|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Software Design Specifications**  ***[Student Management System for Azaan]***  **Version: [2.0]**   |  |  | | --- | --- | | Supervisor | Asim ali | | Co Supervisor (if any) |  | | Project Team | * Abubakar Siddiq(1312166) * Amirul Sunesara(1312169) * Aamir shehzad(1312165) | | Submission Date | 07/05/2016 | |     Document History   |  |  |  |  | | --- | --- | --- | --- | | Version | Name of Person | Date | Description of change | |  | N.A |  | |  |  |  | |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |  |  |  |         Distribution List   |  |  | | --- | --- | | **Name** | **Role** | | **Asim Ali** | Supervisor | |  | Co Supervisor (if any) | |  |  |       Document Sign-Off   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Version** | **Sign-off Authority** | **Project Role** | **Signature** | **Sign-off Date** | | 1.0 | Faria Jameel | 15/12/2015 | | 2.0 | M. Asim Ali | 5/07/2016 | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |

Document Information

| **Category** | **Information** |
| --- | --- |
| Customer | SZABIST |
| Project | Azaan Student Management System |
| Document | Software Design Specification |
| Document Version | 1.0 |
| Status | Draft |
| Author(s) | * Abubakar Siddiq * Amirul sunesra * Aamir Shehzad |
| Approver(s) | Asim Ali |
| Issue Date | 06/05/2016 |
| Document Location |  |
| Distribution | Advisor |
|  |  |

Definition of Terms, Acronyms and Abbreviations

[This section should provide the definitions of all terms, acronyms, and abbreviations required to interpret the terms used in the document properly. ]

| **Term** | **Description** |
| --- | --- |
| ASP | Active Server Pages |
| DD | Design Specification |
| Sql | Structured Query language |

**Table of Contents**

Contents

[2 Introduction 7](#_Toc450374194)

[2.1 Purpose of Document 7](#_Toc450374195)

[2.2 Intended Audience 7](#_Toc450374196)

[2.3 Document Convention 7](#_Toc450374197)

[2.4 Project Overview 7](#_Toc450374198)

[2.5 Scope 7](#_Toc450374199)

[3 System Architecture 8](#_Toc450374200)

[3.1 System Level Architecture 8](#_Toc450374201)

[3.2 Software Architecture 8](#_Toc450374202)

[4 Design Strategy 9](#_Toc450374203)

[5 Detailed System Design 10](#_Toc450374204)

[5.1 Database Design 10](#_Toc450374205)

[5.1.1 ER Diagram 11](#_Toc450374206)

[5.1.2 Data Dictionary 12](#_Toc450374207)

[5.2 Application Design 20](#_Toc450374208)

[5.2.1 Sequence Diagram 20](#_Toc450374209)

[5.2.2 Activity Diagram 27](#_Toc450374210)

[5.3 Data Flow Diagram Level 1 Diagram 34](#_Toc450374211)

[5.4 State Transition diagram 35](#_Toc450374212)

[5 GUIs of your Prototype/Software (not applicable) 38](#_Toc450374213)

[6 References (not applicable) 38](#_Toc450374214)

[7 Appendices(not applicable) 39](#_Toc450374215)

# Introduction

## Purpose of Document

This document is made to convey or compile all the proposed designs for the project. This document will cover data flow design, application flow design and other stuff related to design of application. It is intended to be used by developers (to construct application as per design), Customer (to verify that design cover all of their requirements) and later by testing team (to verify final product against design).

Please note that the application is built using object oriented programming approach.

## Intended Audience

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

* Development team
* Customer
* Testing team

## Document Convention

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

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

## Project Overview

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

Project will be designed using agile approach and will be built in increments.

## Scope

System is developed to provide our client with the following:

1. Platform with it can control all of its activity.
2. A forum where instructor and students can communicate
3. A comfortable way for its students where they can register courses and keep updated with latest happenings
4. Course resource sharing environment

# System Architecture

[This section should provide a high-level overview of how the functionality and responsibilities of the system are partitioned and then assigned to subsystems or components. The main purpose is to gain a general understanding of how the system is decomposed, and how the individual parts work together to provide the desired functionality].

## System Level Architecture

[The architecture should decompose the system at a top level in a way that provides a foundation for more detailed design work. The architecture discusses relationships and roles of system elements (subsystems, components, modules, etc.), but does not provide internal details. Areas for consideration are:

* System decomposition into elements
* The relationship between the elements
* Interfaces to external systems
* Major physical design issues such as where elements will execute
* Global design strategies such as error handling

**NOTE**: You may use appropriate UML diagrams (Package and Component diagrams) to document the overall system architecture. ]

## Software Architecture

[The software architecture should include how User level Layer will interact with Database layer. Use diagram for showing the interaction between the layers.

* User Interface Layer
* Middle Tier
* Data Access Layer.
* Or other

You can give any other architecture also]

# Design Strategy

System will be designed keeping its performance on priority. Fast access to data will a stream line focus of developing team. As the application will be hosted on remote servers, any tool that might create problems to access the system will be avoided.

Application is intended to be used by variety of users upon several different types of devices regardless of age, so design will be made in such a manner that can easily be used by every group of people. Below are some other strategies used while designing the system:

1. Reusability: With emerging software market, this system will be developed keeping in mind that it by used by any other client with slight changes to his/her need.
2. Extension: As our client is one of the leading emerging and innovative organization of its type in Pakistan, it contains a high priority that it might increase its operation in several other fields as well in future which must be accommodated under the umbrella of this organization so therefore system will be developed in a such manner that it can be extended as per our clients need in future.
3. Data management: This system not only intends to provide one umbrella working solution to our client but it will also act as a backbone data manger for the organization so no matter how the system is developed its primary feature must be kept while designing.

.

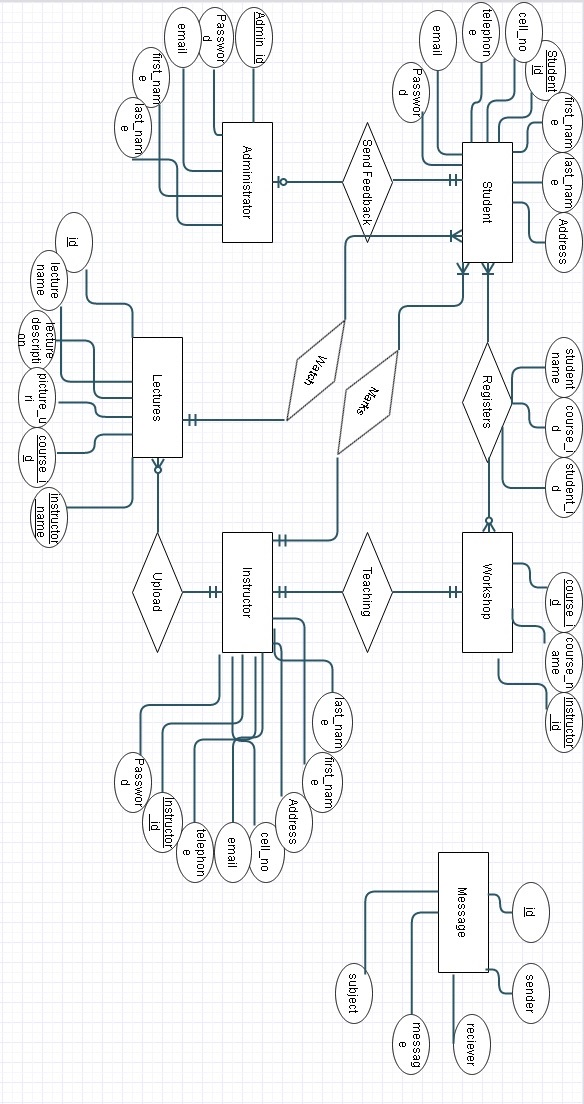
# Detailed System Design

[A detailed design should include the following:

* Detailed class diagram along with a detailed description of all attributes, functions or methods specifying interactions between different classes/modules.
* Logical data model (E/R model)
* Detailed GUIs]

## Database Design

### ER Diagram



### Data Dictionary

#### Instructor

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Instructor | | | | | | | |
| **Name** | | Instructor | | | | | |
| **Where-used/how-used** | | * Login form * Adding new course * Messaging * Video upload | | | | | |
| **Content description** | |  | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| Instructor\_id | Holds unique id of instructor | | Int | 15 | No | Nil | Primary key |
| Instructor\_fname | Contains first name of instructor | | Varchar | 50 | No | Nil |  |
| Instructor\_lanme | contains last name of instructor | | Varchar | 50 | No | Nil |  |
| Instructor\_no | Contains contact number of instructor | | Varchar | 50 | No | Null |  |
| Instructor\_email | Contains Instructors email Id | | Varchar | 50 | No | Null |  |
| Password | Contains Password for user login of Instructor | | Varchar | 50 | No | Null |  |
| Telephone | Contains phone number of instructor | | Varchar | 50 | No | Null |  |
| Address | Contains instructor residential Address | | Varchar | 50 | No | Null |  |

#### Marks

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Marks | | | | | | | |
| **Name** | | Marks | | | | | |
| **Where-used/how-used** | | * Marks upload * View marks | | | | | |
| **Content description** | |  | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| course\_id | Unique identifier for course | | Int | 15 | No | Nil | Foreign key |
| Student\_id | Unique identifier for student | | Int | 15 | No | Nil | Foreign key |
| Marks | Contains marks of student in particular course | | Int | 15 | No | Nil |  |

#### Course Registration

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Course Registration | | | | | | | |
| **Name** | | Course registration | | | | | |
| **Where-used/how-used** | | * Course registration | | | | | |
| **Content description** | |  | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| Student\_id | Unique identifier for student | | Int | 15 | No | Nil | Foreign key |
| Course\_id | Unique identifier for student | | Int | 15 | No | Nil | Foreign key |
| Student\_name | Contains name of student how registered in the course | | Varchar | 50 | No | Nil |  |

#### Student

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Student | | | | | | | |
| **Name** | | Student | | | | | |
| **Where-used/how-used** | | * Login form * New Student sign up * Messaging * Course registration | | | | | |
| **Content description** | |  | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| Student\_id | Holds unique id of Atudent | | Int | 15 | No | Nil | Primary key |
| First\_name | Contains first name of Student | | Varchar | 50 | No | Nil |  |
| Last\_name | Containslast name of Student | | Varchar | 50 | No | Nil |  |
| Telephone | Contains phonenumber of Student | | Varchar | 50 | No | Null |  |
| Cell\_no | Contains Students cell\_no | | Varchar | 50 | No | Null |  |
| Address | Contains residential address of student | | Varchar | 50 | No | Null |  |
| Passsword | Contains password for user logn for student | | Varchar | 50 | No | Null |  |
| Email | Conatins email address of student | | Varchar | 50 | No | Null |  |

#### Course

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Course | | | | | | | |
| **Name** | | Course | | | | | |
| **Where-used/how-used** | | * Adding new course * Course registration | | | | | |
| **Content description** | | Notation for representing content. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| Course\_id | Unique identifirer for each course | | Int | 15 | No | Nil | Primary key |
| Course\_name | Conatins name of the Course | | Varchar | 50 | No | Nil |  |
| Instructor\_id | Contais Id of the instructor assigned to Course | | Int | 15 | No | Nil | Foreign key |

#### Administrator

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Administrator | | | | | | | |
| **Name** | | Administrator | | | | | |
| **Where-used/how-used** | | * Login * Messaging | | | | | |
| **Content description** | | Notation for representing content. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| Admin\_id | Unique identifier for Administartor | | Int | 15 | No | Nil | Primary key |
| Admin\_password | Contains password for Administrator user login | | Varchar | 50 | No | Nil |  |
| Email | Contains Email ID of Administrator | | Varchar | 50 | No | Nil |  |
| Admin\_fname | Contains first name of administrator | | Varchar | 50 | No | Null |  |
| Admin\_lname | Contains last name of instructor | | Varchar | 50 | No | Null |  |

#### Feedback

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Feedback | | | | | | | |
| **Name** | | Feedback | | | | | |
| **Where-used/how-used** | | * Giving feedback * Reviewing Feedback | | | | | |
| **Content description** | | Notation for representing content. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| Student\_id | Holds unique id of Student | | Int | 15 | No | Nil | Foreign key |
| Instructor | Contains points allotted to instructor | | Int | 15 | No | Nil |  |
| Vanue | Constins points allotted to venue | | Int | 15 | No | Nil |  |
| Volunteer | Contains points allotted to Volunteer | | Int | 15 | No | Null |  |
| Hospitality | Contains points allotted to hospitality | | Int | 15 | No | Null |  |
| Course | Contains points allotted to course | | Int | 15 | No | Null |  |
| Registration | Contains points allotted to registration | | Int | 15 | No | Null |  |
| Atmosphere | Contains points allotted to atmosphere | | Int | 15 | No | Null |  |

#### Message

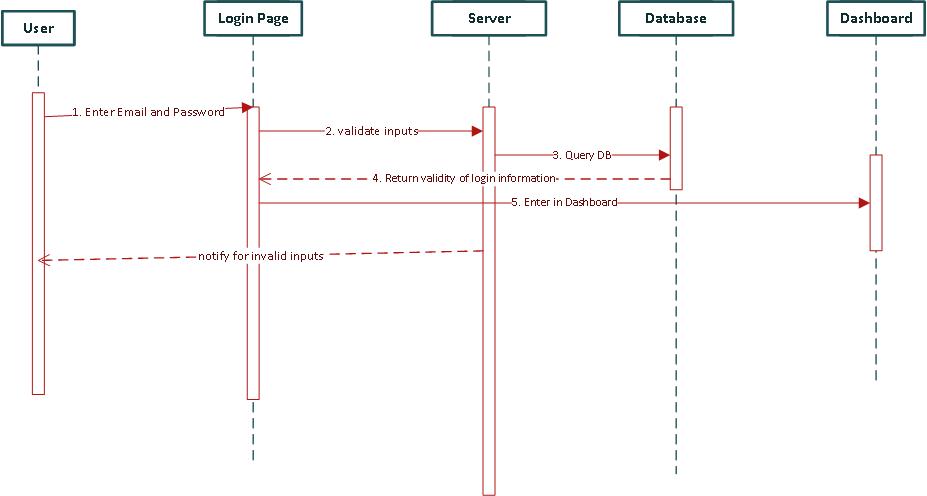
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Message | | | | | | | |
| **Name** | | Message | | | | | |
| **Where-used/how-used** | | * Send Message * Receive Message | | | | | |
| **Content description** | | Notation for representing content. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| Id | Unique id for message | | Int | 15 | No | Nil | Primary key |
| Sender | Name of message sender | | Varchar | 50 | No | Nil |  |
| Receiver | Name of message receiver | | Varchar | 50 | No | Nil |  |
| Message | Content of message | | Varchar | 50 | No | Null |  |
| Subject | Subject of message | | Varchar | 50 | No | Null |  |

## Application Design

### Sequence Diagram

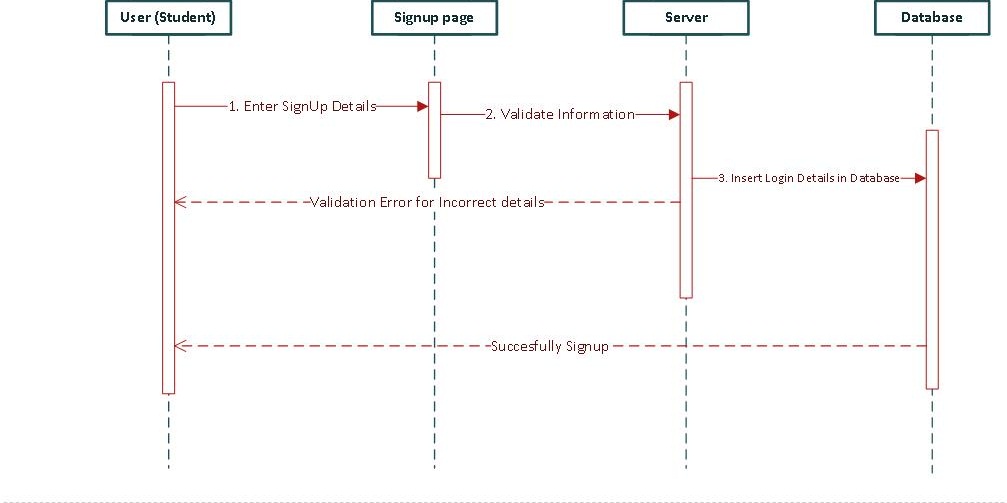
#### <Sequence Diagram 1>

A user of the system could be anyone of student, administrator or instructor. On successful login attempt user will see dashboard.



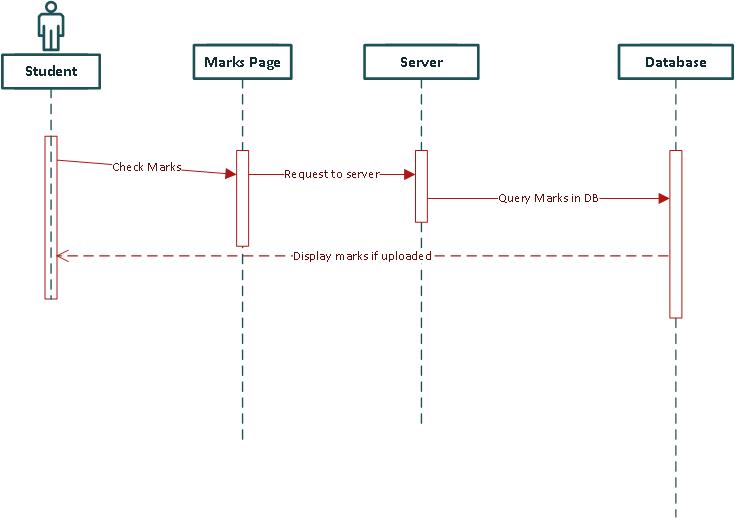
#### <Sequence Diagram 2>

Sign up procedure is simple. User have to enter signup details in home page and get notification on successful signup attempt.



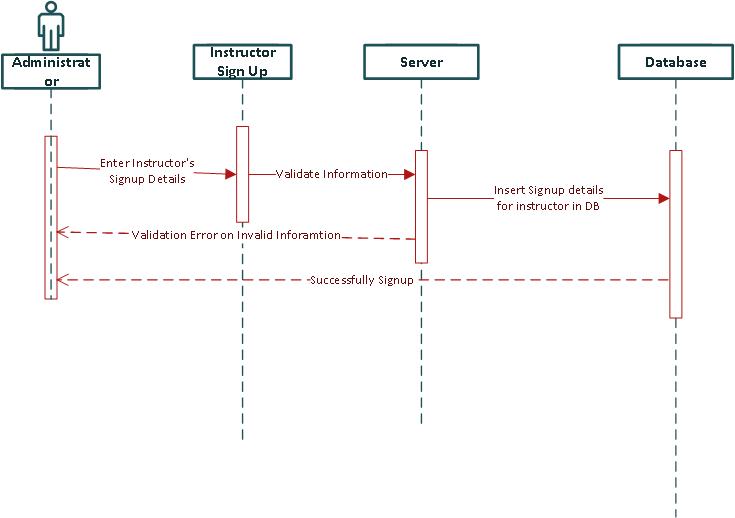
#### <Sequence Diagram 3>

Student can check their marks periodically . Once they are uploaded I can be seen on view marks page.



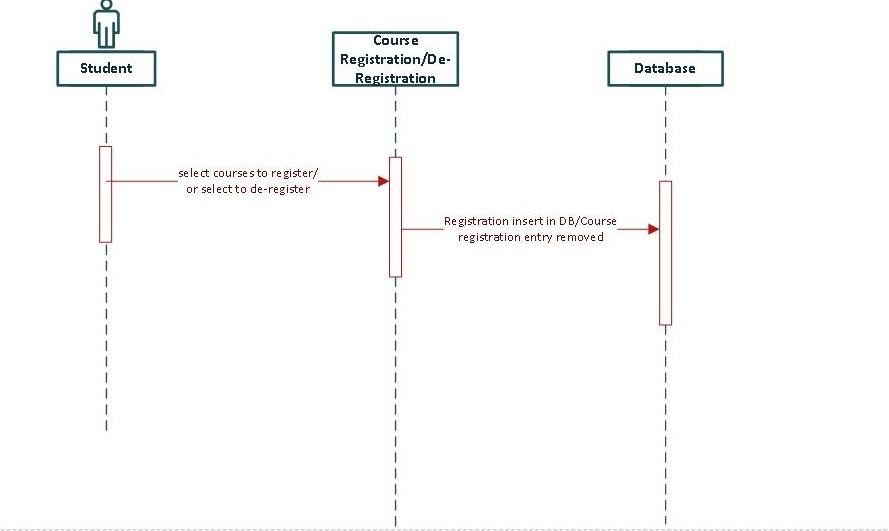
#### <Sequence Diagram 4>

An administrator is privileged to create instructor’s account. Signup procedure is as simple as user signup



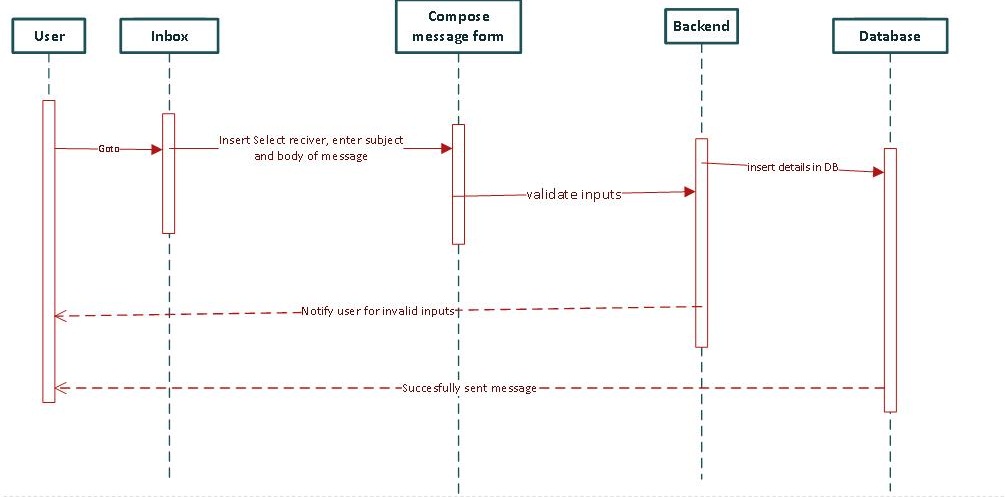
#### <Sequence Diagram 5>

Student can register course once they are being offered and can be able to deregister aswell



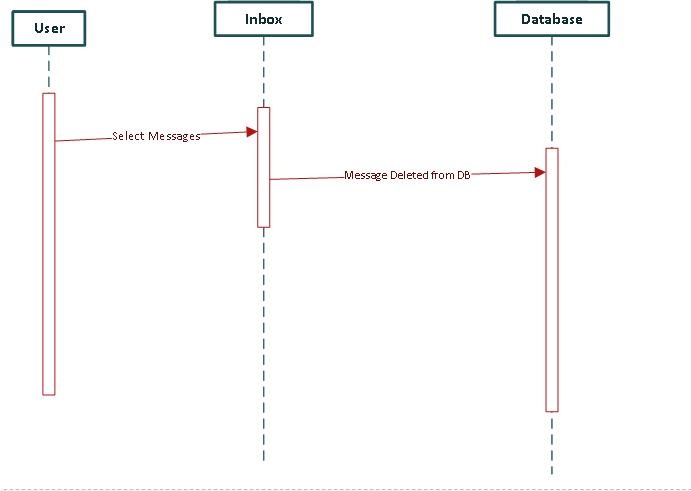
#### <Sequence Diagram 6>

Any user(student, admin, instructor) can compose a message to each other.



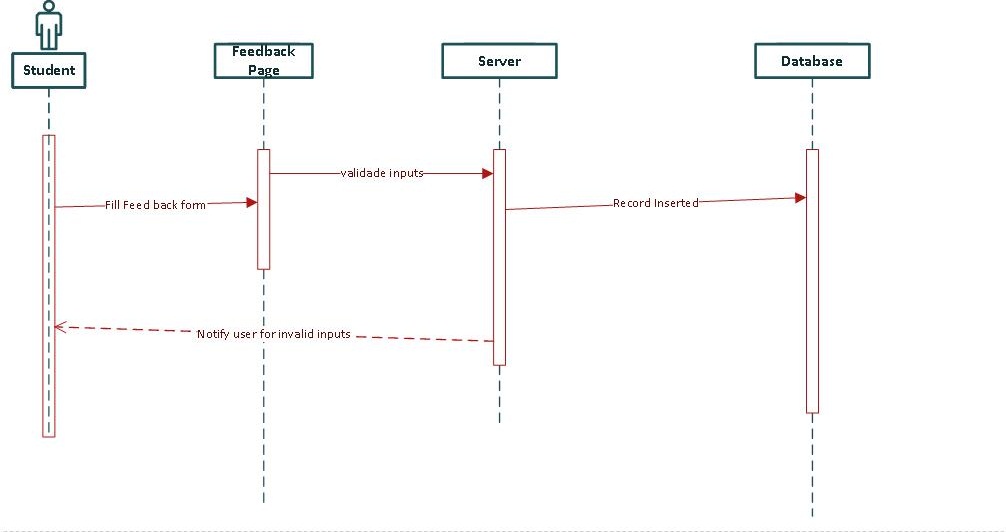
#### <Sequence Diagram 7>

Students can select messages from inbox and perform delete operation by pressing deleting button.



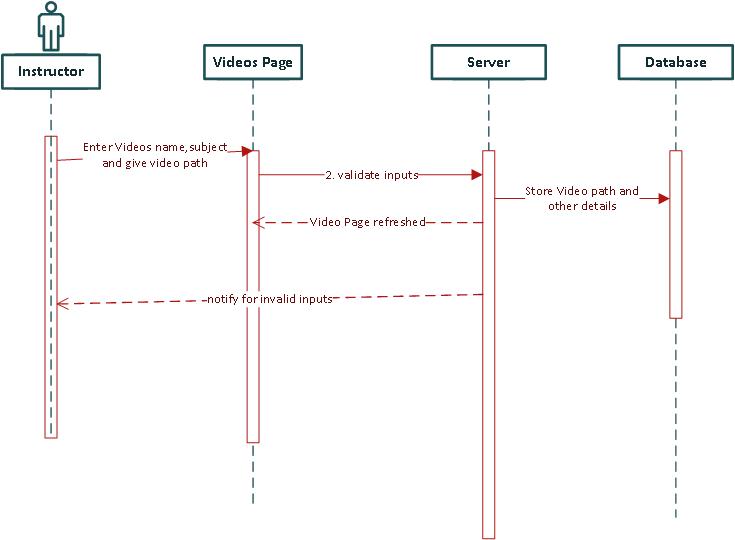
#### <Sequence Diagram 8>

Student can submit feedback only once. When they are able to signup in portal.



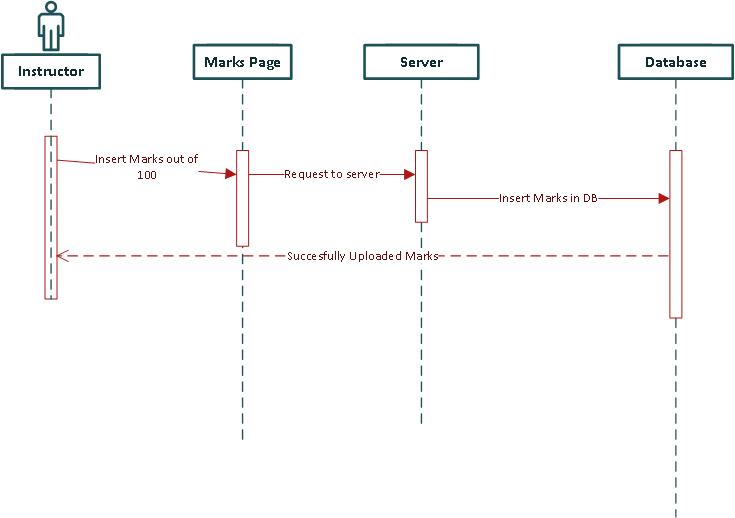
#### <Sequence Diagram 9>

Instructor may occasionally upload their sample video of lectures to motivate or to bring student interest.



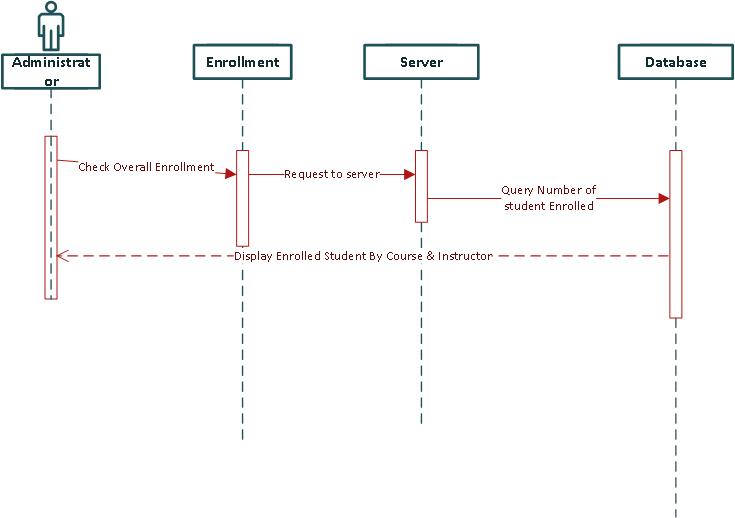
#### <Sequence Diagram 10>

Instructor can upload marks after test are conducted during session. Marks will be out of 100.



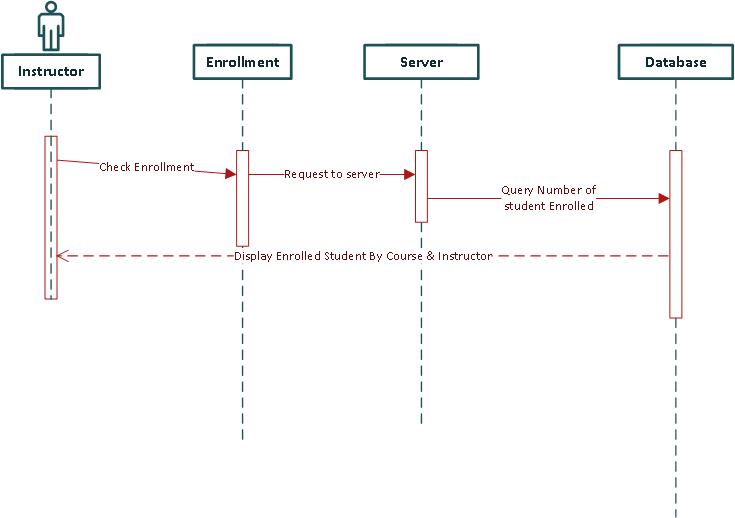
#### <Sequence Diagram 11>

Administrator can check overall enrollment of students. In result he get student record sequentially.



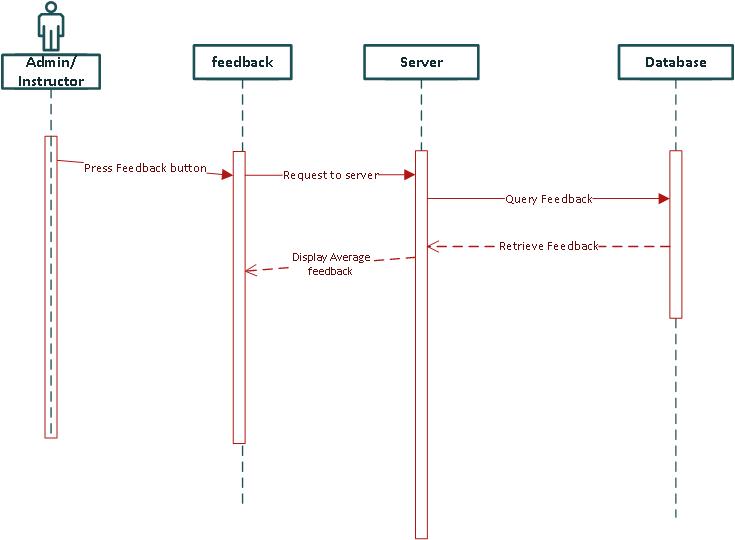
#### <Sequence Diagram 12>

Instructor can also check student enrollment in their courses. In result he get student related information.



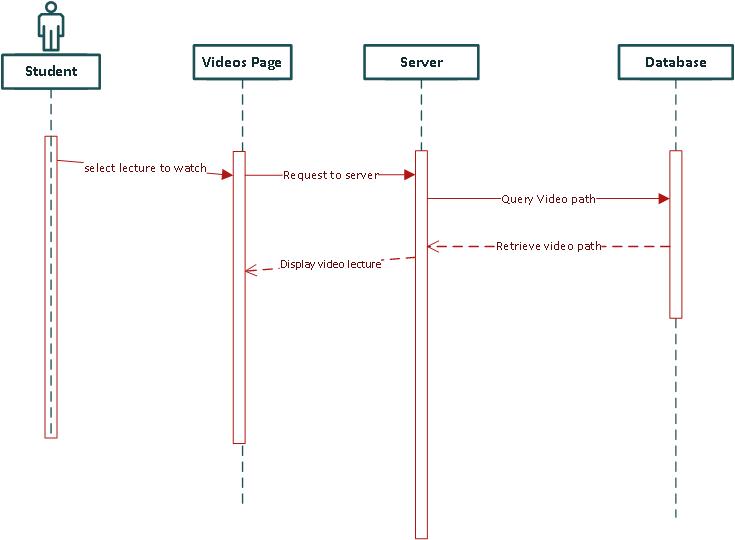
#### <Sequence Diagram 13>

Admin and instructor both check student’s feedback. The feedback is on average basis.



#### <Sequence Diagram 14>

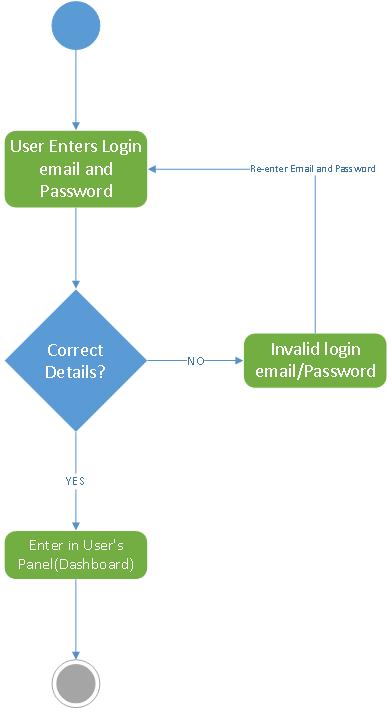
Students can watch uploaded lectures by instructors.



### Activity Diagram

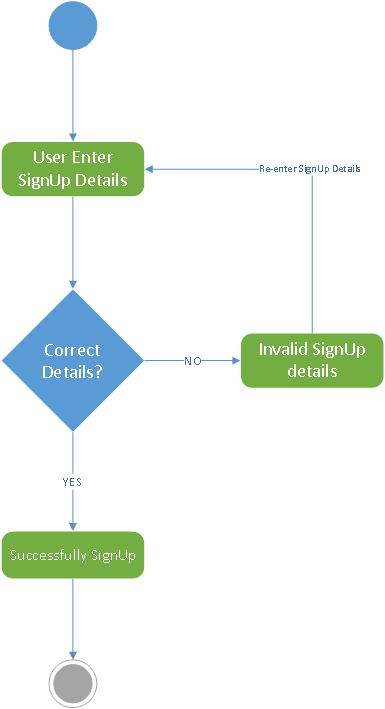
#### <Activity Diagram 1>

User with invalid information during login process with notify with invalid login attempt. On successful attempt he can see dashboard.



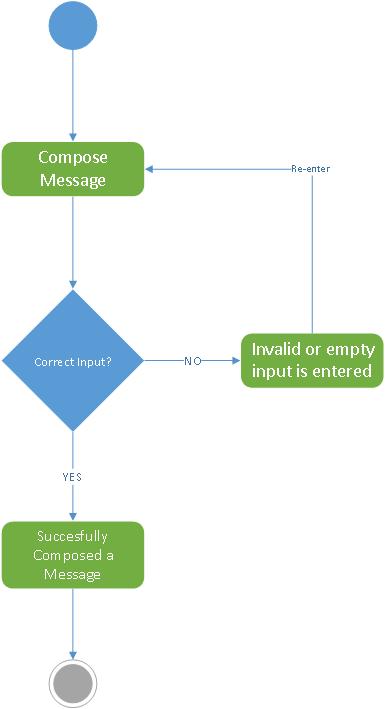
#### < Activity Diagram 2>

Invalid Signup process will result in validation error. On successful signup attempt user will remain on same page with successful signup notification.



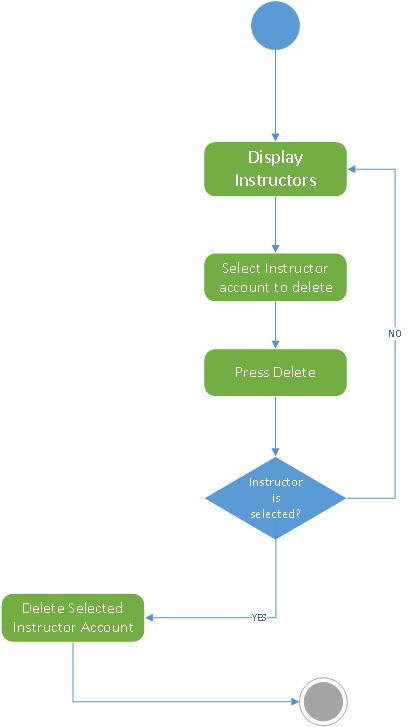
#### < Activity Diagram 3>

Invalid message compose process will result in validation error. On successful compose attempt user will redirected to inbox and able to see sent message in sent items.



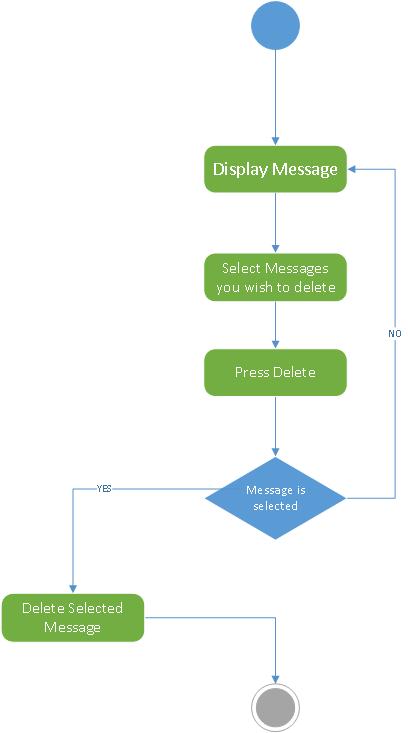
#### < Activity Diagram 3>

Administrator is privileged to remove instructor account. It is simple admin have to check instructor and press delete button



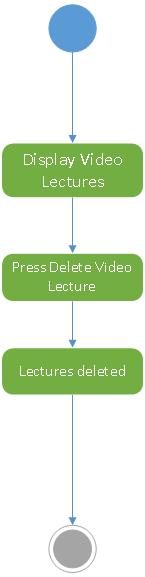
#### < Activity Diagram 4>

User(Admin, instructor or student) can display message and delete selected message from inbox.



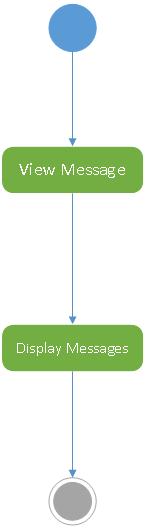
#### < Activity Diagram 5>

Instructors can delete messages with link button right next to upload lecture.



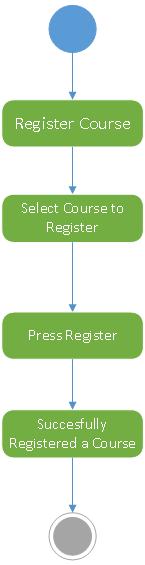
#### < Activity Diagram 6>

Anyone of user can display message(inbox)



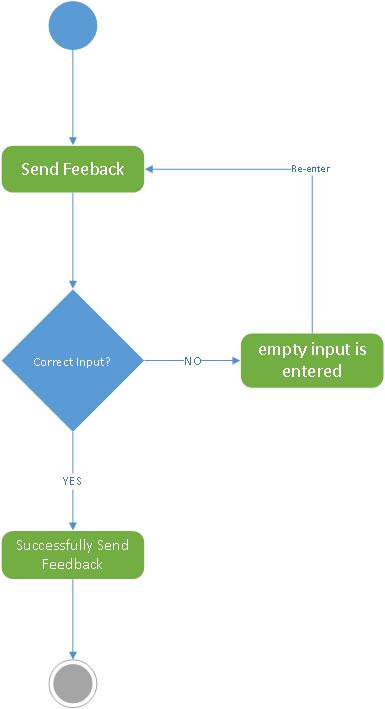
#### < Activity Diagram 7>

Students check courses which they wish to register. On successful registration they will get notification.



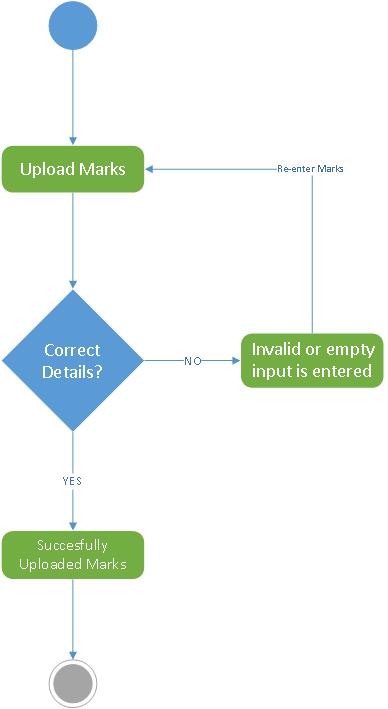
#### < Activity Diagram 8>

Students can send feedback. If he send invalid input a notification error will occur.



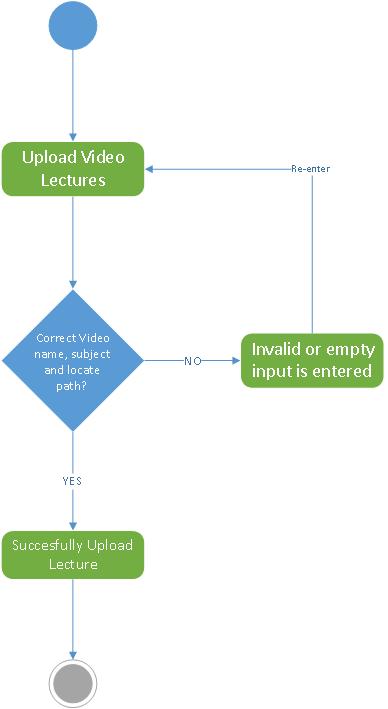
#### < Activity Diagram 9>

Instructor can upload marks out of 100. If less than 0 or greater than 100 validation error occurs.



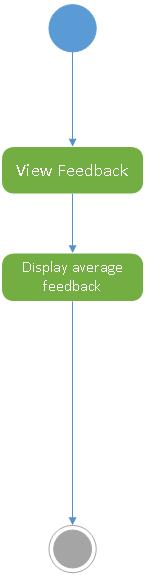
#### < Activity Diagram 10>

Instructor can upload video lectures. On successful attempt page will refreshed with newly added video



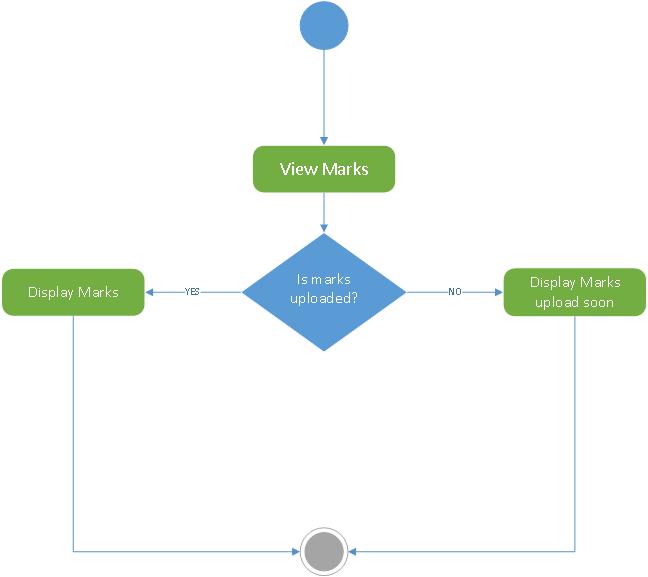
#### < Activity Diagram 11>

Instructor or admin can view feedback. Which will shown as average of all students.



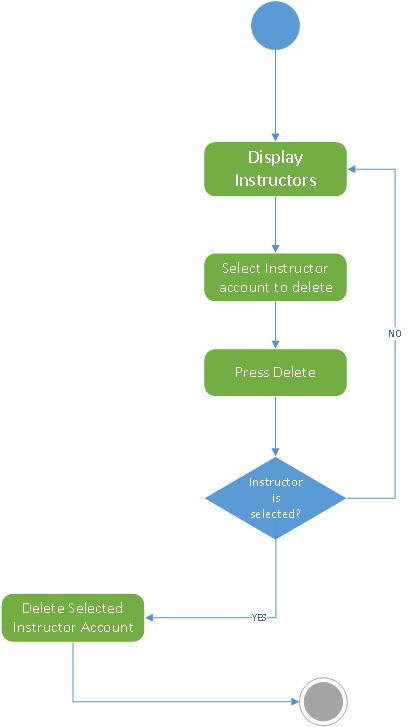
#### < Activity Diagram 12>

Students can view marks. When it is uploaded marks can be seen. If not then notify user with marks upload soon.

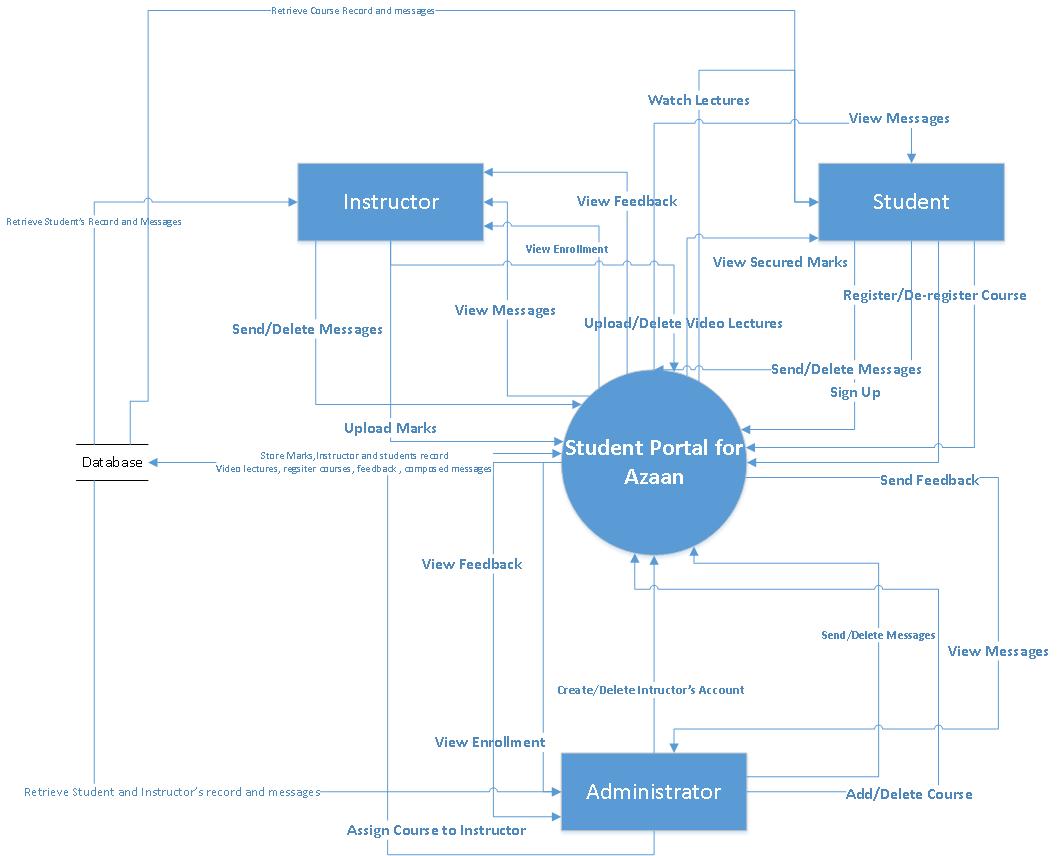


#### < Activity Diagram 13>

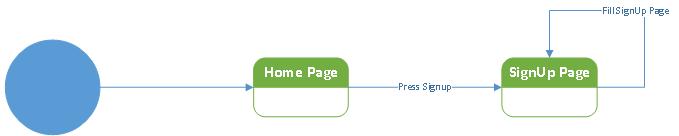
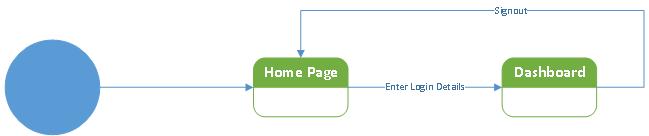
Instructor’s account is deleted by checking one of them and press delete button.



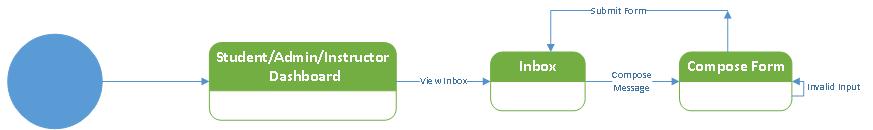
## Data Flow Diagram Level 1 Diagram



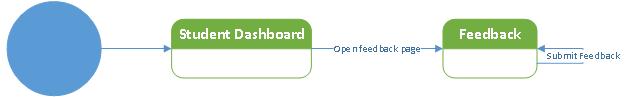
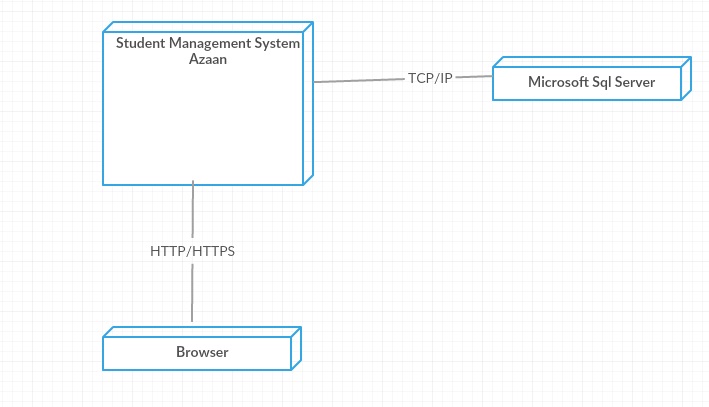
## State Transition diagram









**Deployment Diagram**

# GUIs of your Prototype/Software (not applicable)

# References (not applicable)

# Appendices(not applicable)