SOFTWARE REQUIREMENTS SPECIFICATION:

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 – 19/02/2017

Table of contents

1. Introduction

1.1 Purpose

The purpose of this document is to provide a detailed description of the “student activity detector and alert generation for instructor” app. It will explain the purpose and features of the app, what the app will do, the constraints under which it must operate and how the app will react to external stimuli. This document is intended for both the instructor and the students of the IT-enabled classroom.

1.2 Scope

The app will be a monitoring aid for the instructor of a large IT-enabled classroom. The app will be designed to automatically detect the attentiveness of the students and alert the instructor accordingly. In a classroom with large number of students it is difficult for the instructor to monitor whether the students are paying attention to the class or not. The app will constantly monitor the activities of the students via their smartphones (front camera) and hence detect if a student is being attentive or not. The app will notify the instructor if a particular student is not paying attention in the class.

1.3 References

IEEE standard template for SRS.

https://en.wikipedia.org/wiki/Software_requirements_specification

2. Overall Description

2.1 Product Perspective

This product is a part of larger system “IT-enabled smart classroom system”, which can automatically detect the attentiveness of all the students in the classroom and alert the instructor accordingly. This app uses front camera and sensor data to detect the on-time engagement of student i.e. whether a student is watching the lecture slide or sleeping (or looking somewhere else) during the course. Then, this data *is fed* to main *app (*mentioned above) to use it in the instructor’s interface for showing the *state (*active/inactive) of each student.

2.2 Product Functions

Major functions which the app will perform are as follows:

2.2.1 App will regularly refresh the state (active/inactive) of each student after every certain amount of time which is then shown on instructors interface.

2.2.2 Provides detail analysis of activity of each student to the instructor, after the class hour is over so that he can take required/necessary action against the student or use it to give attendance.

2.2.3 Instructor can send on-time alert notification to those student who are not attentive during the class, so that students could start paying attention thereafter.

2.3 User Classes and Characteristics

This app is mostly applicable for any student-instructor community. The app will have a very minimalistic or basic interface so that it could be used even for primary level school students to college students which could also be a part of digital literacy programs intended to educate students located in remote areas. It requires user to have only basic knowledge of using android smartphone, thus previous use experience is not compulsory.

2.4 Operating Environment

This app will operate on android smartphones having working front camera and proximity sensor. Should work on android version 4.4 KitKat and higher. Local area connectivity is also required for server purpose.

2.5 Assumptions and Dependencies

The app is completely independent and self-contained.

2.5.1 The assumption of ideal condition for engagement is:

Head/eye movement frequency of user should be <= 2/min. and the distance between user & device should be < 25cm for being in active state.

2.5.2 Phone should be charged enough to carry on the function throughout the course timings.

2.5.3 There will be institute provided username and password for each student and each student is allowed to use his/her smartphone only.

2.5.4 Regulatory policies: App shall not access any personal information of users.

2.5.5 Device should allow the use of different sensor's data, image captured by front camera, response to be send over network or Wi-Fi connection and permission for receiving push notification.

3. Specific requirement

3.1 Functional requirement

This section defines the fundamental actions that must take place in the software in accepting and processing the inputs and in processing and generating the outputs. These are listed below:

3.1.1 Login:

Input: username and login password.

Output: Access to the account is permitted or error message will be displayed.

Description: Users will only be able to login with valid login details. The system shall authenticate login credentials (username and password). If the user's login credentials fail authentication, the system shall display a message to the user indicating that authentication has failed.

3.1.2 Start_course:

Input: choose the ongoing classroom course.

Output: selected course starts according to the start timing of course.

Description: student has to choose the course which is currently being followed by the instructor in the class, to start the lecture.

3.1.3 Student_engagement:

Input: N/A

Output: active/inactive status of student.

Description: Function shall use face movement function and face distance function to detect (based on assumptions made) whether a student is thoroughly

engaged in class or not. As a result the output will be updated in user's and instructor's database which will be reflected in their interface.

3.1.3.1 Face_distance:

Input: N/A

Output: proximity sensor data of distance between student's face from the screen.

Description: When there is a request from student_engagement function, function shall return the proximity sensor data of distance between student's face from the screen.

3.1.3.2 Face_movement:

Input: Images captured by front camera

Output: Function shall return the count value and shall update it regularly.

Description: Function shall use images captured by front camera (at specific interval of time) as input to process and then count/keep-track of number of face/eye movement per minute.

3.1.4 Active_student_count:

Input: active/inactive status of student.

Output: Return the total number of students engaged in the class.

Description: This function shall use the data from student_engagement function to count the total number of active students in the class during the course.

3.1.5 Alert_notification:

Input: Instructor's permission.

Output: warning notification to inactive students.

Description: Function shall list all the students who are not engaged (by using student_engagement function data) and then send warning notification to those students for whom the status is inactive.

3.1.6 Active_duration_count:

Input: active/inactive status of student.

Output: Return the total time for which student was in active state.

Description: This function shall use the data from student_engagement function to count the total time for which a student was in active state during the lecture.

3.1.7 Exit:

Input: N/A

Output: Exit from the course and return active_duration_count.

Description: when a student exits from the lecture, this function returns the active_duration_count of a student to the instructor.

4. Non-functional requirements

4.1 Maintainability:

- The app shall be updated from time to time for various bug fixes and improvements.
- The app shall take feedback and review from the users which will help in pointing out and fixing various issues of the app for the next update.

4.2 Scalability:

- The app can be implemented in college as well as school level.
- The app can also be used in various online courses where the lectures are streamed live over the internet.

4.3 Reliability:

- While the lecture is going on there must be no interruption in teaching and learning. Hence, the app must be highly reliable in broadcasting any information to the instructor and students.
- There must be minimum delay in sending alert to the instructor regarding student activity.

4.4 Security:

- Only students registered for the particular course will be able to log in the app to participate in the class.
- Protection of the software from accidental or malicious access, use, modification, destruction, or disclosure.
- End-to-end encryption and keeping the log history of each user.

4.5 Extensibility:

- The app shall be able to extend to other platforms like web for laptops.