --- | --- | --- |
| **Student** | | | |
| **Actor:**  Student  . | | | |
| **Feature:** Forgotten Password | | | |
| **Use case Id:** | | UC-2.9.4 | |
| **Pre-condition:** | | Student should be Registered. | |
| **Scenarios** | | | |
| **Step#** | **Action** | | **Software Reaction** |
| **1.** | Show GUI | | The student can change or update password after log in or before log in |
| **Alternate Scenarios: Null** | | | |
| **Post Conditions** | | | |
| **Step#** | **Description** | | |
| **1** | Change password after login. | | |
| **2** | Recover password if password is forgotten. | | |

#### 2.11.2.5) Student Update Profile

|  |  |  |  |
| --- | --- | --- | --- |
| **Student** | | | |
| **Actor:**  Student  . | | | |
| **Feature:** Change/Update profile | | | |
| **Use case Id:** | | UC-2.9.5 | |
| **Pre-condition:** | | Student should be Login. | |
| **Scenarios** | | | |
| **Step#** | **Action** | | **Software Reaction** |
| **1.** | Show GUI | | The Student can update or change profile after login. |
| **Alternate Scenarios: Null** | | | |
| **Post Conditions** | | | |
| **Step#** | **Description** | | |
| **1** | Change profile | | |
| **2** | Update profile info | | |

### 2.11.3) Institute Use Case

#### 2.11.3.1) Institute Registration

|  |  |  |  |
| --- | --- | --- | --- |
| **Institute** | | | |
| **Actor:**  Institute  . | | | |
| **Feature:** Institutes can add their information after registration. | | | |
| **Use case Id:** | | UC-2.10.1 | |
| **Pre-condition:** | | Institute should be registered. | |
| **Scenarios** | | | |
| **Step#** | **Action** | | **Software Reaction** |
| **1.** | Institute sign up | | The Institute will enter required information for registration. |
| **Alternate Scenarios:** NULL | | | |
| **Post Conditions** | | | |
| **Step#** | **Description** | | |
| **1** | Successful registration. | | |
| **2** | Show Main GUI. | | |

#### 2.11.3.2) Institute Login

|  |  |  |  |
| --- | --- | --- | --- |
| **Institute** | | | |
| **Actor:**  Institute  . | | | |
| **Feature:** Log In | | | |
| **Use case Id:** | | UC-2.10.2 | |
| **Pre-condition:** | | Institute should be Registered. | |
| **Scenarios** | | | |
| **Step#** | **Action** | | **Software Reaction** |
| **1.** | Institute registration | | Institute can login after sing up. |
| **Alternate Scenarios:** If password is wrong then institute can’t log in. | | | |
| **Post Conditions** | | | |
| **Step#** | **Description** | | |
| **1** | Successful registration. | | |
| **2** | Log in | | |
| **3** | Show Main GUI | | |

#### 2.11.3.3) Institute Change/Update Password

|  |  |  |  |
| --- | --- | --- | --- |
| **Institute** | | | |
| **Actor:**  Institute  . | | | |
| **Feature:** Forgotten Password | | | |
| **Use case Id:** | | UC-2.10.3 | |
| **Pre-condition:** | | Institute should be Registered. | |
| **Scenarios** | | | |
| **Step#** | **Action** | | **Software Reaction** |
| **1.** | Show GUI | | Institute can change password after log in or recover password before login. |
| **Alternate Scenarios: Null** | | | |
| **Post Conditions** | | | |
| **Step#** | **Description** | | |
| **1** | Change password after login. | | |
| **2** | Recover password if password is forgotten. | | |

#### 2.11.3.4) Institute Add Information

|  |  |  |  |
| --- | --- | --- | --- |
| **Institute** | | | |
| **Actor:**  Institute  . | | | |
| **Feature:** Institutes can add details after login. | | | |
| **Use case Id:** | | UC-2.10.4 | |
| **Pre-condition:** | | Institute should be login. | |
| **Scenarios** | | | |
| **Step#** | **Action** | | **Software Reaction** |
| **1.** | Show GUI | | Institute can add their details after login. |
| **Alternate Scenarios: Null** | | | |
| **Post Conditions** | | | |
| **Step#** | **Description** | | |
| **1** | Login after registration. | | |
| **2** | Add details about the institute. | | |
| **3** | Information added on the system. | | |

#### 2.11.3.5) Institute Update Information

|  |  |  |  |
| --- | --- | --- | --- |
| **Institute** | | | |
| **Actor:**  Institute  . | | | |
| **Feature:** Institutes can update their details after login. | | | |
| **Use case Id:** | | UC-2.10.5 | |
| **Pre-condition:** | | Institute should be login. | |
| **Scenarios** | | | |
| **Step#** | **Action** | | **Software Reaction** |
| **1.** | Show GUI | | Institute can update their details after login. |
| **Alternate Scenarios: Null** | | | |
| **Post Conditions** | | | |
| **Step#** | **Description** | | |
| **1** | Login after registration. | | |
| **2** | Update details about the institute. | | |
| **3** | Details are updated and published on the system. | | |

# CHAPTER 3: DESIGNING

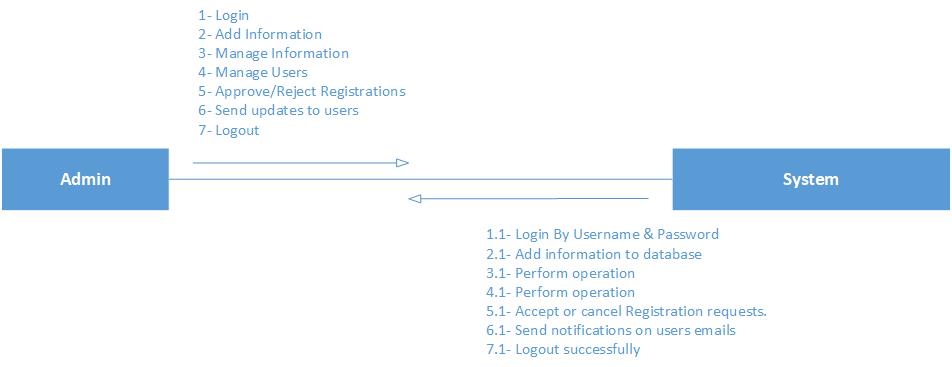
* 1. **Collaboration Diagram**

3.1.1 Admin Collaboration Diagram

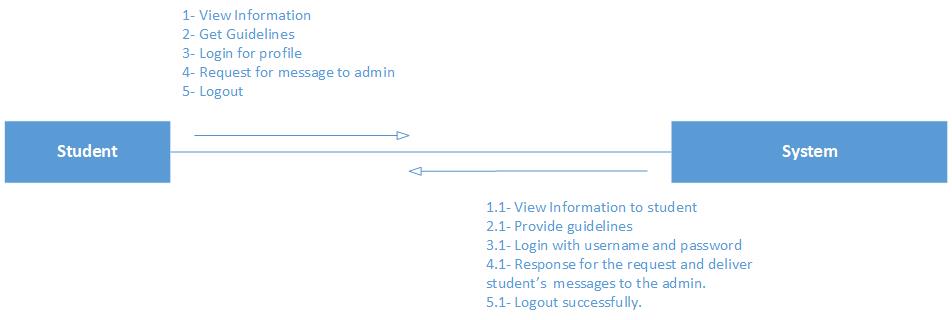
* + 1. Student Collaboration Diagram
    2. Institute Collaboration Diagram
  1. **Activity Diagram**
     1. Admin Activity Diagram
     2. Student Activity Diagram
     3. Institute Activity Diagram
  2. **Data Flow Diagram**
     1. Context Level Data Flow Diagram
     2. Admin Side Level 1 Data Flow Diagram
     3. User Side Level 1 Data Flow Diagram

## 3.1) System Collaboration Diagram

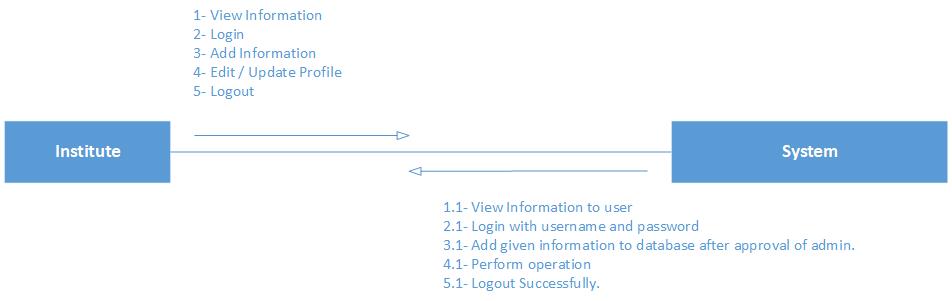
### 3.1.1) Admin Collaboration Diagram

****

### 3.1.2) Student Collaboration Diagram

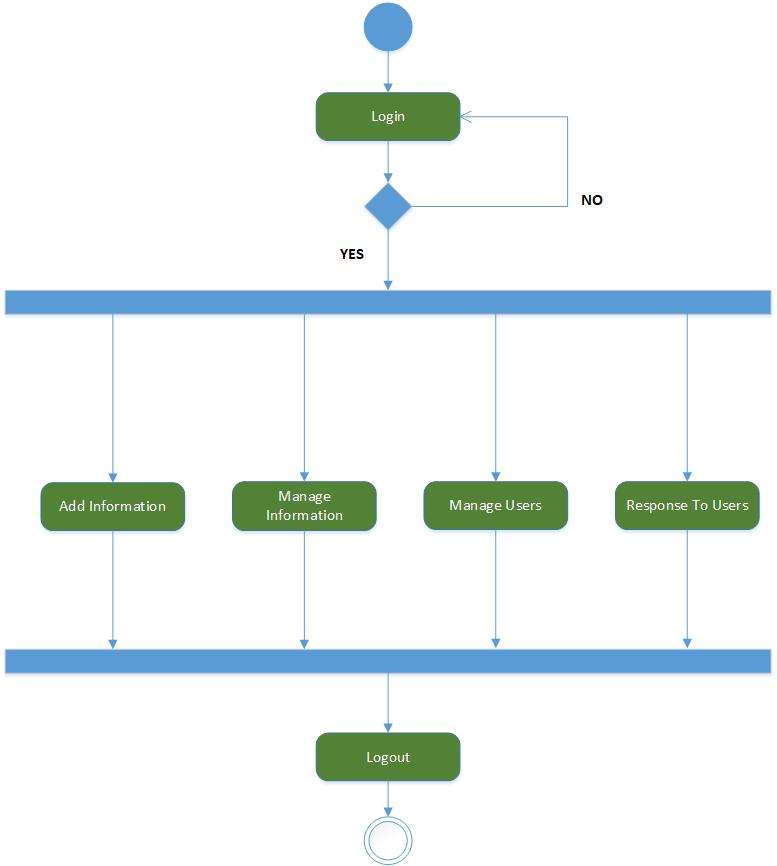


### 3.1.3) Institute Collaboration Diagram

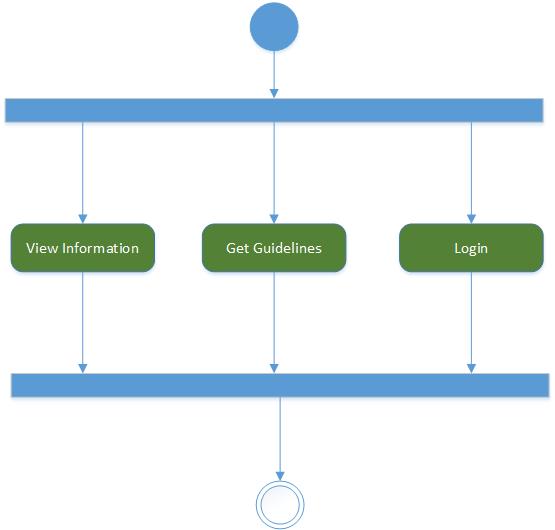


## 3.2) Activity Diagram

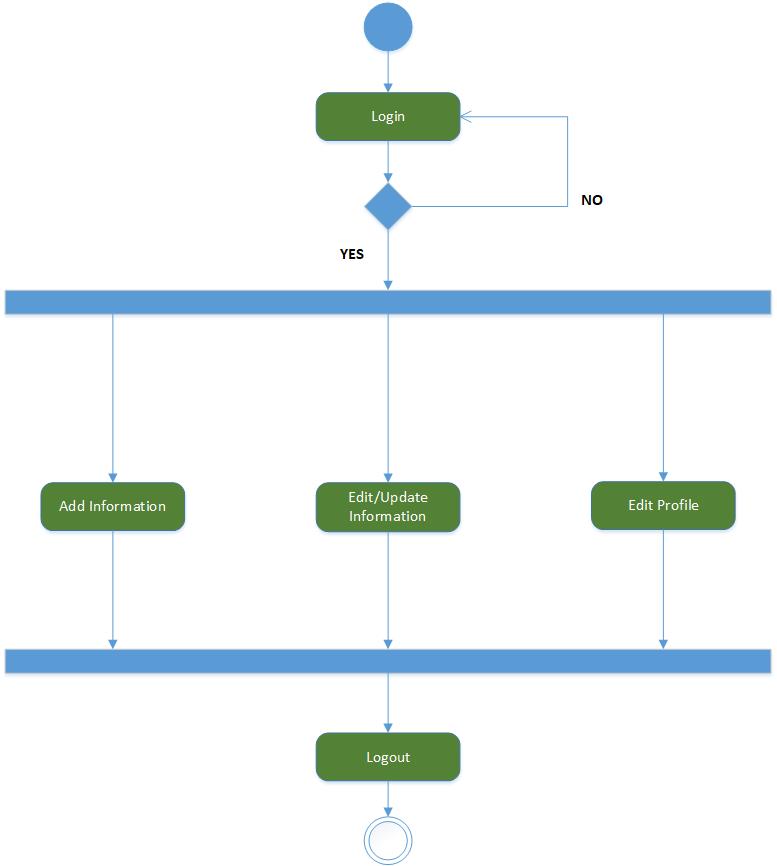
### 3.2.1) Admin Activity Diagram



### 3.2.2) Student Activity Diagram

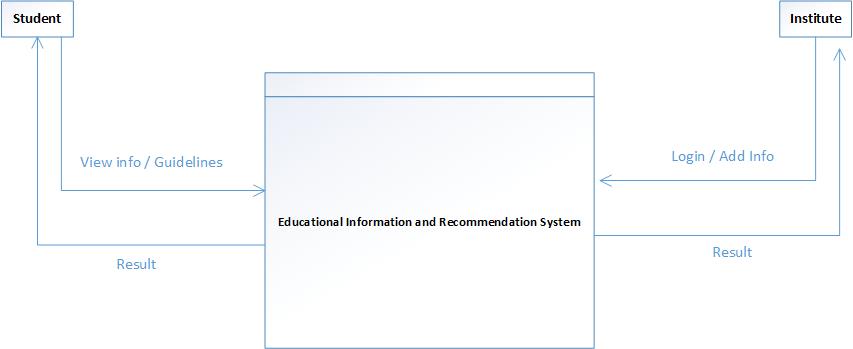


### 3.2.3) Institute Activity Diagram

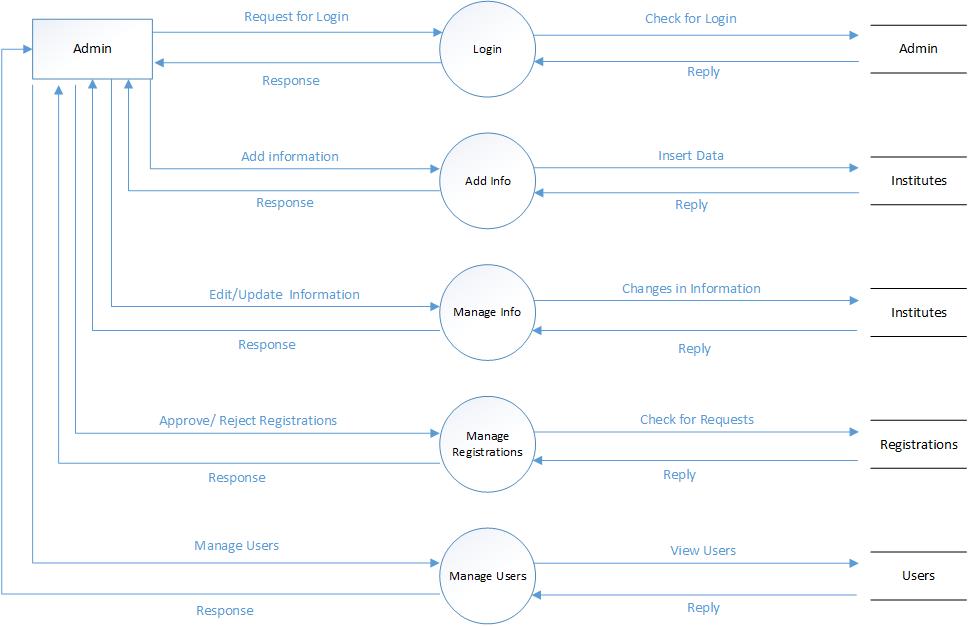


## 3.3) Data Flow Diagram

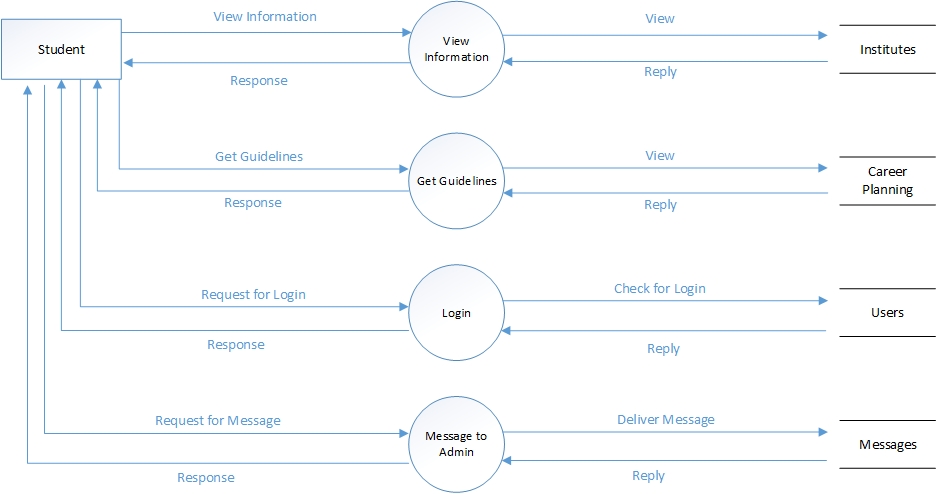
### 3.3.1) Context Level Diagram



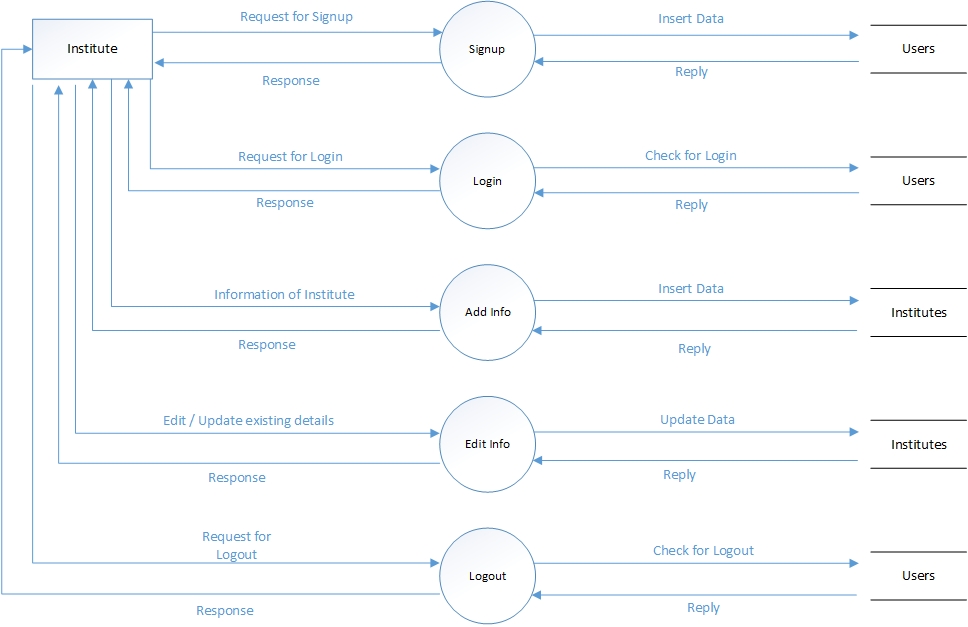
### 3.3.2) Admin Side Data Flow Diagram - Level 1



### 3.3.3) Student Side Data Flow Diagram - Level 1



### 3.3.4) Institute Side Data Flow Diagram - Level 1



# CHAPTER 4: DELIVERABLE (User Interface Design)

## Introduction

A user interface design consists of page elements that should be visualized GUI snapshot of the project.

## Interface Detail

|  |
| --- |
| **User Interface :** Login Admin |
| **Snapshot** |
|  |

|  |
| --- |
| **User Interface :** Admin Change Password |
| **Snapshot** |
|  |

|  |
| --- |
| **User Interface :** Admin Add Information |
| **Snapshot** |
|  |

|  |
| --- |
| **User Interface : Admin Manage Information** |
| **Snapshot** |
|  |

|  |
| --- |
| **User Interface :** Admin Manage Registrations |
| **Snapshot** |
|  |

|  |
| --- |
| **User Interface :** Admin Manage Users |
| **Snapshot** |
|  |

|  |
| --- |
| **User Interface :** Student View Details |
| **Snapshot** |
|  |

|  |
| --- |
| **User Interface :** Student Registration |
| **Snapshot** |
|  |

|  |
| --- |
| **User Interface :** Student Log In |
| **Snapshot** |
|  |

|  |
| --- |
| **User Interface :** Forget Password |
| **Snapshot** |
|  |

|  |  |
| --- | --- |
| **User Interface :** Student Get Guidelines | |
| **Interface Id.** | 13 |
| **Use case Reference** | UC-2.17.5 |
| **Snapshot** | |
|  | |

|  |
| --- |
| **User Interface :** Student Message to Admin |
| **Snapshot** |
|  |

|  |
| --- |
| **User Interface :** Institute signup |
| **Snapshot** |
|  |

|  |
| --- |
| **User Interface :** Institute Log In |
| **Snapshot** |
|  |

|  |
| --- |
| **User Interface :** Forget Password |
| **Snapshot** |
|  |

|  |
| --- |
| **User Interface :** Institute Add information |
| **Snapshot** |
|  |