# CS-223
# SOFTWARE DESIGN DOCUMENT

## for

# Project 8
# Classroom Visualisation App-2

Prepared by:
Group-2
Harshit Agrawal - 160101030
Harshit Gupta - 160101031
Harshit Sharma - 160101032

March 15, 2018

# Contents

# List of Tables

# 1 Introduction

This Software Design Document provides documentation which will be used to aid in software development by providing the details for how the software should be built.It deals with transforming the customer requirements, as described in the SRS document, into a form (a set of documents) that is suitable for implementation in a programming language. Within this Software Design Document, are narrative and graphical documentation of the software design for the project including use case models,data flow diagrams, entity relationship diagrams and other supporting requirement information.

## 1.1 Purpose

The purpose of this document is to give a detailed description of the design for the Classroom visualization app and to provide a description of the design for a system fully enough to allow software development to proceed with an understanding of what is to be built and how it is expected to built.This document is intended for both the developers and students of the system

## 1.2 Document Conventions

Table 1.1: Document Conventions

| Term | Definition |
|---|---|
| Instructor | Person who shall be using the software for monitoring |
| Student | Person who shall be monitored by the instructor |
| Users | Collectively refers to the professors or teaching assistants |
| Device | An electronic device using which the instructor is delivering their lecture |
| Android | Android is a mobile operating system developed by Google |
| Google | Google is an American multinational technology company. |

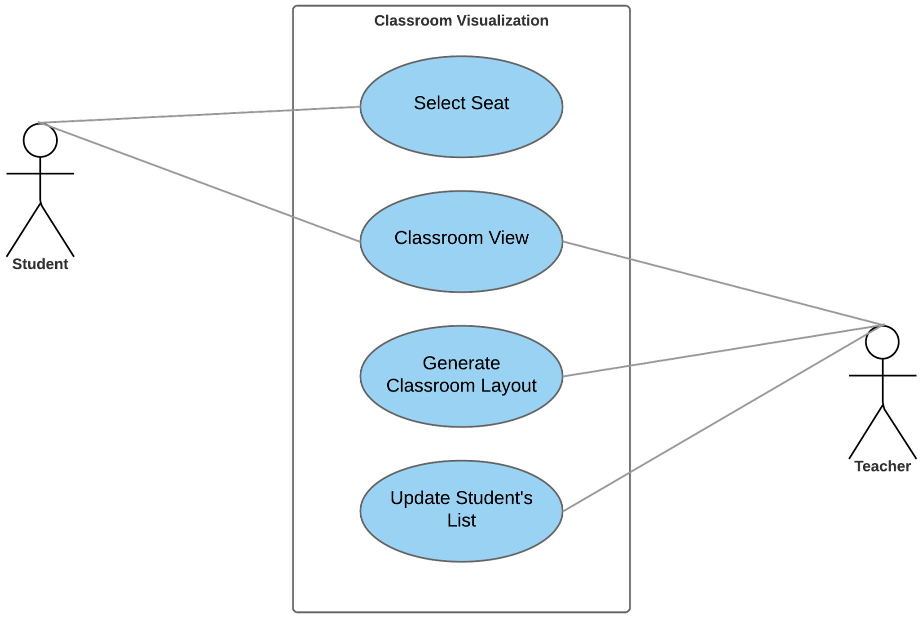
## 1.3 Project Scope

This software is meant to be deployed in an IT-enabled large classroom environment. This software shall allow the instructor to conveniently monitor the attention of their students in real time.This system will also help the teacher in taking attendance of the class.

## 1.4 References

- Tools for creating Data Flow diagrams :
  - https://www.smartdraw.com/
  - https://www.lucidchart.com/
- Other Sources for reference
  - R. S. Pressman, Software Engineering: A Practioner's Approach, 7th Ed., McGraw Hill, 2010.
  - http://nptel.ac.in/courses/Webcoursecontents/IIT%20Kharagpur/Soft%20Engg/New_index1.htm

# 2 Use Cases



## 2.1 U1: Select Seat

- **Actors :** Student
- **Precondition :** The user should be present in the class
- **Scenario 1:** Mainline Sequence:
  1. **System :** Enter your student ID
  2. **User :** Enters the student ID
  3. **System :** Select the seat
  4. **System :** Enters the seat number
  5. **System :** Display the seat number of the confirmed seat
- **Scenario 2:** At step 3 in mainline:

    3. **System :** Error message - “ The student details are incorrect “

- **Scenario 3:** At step 5 in mainline:
    5. **System :** : Error message - “ The seat selected is already occupied “

## 2.2 U2 : Classroom View

- **Actors :** Professor,Student
- **Precondition :** The classroom layout has been generated .
- **Scenario 1:** Mainline Sequence:
    1. **User :** Display the classroom status
    2. **System :** Classroom’s seat visualisation is displayed

## 2.3 U3: Classroom layout Generator

- **Actors :** Professor
- **Scenario 1 :** Mainline Sequence:
    1. **System :** Enter the number of rows of the classroom
    2. **User :** Enters the number of rows.
    3. **System :** Enter the number of columns
    4. **User :** Enters the number of columns
    5. **System :** Confirmation Message -” Layout generated “ .
    - **Scenario 2 :** At step 5 of mainline sequence:
        5. **System :** Error Message - “ Input Data is not correct “

## 2.4 U4: Update student list

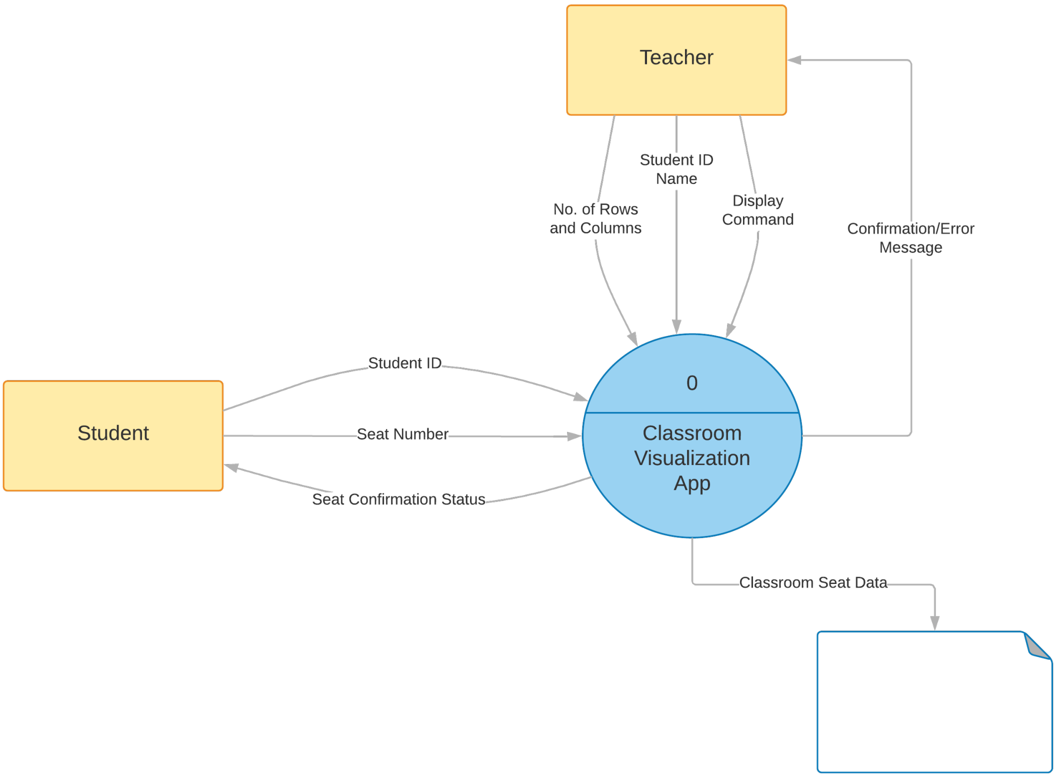
- **Actors :** Professor
- **Scenario 1 :** Mainline Sequence:
    1. **System :** Select among Add , Delete , Update
    2. **User :** Selects Add
    3. **System :** Enter Student ID and Student name
    4. **User :** Enters the Name , Student ID .
    5. **System :** Confirmation - “ Student has been added “.
- **Scenario 2 :** At step 4 of mainline sequence:

2. **User :** Selects Update
3. **System :** Enter Student ID
4. **User :** Student ID .
5. **System :** Enter the new details
6. **System :** Changed Information / Name
7. **User :** Confirmation Message

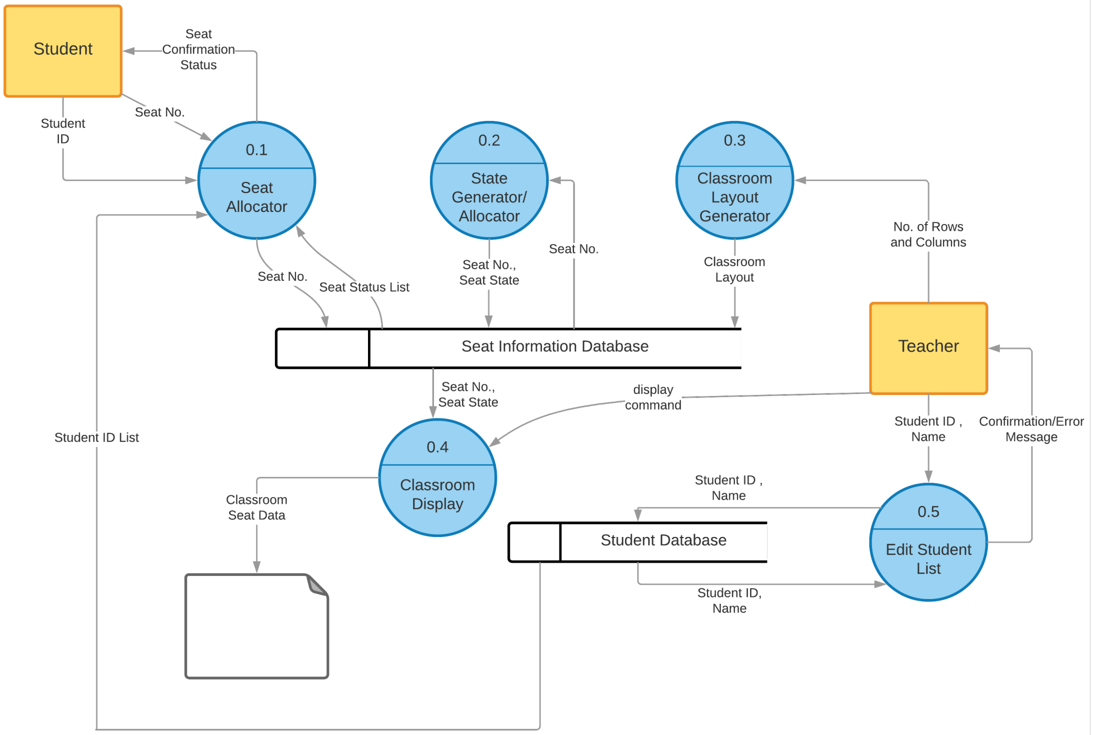
- **Scenario 3 :** At step 5 of mainline sequence:
    5. **System :** Error no such student found in the records

# 3 Data Flow diagram

## 3.1 Level 0 DFD



## 3.2 Level 1 DFD

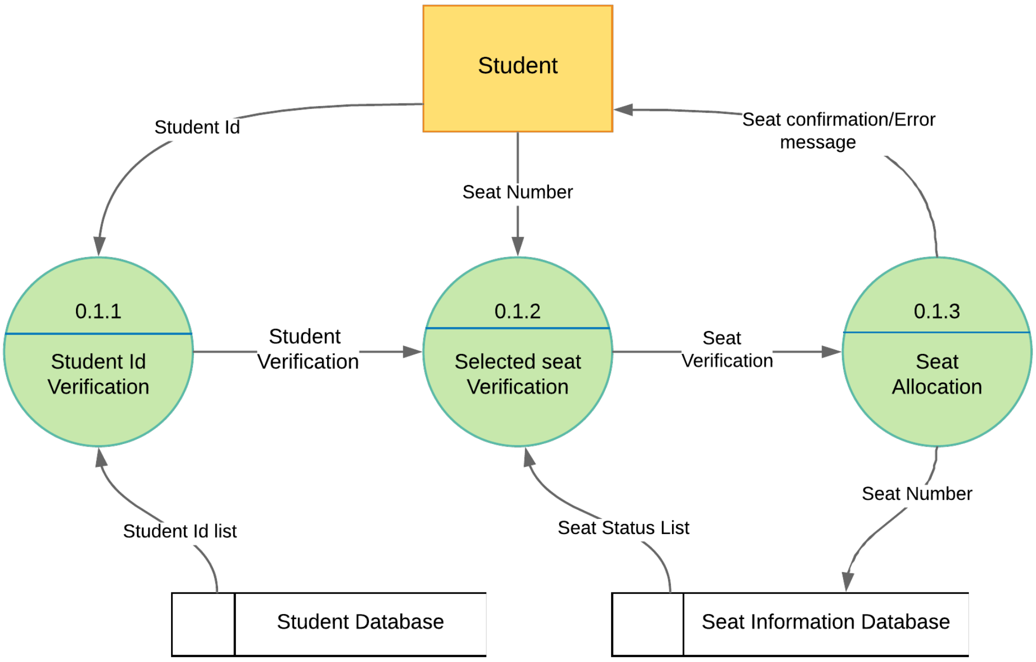


## 3.3 Data Dictionary for Level 0 and Level 1 DFD

- **Student ID** : [ integer ] /*Roll Number of the student if in the class*/
- **Seat Number** : [ integer ] /*Seat number of seats in the classroom*/
- **Seat Confirmation Status** : [“Seat is available and confirmed “, ”Seat is not available “,”Student Id is invalid”]
- **No. of Rows and Columns** : [ integer + integer] /* number of rows and columns of seats in the classroom */
- **display command** : [ boolean ] /* to display the classroom view or not */
- **Confirmation/Error Message** : [“ Student details are updated”,” details entered are invalid“]
- **Classroom Seat Data** : [ character 2d matrix ] /*each block of matrix represents seat number along with the state of student sitting on the seat*/
- **seat status list** : [boolean array] /*Seat status whether occupied or not*/
- **seat state** : [“Red Cross Mark“ ,”Blue Question Mark”, ”Green Tick Mark“]
- **Classroom layout** : [ character 2d matrix ] /* Creation of 2d matrix, each block representing seat*/
- **Name** : [string] /* name of the student */
- **Student id list** : [integer array] /* array containing student ids of all the students */

## 3.4 Level 2 DFD

### 3.4.1 Seat Allocator

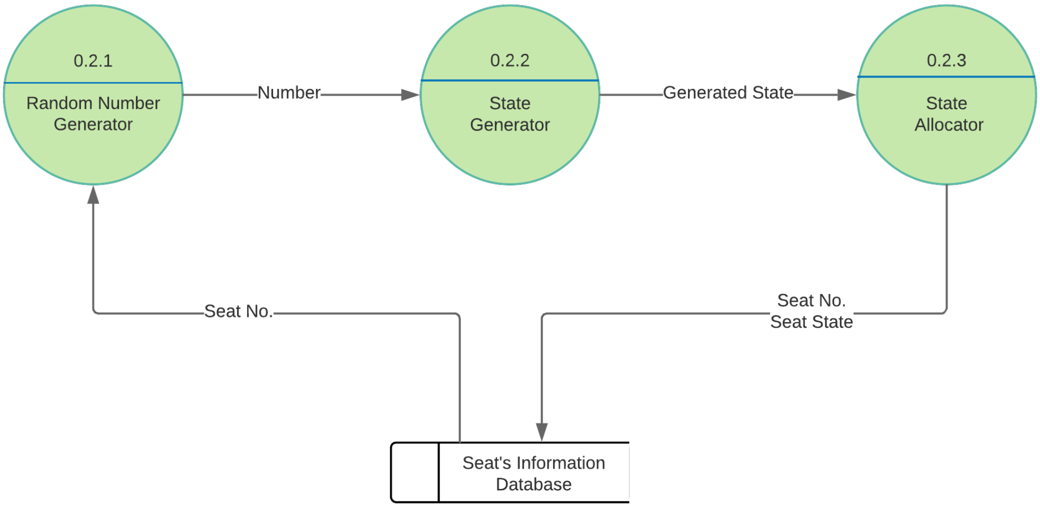


- Func 0.1.1 : Verifies if the student is enrolled in the course or not
- Func 0.1.2 : Verifies if the seat is occupied or not
- Func 0.1.3 : Allocates the state to the seat

ADDITION TO DATA DICTIONARY

- **Student Verification** : [boolean] /*Student id is valid or not */
- **Seat Verification** : [boolean] /*seat is occupied or not*/

### 3.4.2 State generator / Allocator

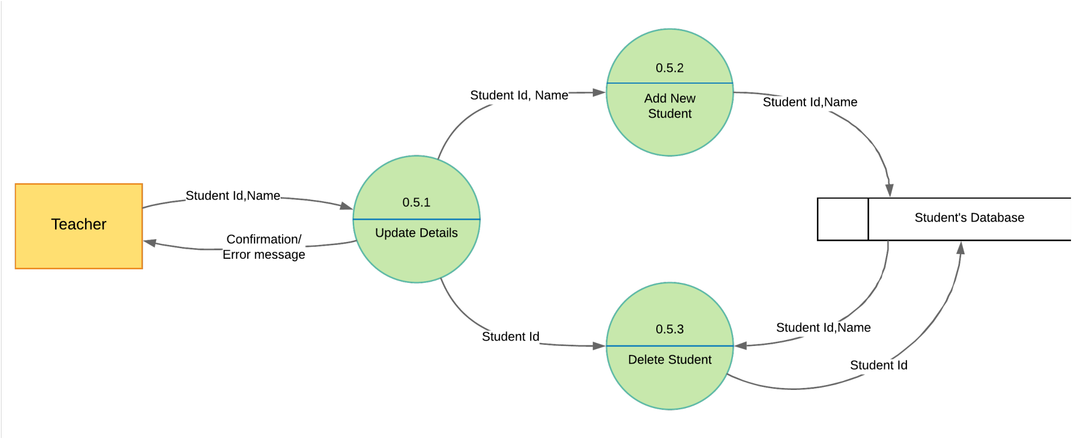


- Func 0.2.1 : Generates a random number
- Func 0.2.2 : Generates state according to the number generated
- Func 0.2.3 : Assigns the state to the seat

ADDITION TO DATA DICTIONARY

- **Number** : [integer] /*integer ranging from 1 to 10 */
- **Generated state** : [“Red Cross Mark“ ,”Blue Question Mark”, ”Green Tick Mark“]

### 3.4.3 Edit Student List

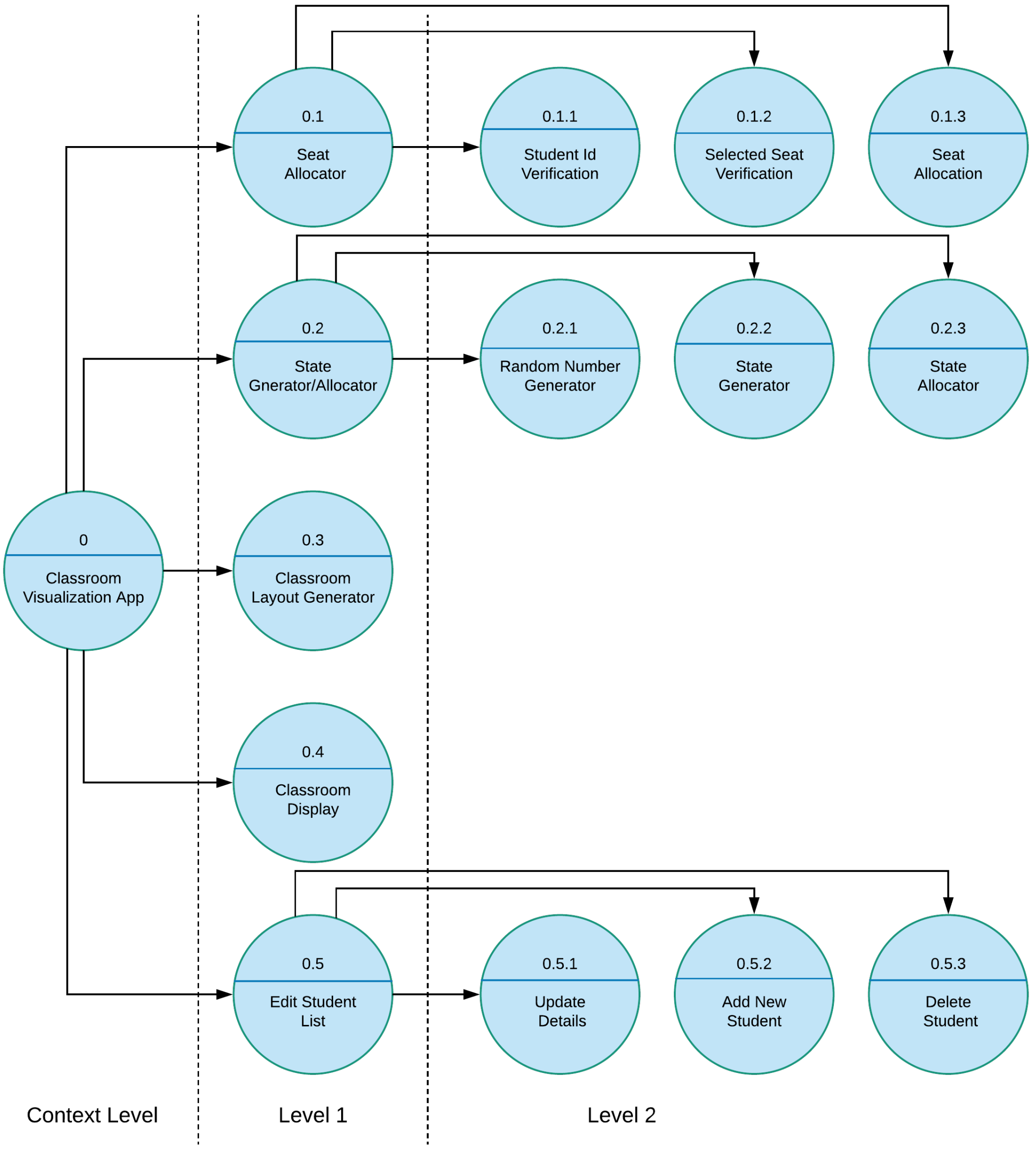


- Func 0.3.1 : Updates the Student database
- Func 0.3.2 : Adds a new entry to the Student database
- Func 0.3.3 : Removes a entry from the Student database

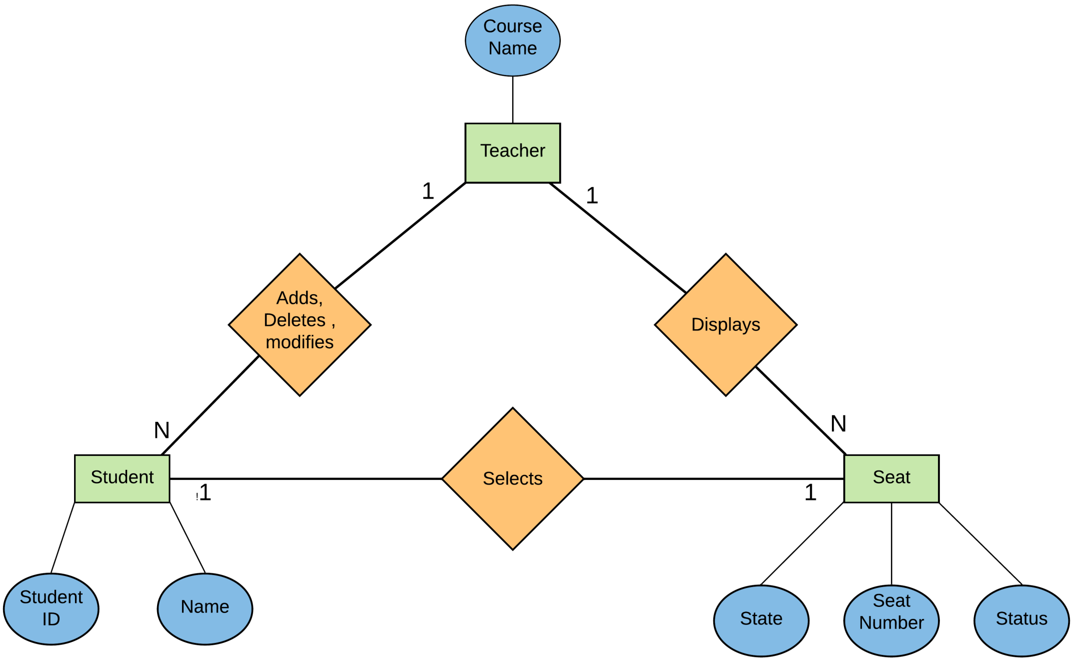
ADDITION TO DATA DICTIONARY

- **new student entry details** : [string+integer] /*name and roll number*/

# 4 Process Decomposition Diagram



# 5 ER Diagram


Course Name
Teacher
1
1
Adds, Deletes , modifies
Displays
N
N
Student
Selects
Seat
1
1
Student ID
Name
State
Seat Number
Status

# 6 Cohesion and Coupling

Cohesion is the measure of the functional strength of a module , whereas coupling is the measure of degree of interaction between two modules . The design of project exhibits low coupling between the modules ( level 1 functions) because the modules are functionally independent of each other . The functioning of one module does not depend upon the other module .

Also the design exhibits high cohesion ( Sequential Cohesion) . Inside a module the functions forms a part of sequence where output from one element of sequences input to the next . Modules also exhibit Communicational cohesion as they refer to the same database .