Indian Institute of Technology Guwahati

CS243 : Software Engeneering Lab

Under the guidance of : Prof. Samit Bhattacharya

Project 1 : An app to detect student activity and alert generation for the instructor

Software Design Document

Group 7 :

- Mukul Verma , 150101038
- Piyush Jain , 150101046
- Shubhanshu Verma , 150101073

Date : April 2, 2017

Contents

1. Introduction

1.1 Purpose

This software design document describes the architecture and system design of the app. It gives a complete structure of designing of the project which will surely act as a bridge between the Software Specification Report and the Coding stage.

1.2 Scope

The project is scoped to develop an student activity detection and alert generation application. Using the features like orientation of the device, distance between user and device and movement/shaking frequency of the device, the app will determine the degree of focus an student is putting in.Using these features as input, the app is aimed to give the instructor

- an alert whether a particular student is engaged with the lecture or doing something else.
- the number of students who are engaged in the lecture and the number of students who are not.

1.3 Overview

This document first covers the architectural design and decomposiition description of the system. The document then explains the data design and interface design of the system. The design is in a functional approach.

1.4 Assumptions and acronyms

- There are two kinds of user - the students and the instructor.
- The document follows the following coordinate system for describing the design architecture and data.

2. System Overview

The system takes readings from different sensors of the device , processes it in usable form and then checks on some conditions to decide the engagement of the user (student). Accordingly, it informs the instructor if there is any student who is not engaged to the lecture. The app may give two levels of alert w.r.t. each student, say mild and severe. There will be two sets of conditions, one strong and one weak to check the engagement level. If the student(user) violates the strong conditions, the instructor will get a mild alert for him/her. It the students violates the weaker conditions as well, the instructor will get a severe alert for him/her. Accordingly, the instructor can take action. If the student minimizes the app, this will also be directly counted as violation. Finally, at the end of the lecture, the system sends the number of students who are engaged and the number of students who are not, to the instructor.

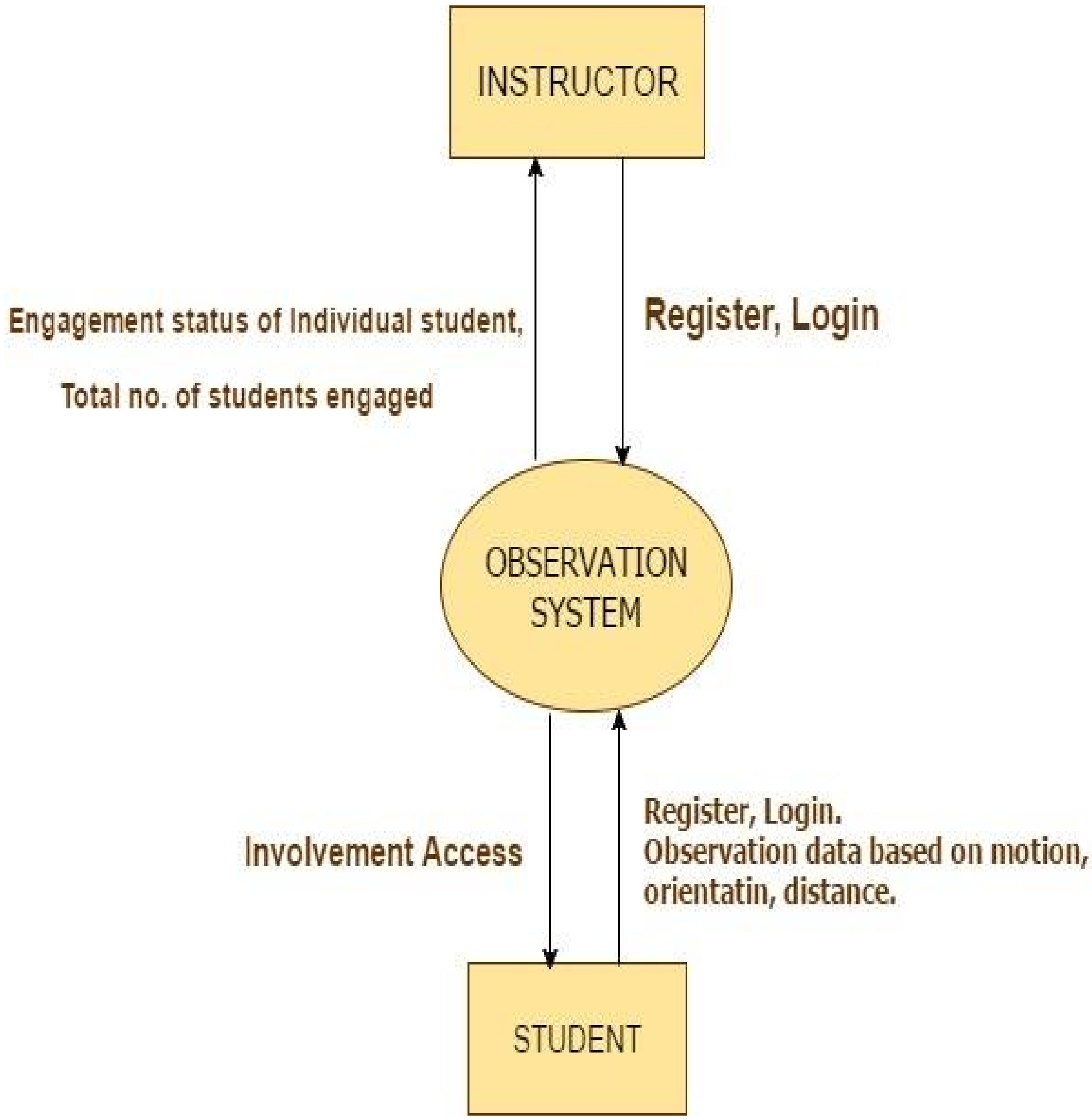
3. System Architecture

3.1 Data Flow Diagram

The overall functionality and the architecture of the system can be understood by the following diagrams.

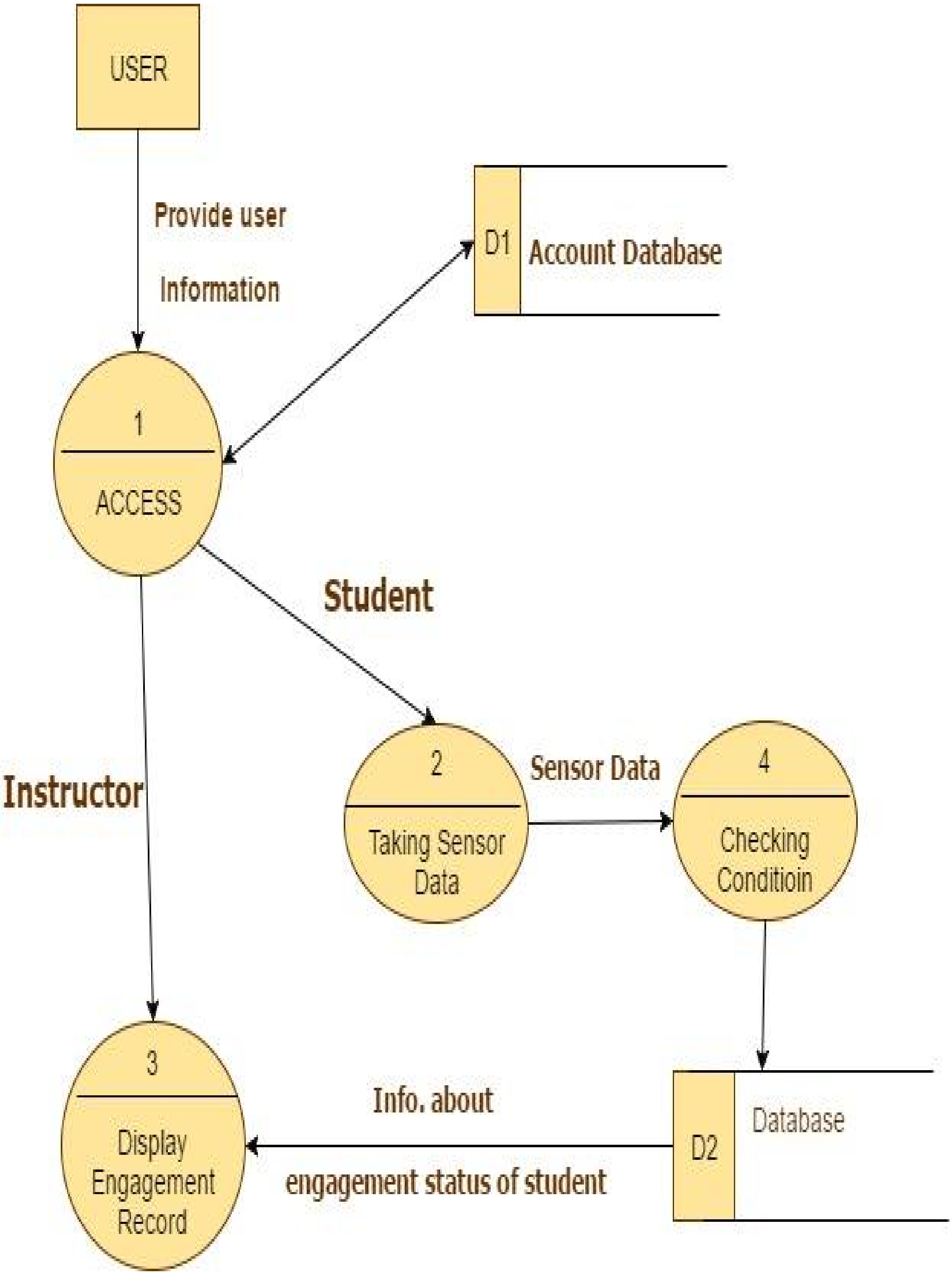
3.1.1 Context Level DFD

The following diagram shows the context level of the system.



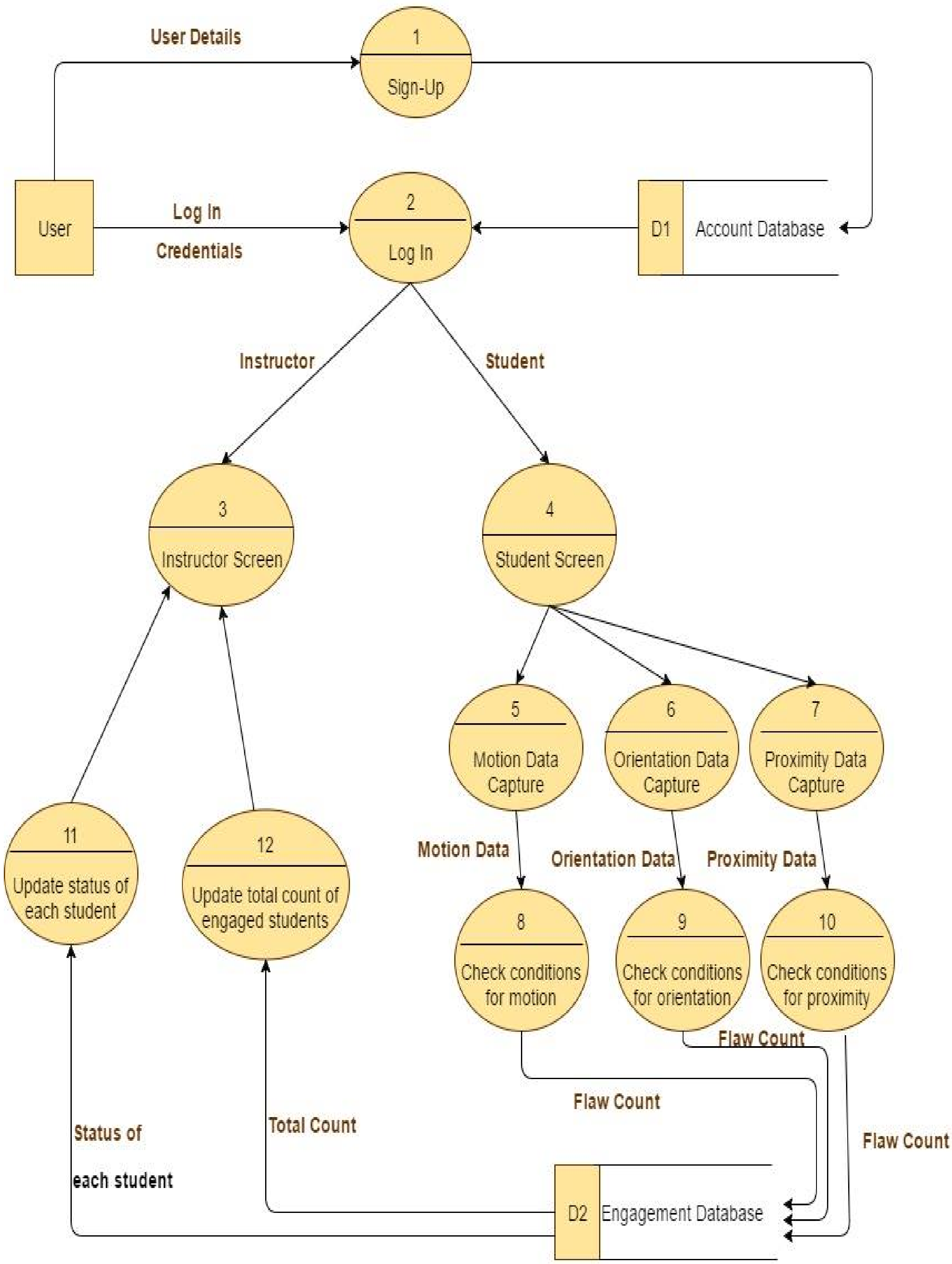
3.1.2 Level 0 DFD

The following diagram well explains the Level 0 Data Flow.

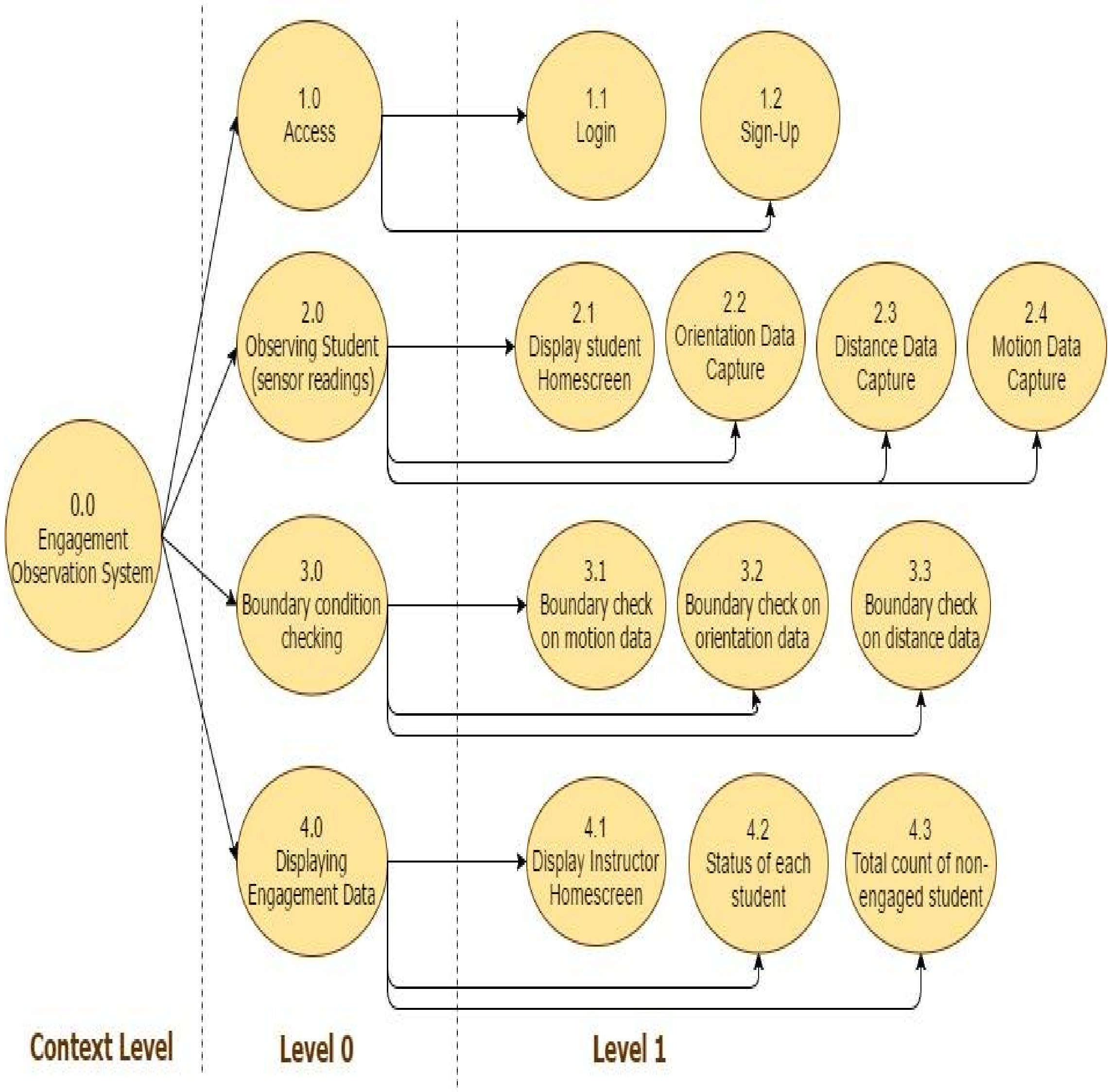


3.1.3 Level 1 DFD

The following diagram well explains the Level 1 Data Flow.



3.2 Process Decomposition Diagram



3.3 Design Rationale

The functional design approach has been chosen to describe the architecture because in this project,it is easier to explain the architecture in terms of functions rather than entities.

4. Data Design

4.1 Data Description

There will be an authentication database which will conatain the login credentials of all the users. This database will be updated on SignUp (new user registration). Then, there will be another database which will contain the threshold/violation conditions in the form of simple dictionary whose values may be floats of integers. The threshold conditions will be decided by analyzing the ideal conditions of orientation, motion and proximity of the device. Other than that, there will be temporary buffer data which will be the readings of the sensors.

4.2 Data Dictionary

The app may produce any of the two levels levels of alert for any student to the instructor. The engagement status will be updated per minute on the instructor's screen as per the following rule -

- Formula for final flaw_count = ((mild_flaws *1) + (severe_flaws X 2))
- If final flaw_count is greater than equal to 4 and less than equal to 7 , it will be considered as a mild alert.
- If final flaw_count is greater than 7, it will be considered as a severe alert.

Finding mild_flaws and severe_flaws : /newline

- **Distance between device and user**

  Type - integer

  SENSOR USED: PROXIMITY SENSOR

  Allowed range : Distance of user from smartphone should be in the range 20cm to 40cm.

  Working Idea : Generate an alert (increase count of variable which keeps track of this condition) when condition not satisfied

  - Mild Flaw (a variable keeps count of mild flaws) If distance of user from device exceeds limit for 3-6 seconds, increase count of mild flaws.
  - Severe Flaw (a variable keeps count of mild flaws) If distance of user from device exceeds limit for more than 6 seconds, increase count of mild flaws.

- OVERALL (a variable keeps count of total no. of times distance is out of range).
- The observation is done on per minute basis. So all the variables are initialized to 0 after every minute.

- **Shaking Frequency**

  Definition of shake - Shake is observed by observing rate of change of acceleration. If (sum of changes in accelerations in x, y, z directions)/(dT*10000) is greater than 250, then the event is called shake.

  dT: small change in time between events.

  Time is observed in order of milliseconds. Type - integer

  SENSOR USED: Accelerometer

  Allowed Value : Should be less than or equal to 2 shakes per minute.

  Working Idea : Generate an alert (increase count of variable which keeps track of this condition) when condition not satisfied

  - Mild Flaw (a variable keeps count of mild flaws) - If number of shakes per minute is between 2 to 4 (inclusive), increase count of mild flaws.
  - Severe Flaw (a variable keeps count of mild flaws) - If number of shakes per minute is greater than 4, increase count of severe flaws.
  - The observation is done on per minute basis. So all the variables are initialized to 0 after every minute.

- **Linear Movement**

  Type - integer

  SENSOR USED: Accelerometer

  Definition of linear movement - Movement is observed by change in acceleration values. If (sum of changes in accelerations in any two directions x, y,z directions) greater than 1.5 or 2, then the event is called linear movement.

  (dx + dy greater than 1.5 or dz + dy greater than 1.5 or dx + dz greater than 2)

  Allowed Value : The way we are observing movement, the no. of movement counts per 10 seconds should be less than or equal to 15.

  Working Idea : Generate an alert (increase count of variable which keeps track of this condition) when condition not satisfied

  - Mild Flaw (a variable keeps count of mild flaws) - If no. of move counts per 10s are between 15 to 25.

- Severe Flaw (a variable keeps count of mild flaws) - If no. of move counts per 10s is greater than 25.
- OVERALL(a variable keeps track of move counts per minute)

  Total move counts per minute greater than 90 but less than 120 (mild alert)

  Total move counts per minute greater than 120 (strong alert)
- The observation is done on per 10s basis. So all the variablaes are initialized to 0 after 10s (except overall observation which is initialized to zero per minute).

- **Orientation**

  Type - integer

  SENSOR USED: Accelerometer, Magnetic_Field Sensor (if avalable)

  Obtaining Orientation : obtaining orientation using only accelerometer, a low-pass filter can be used to isolate the force of gravity. Further computation can be done using basic trigonometry.

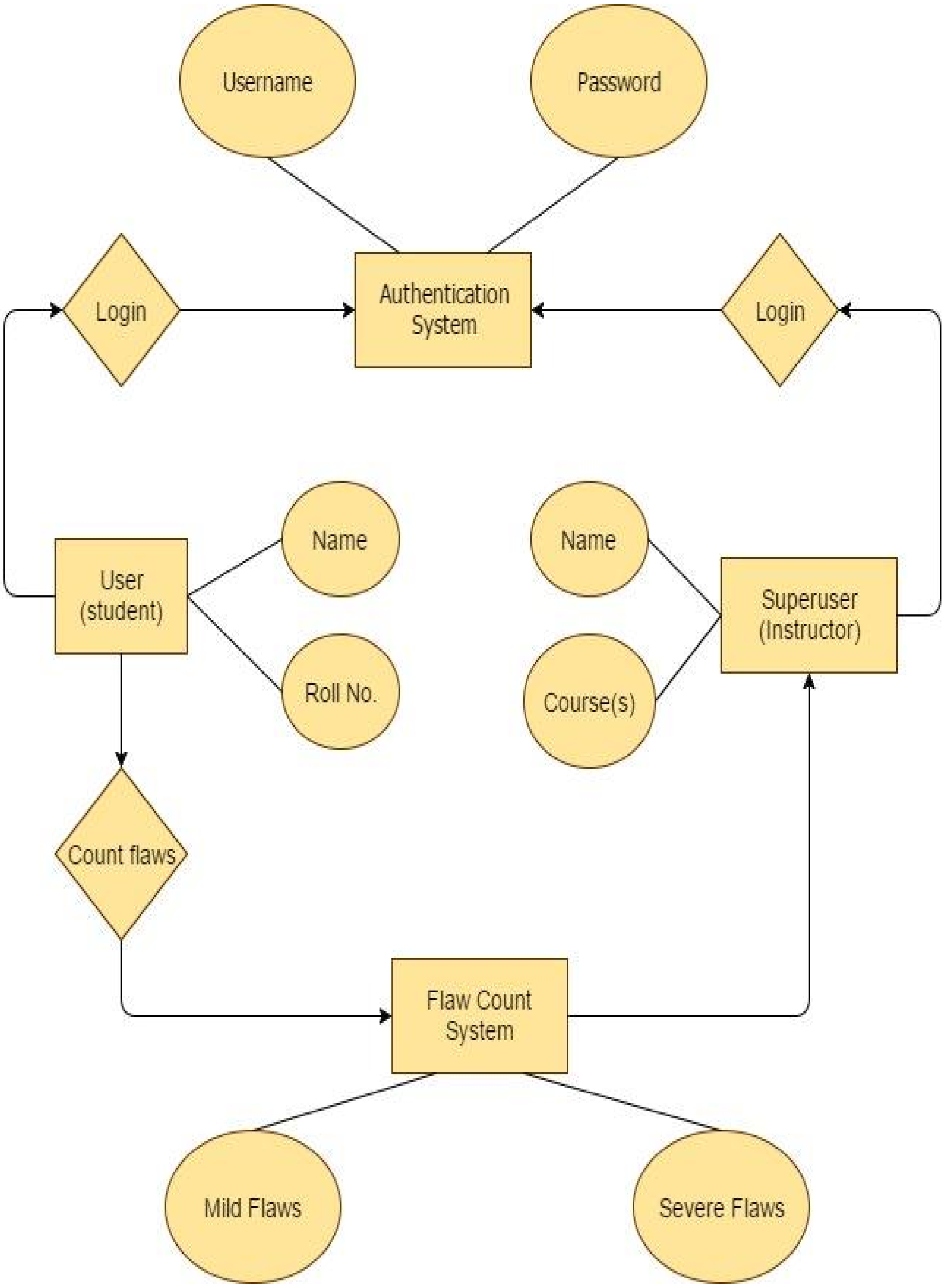
  If magnetic field sensor is available, then orientation of device is obtained using values of magnetic field sensor and accelerometer and using predefined functions.
  If not, then the value of component of gravity along x, y, z axes is obtained from accelerometer and these values of gravity are used to obtain orientation of device.

  Allowed Value : If mode is potrait, absolute Value angle of Y-axis from surface plane should be between 30 degrees and 60 degrees. If mode is landscape, absolute Value angle of X-axis from surface plane should be between 30 degrees and 60 degrees.
  Working Idea : If ideal range is exceeded for a certain time period, generate alert.

  - Mild Flaw (a variable keeps count of mild flaws) - If violation period between 5 to 10 seconds
  - Severe Flaw (a variable keeps count of severe flaws) - If violation period greater than 10 seconds
  - If violation period is greater than 15 second continuously, increment severe alert by 4. this is to ensure a direct severe alert is generated if the student has constantly put the phone in unallowed position.

4.3 Entity-Relationship Diagram

The following diagram shows the entity-relationship model of the system.



5. Human Interface Design

5.1 Overview

The first screen will be a login/signup page. After successful login, if the user is student, he will be redirected to a student homepage(dummy), if the user is instructor, he will be directed to an instructor homepage(dummy). For the students, this will be the point where the student's activities will be monitored. Accordingly, alert will be generated.

5.2 Screen images

These are some screenshots of the dummy pages of the app.