

Design Phase Document:

FOR

**STUDENT ACTIVITY DETECTOR AND ALERT
GENERATION APP FOR INSTRUCTOR**

VERSION 1.0

PREPARED BY –

MAYANK KUMAR AGRAWAL (150101033)

CHINMOY KACHARI (150101020)

SASWATA DE (150101058)

Date – 2/04/2017

Table of contents

1. Data Flow Diagram (DFD).....	1
1.1 Notation used.....	1
1.2 Context diagram	2
1.3 Level 0 Diagram	3
1.3.1 Level 1 Diagram of Student Processes.....	4
1.3.2 Level 1 Diagram of Instructor Processes	5
 2. Process Decomposition Diagram	 6
 3. Entity Relationship Diagram (ERD)	 7
 4. Data Dictionary	 8
4.1 Data Elements	8
4.2 Data Stores	13

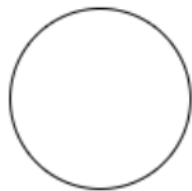
1.Data Flow Diagram (DFD):

These data flow diagrams (DFD) illustrate how data is processed by the system in terms of inputs and outputs. As its name indicates its focus is on the flow of information, where data comes from, where it goes and how it gets stored.

1.1 Notations used



External entity



Process



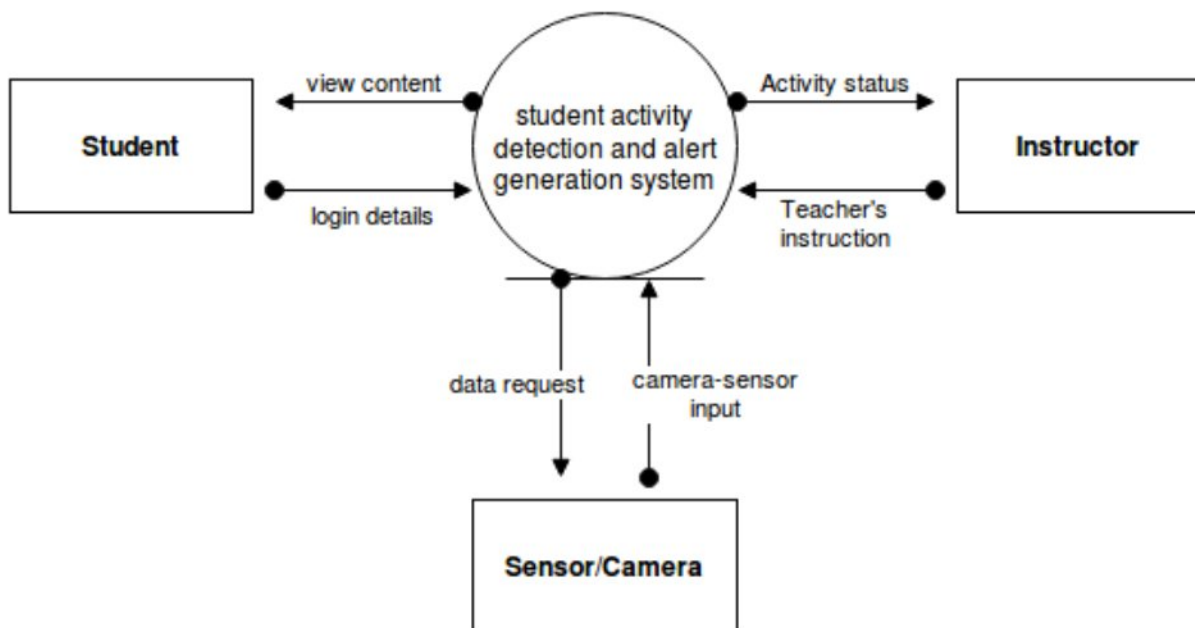
Data Storage



Data flow

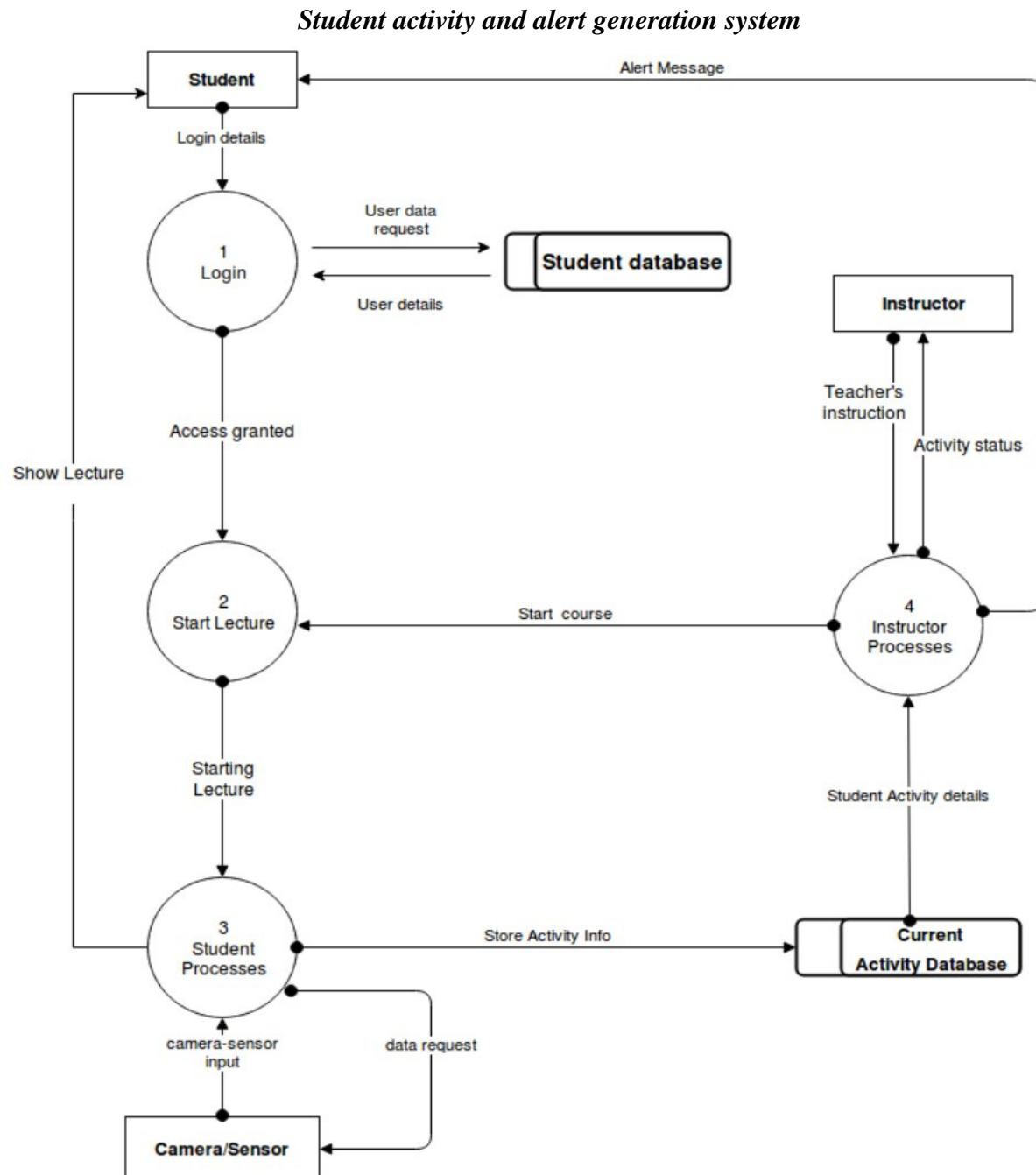
1.2 Context Diagram

The Context Diagram shows the system under consideration as a single high-level process and then shows the relationship that the system has with other external entities (systems, organizational groups, external data stores, etc.). Context Diagrams and Data-Flow Diagrams were created for systems analysis and design.



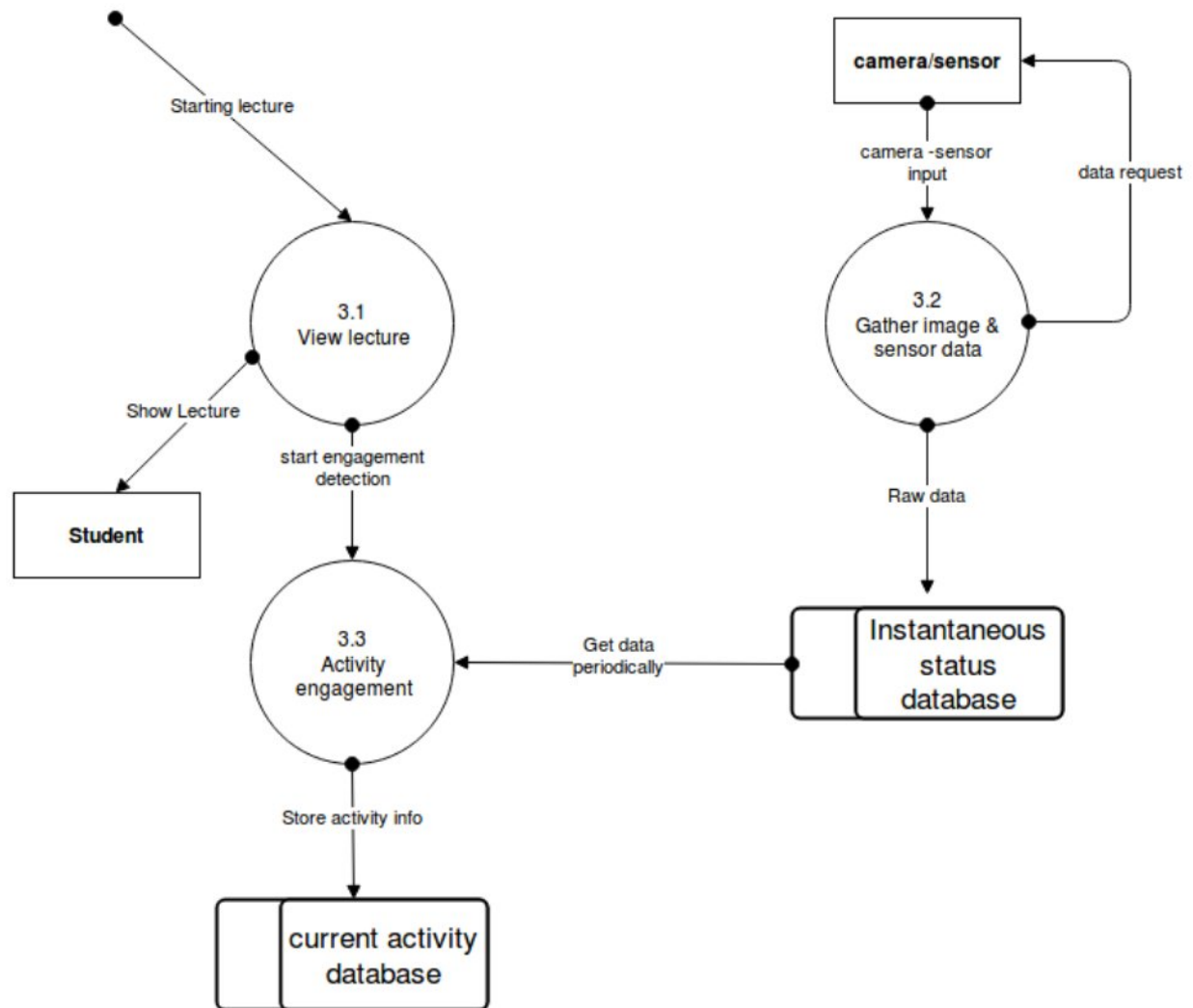
CONTEXT DIAGRAM

1.3 Level 0 diagram



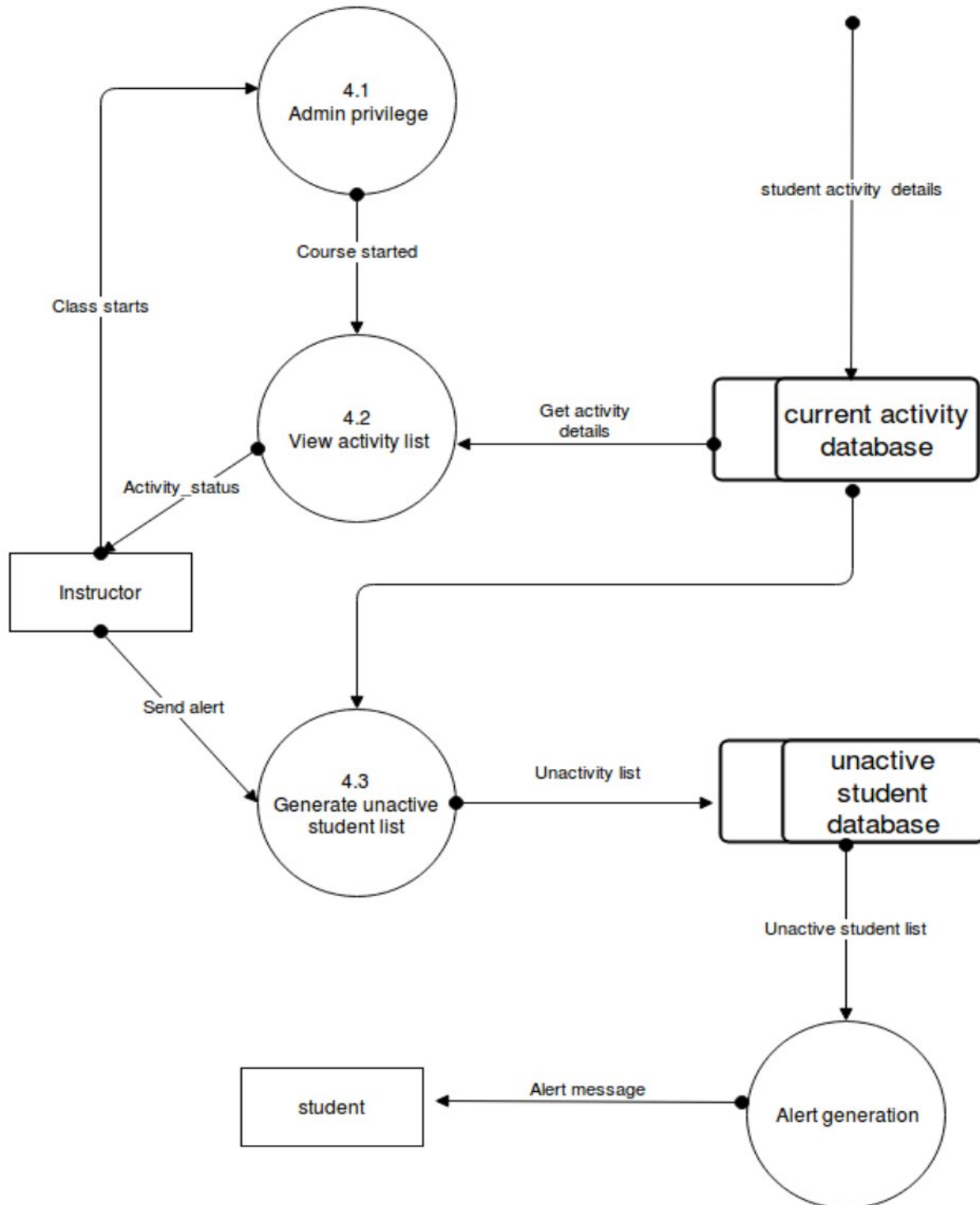
LEVEL 0

1.3.1 Level 1 diagram of *Student Processes*



LEVEL 1

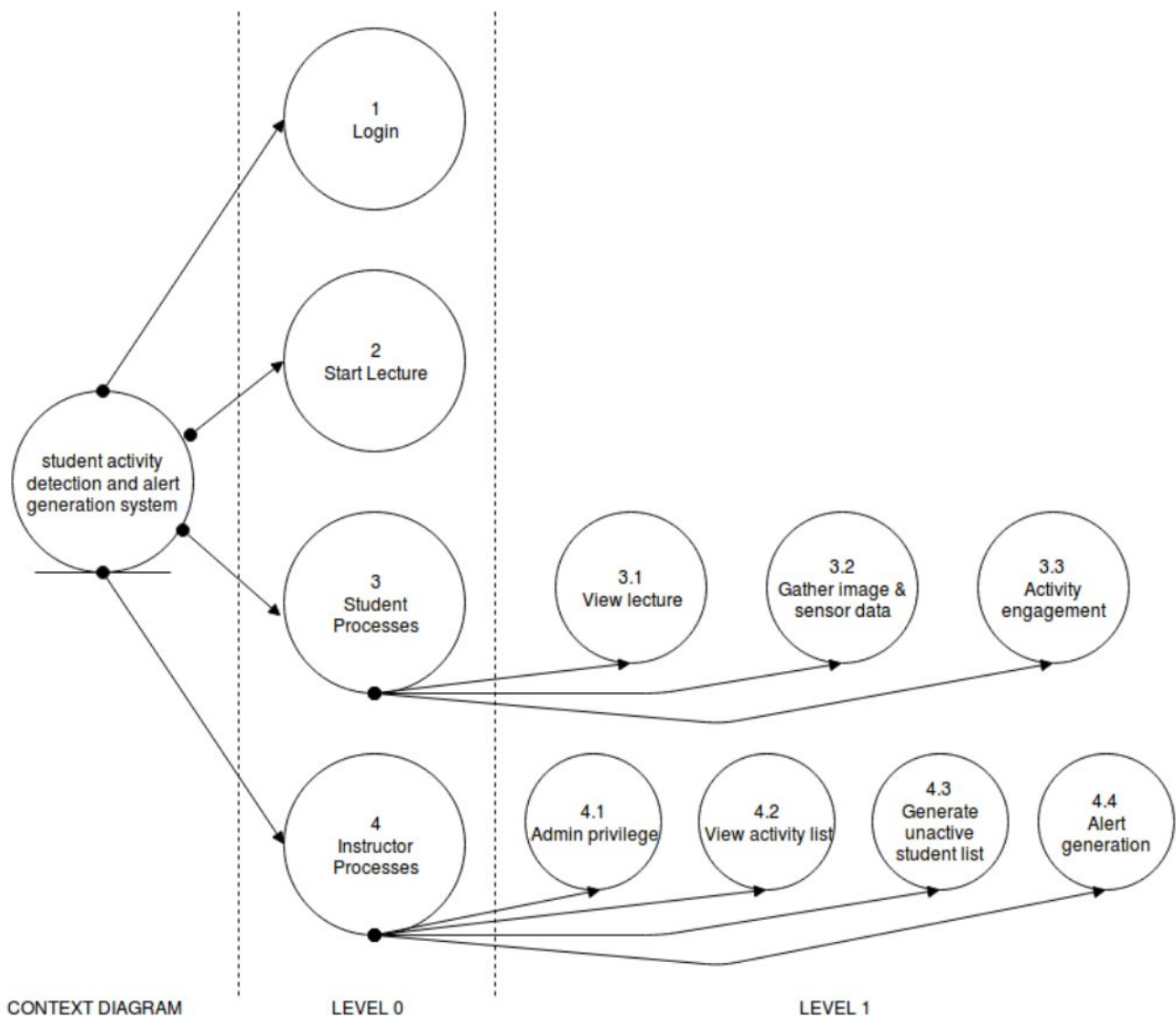
1.3.2 Level 1 diagram of *Instructor Processes*



LEVEL 1

2. Process Decomposition Diagram:

Decomposition is breaking a complex problem or system into parts that are easier to conceive, understand, program, and maintain. Structured analysis breaks down a software system from the system context level to system functions and data entities. The decomposition diagram shows a high-level function, process, organization, data subject area, or other type of object broken down into lower level, more detailed components.

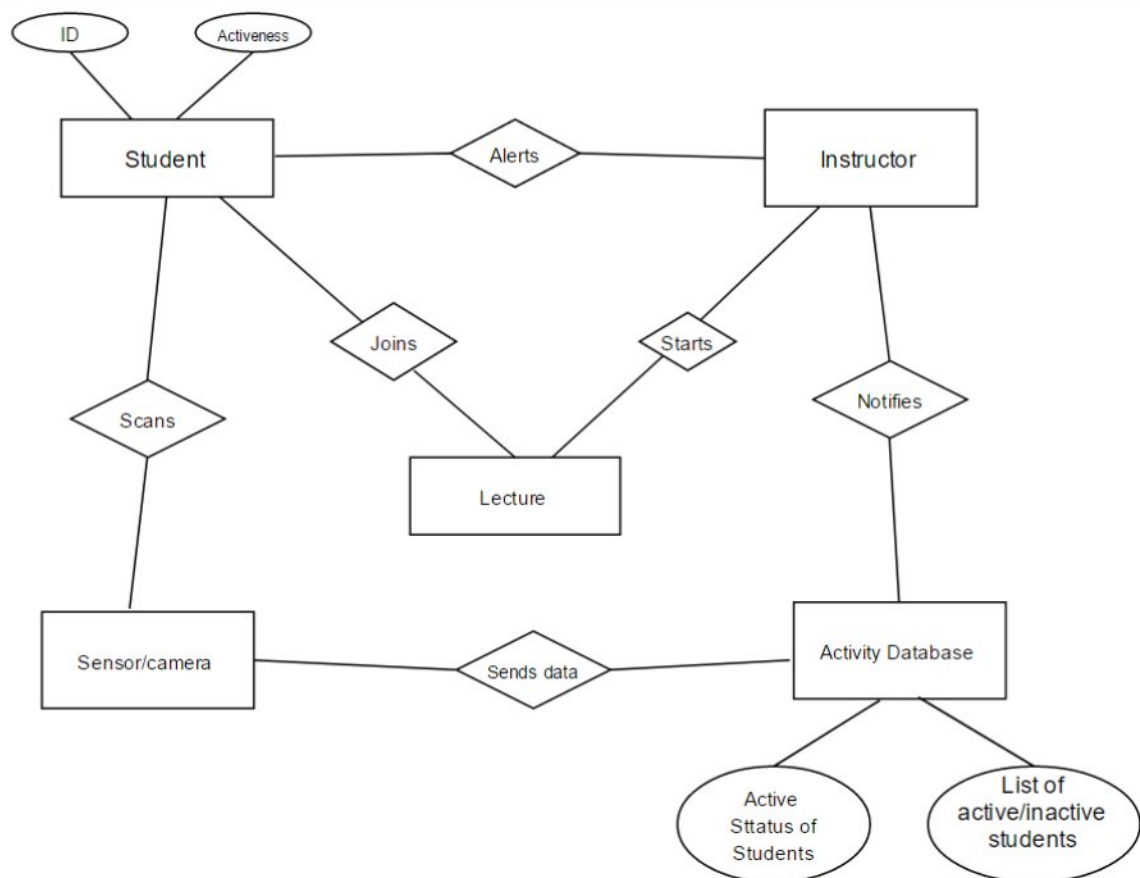


PROCESS DECOMPOSITION

3. Entity Relationship Diagram:

Entities are represented by means of rectangles. Rectangles are named with the entity set they represent. Attributes are the properties of entities. Attributes are represented by means of ellipses. Every ellipse represents one attribute and is directly connected to its entity (rectangle). If the attributes are composite, they are further divided in a tree like structure. That is, composite attributes are represented by ellipses that are connected with an ellipse. Relationships are represented by diamond-shaped box.

The entity relation diagram (ERD) depicts relationships between data objects.



4. Data Dictionary

4.1 Data Elements

A data dictionary lists all data items appearing in the DFD model of a system. The data items listed include all data flows and the contents of all data stores appearing on the DFDs in the DFD model of a system. A data dictionary lists the purpose of all data items and the definition of all composite data items in terms of their component data items.

Composite data items can be defined in terms of primitive data items using the following data definition operators:

+ : denotes composition of two data items

= : represents equivalence or composed of

[,] : represents selection, i.e. any one of the data items listed in the brackets can occur.

() : the contents inside the bracket represent optional data which may or may not appear.

{ } : represents iterative data definition

- **Name: View Content**

Where used/how used: Student activity detection and alert generation system(main process-Output).

Student (external entity- Input)

Description: View Content= [Show lecture + Alert message]

The main process as shown in context diagram returns the course content i.e. class lecture to student using the app which is then displayed on his smart phone.

Show lecture: When the student 'activity process' approves them the student is able to view course on his smart phone.

Alert message: Instructor sends a alert message to students.

- **Name: Login details**

Where used/how used: student activity detection and alert generation system(main process-Input)

Student (external entity- Output)

Description: Login detail refers to the input given by the student i.e. his username and password to the main process for entering into the course.

- **Name: Activity status**

Where used/how used: student activity detection and alert generation system(main process-output)

Instructor (External entity- Input)

Description: Main process returns activity status of each student to the instructor as soon as he starts the course.

- **Name: Teacher's instruction**

Where used/how used: student activity detection and alert generation system (main process-input)

Instructor (External entity- Output)

Description: Teacher's instruction = [Send alert + Class start]

Teacher can direct the app when to start the course and send alert to all inactive student at a given time if required.

Send alert: Instructor can send alert message to students.

Class start: Instructor's approval for starting the class.

- **Name: Data request**

Where used/how used: student activity detection and alert generation system (main process-output)

Camera-sensor (External entity- Input)

Description: Main process requests external entity to provide it with required data i.e. camera data and sensor data.

- **Name: Camera-sensor input**

Where used/how used: student activity detection and alert generation system (main process-input)

Camera-sensor (External entity- Output)

Description: Camera and sensor timely provide their data to the processes whenever required.

- **Name: Access granted**

Where used/how used: Login (Level 0-output)

Start lecture (Level 0-input)

Description: Once a succesful login is made login function permits to load the app further to next function which is ‘start lecture’.

- **Name: Start course**

Where used/how used: Instructor processes (Level 0-output)

Start lecture (Level 0-intput)

Description: After the conformation for starting the lecture from instructor is received the ‘start lecture’ directs to ‘Student processes’.

- **Name: Store activity info.**

Where used/how used: Student processes (Level 0-output)

Current activity database (Level 0 -input)

Description: ‘Student processes’ detects the activity of each student and then stores it periodically in the ‘Current activity database’.

- **Name: User data request**

Where used/how used: Login (Level 0-output)

Student database (Level 0 -input)

Description: ‘Login process’ checks the username and password input by student by matching it with the data present in the database.

- **Name: Start engagement detection**

Where used/how used: View lecture (Level 0-output)

Activity engagement (Level 0 -input)

Description: As soon as the ‘view lecture’ process starts it directs the ‘activity engagement’ process to start detecting the activity.

- **Name: Raw data**

Where used/how used: Gather image & sensor data (Level 0-output)

Instantaneous status database (Level 0 -input)

Description: Once the ‘Gather images and sensor data’ receives data from external entity it periodically stores that data temporarily in the database.

- **Name: Get data periodically**

Where used/how used: Instantaneous status database (Level 0-output)

Activity engagement (Level 0 -input)

Description: ‘Activity engagement’ fetches data periodically from the database.

- **Name: Get activity details**

Where used/how used: current activity database (Level 0-output),

View activity list (Level 0 -input)

Description: ‘View activity’ list process fetches information on activity status of each student from the database which is then shown to the instructor.

- **Name: Course started**

Where used/how used: Admin privilege (Level 0-output),

View activity list (Level 0 -input)

Description: When the instructor uses ‘Admin privilege’ process to start the course, from that time he automatically starts on getting the activity list of students.

- **Name: Unactivity list**

Where used/how used: unactive student database (Level 0-input),

Generate unactive student list (Level 0 -output)

Description: The app has option to list all the inactive student at a particular instant of time by using ‘Generate unactive student list’ process and then storing it in database.

- **Name: Unactive student list**

Where used/how used: unactive student database (Level 0-output)

Alert generation (Level 0 -output)

Description: Instructor has the choice to send on-time notification to all inactive students by using the data stored in ‘unactive student database’

4.2 Data Stores

STUDENT DATABASE		
Description: This database stores login details of user i.e. name and password		
Attributes	Description	Type
Username	Name of the user who is enrolled for the course	String
Password	Password of the user (in encrypted form)	String

INSTANTANEOUS STATUS DATABASE		
Description: It stores the data that needs to be processed immediately for finding the instantaneous activity status using proximity sensor and camera data		
Attributes	Description	Type
Proximity Sensor data	It contains the distance value found by proximity sensor	integer
Camera data	Stores the images periodically captured by camera	image

CURRENT ACTIVITY DATABASE		
Description: It stores the periodic activity status of all the students of the class up to current time		
Attributes	Description	Type
Student details	Details of all the students logged in the course	String
Activity status	Present activity state of each student(active/inactive)	Boolean