CS-242/243

Project - #9

Smart Warning System

Software Requirements Specification (SRS) Document

Course Instructor -

Prof. Samit Bhattacharya

Team - #23

Jatin Goyal - 160101036

Namit Kumar - 160101046

Nitin Kedia - 160101048

Table of Contents:

1. Introduction

1.1 Purpose

The objective of this document is to present a detailed description of the mobile app *Smart Warning System*. It will illustrate the purpose and features of the system. It will also explain system constraints, interfaces and dependencies. This document is intended for both the developers and users of the system.

1.2 Scope

The application will be a mobile app, on the Android platform which implements an intelligent and robust warning system to enhance student engagement during lectures by continuous monitoring and feedback. It will facilitate a more meaningful and interactive teaching sessions.
The app is being developed as a project to gain practical experience in the practices of software engineering, as a part of the course of same name offered by Department of CSE, IIT Guwahati.

1.3 Intended Audience

The app is primarily targeted for usage by students, faculty and administration in educational institutions. NGOs, Bootcamps will also benefit from the use of the app.

1.4 Document Conventions

1.4.1 Definitions

- Image Matrix: The data of a image obtained from camera.
- API level : A measure of the version of Android OS running.
  Eg.- API level 25 for Android 7.0 Nougat

1.4.2 Acronyms and Abbreviations

- **SRS** - Software Requirement and Specifications
- **SDK** - Software Development Kit
- **NDK** - Native Development Kit
- **ReST** - Representational State Transfer
- **API** - Application Programming Interface
- **MP** - Mega Pixel
- **FPS-** Frames Per Second
- **Res** - Screen Resolutions
- **App** - Application

1.5 References

IEEE, IEEE Std. 830-1998 IEEE Recommended Practice for Software Requirements, IEEE Computer Society, 1998

2. Overall Description

This section provides a base for the requirements which will be defined in the following section(s) and therefore gives an overall view of the specific requirements.

2.1 Product Perspective

This product is a teaching app on the Android platform which let's the students and the teachers analyze and maintain records of students' performance in every class. This product allows the teacher to know the attentiveness of particular student and also the students to improve their performance and attentiveness during the lecture. These attributes of students are decided on the basis of their class participation, attentiveness etc.

2.2 Product Functions

Functions included in the final product will be as follows :

1. User Requests
    a. Start Session
    b. Attendance
    c. Class Current Status
2. In-class Alerts
    a. Alert Generator
    b. Send Alert
    c. Alert Response
3. Post-class Action
    a. Post-class Report
    b. Meetup Alert
    c. Performance Discussion

2.3 User Characteristics

The intended user for the product will have the following characteristics :

1. Able to use a touch based Android device with the specified OS version.
2. Able to understand the functioning and operation of the software on a basic level.
3. Able to understand the English Language to operate the application.
4. Belonging to the age group of 16+ years.

2.4 Assumptions and Dependencies

Assumptions :

1. OS used on device has version Android 4.4 or higher.
2. The user device must have a functioning front camera with minimum 5.0 MP capacity.
3. Device must have a minimum storage of 100 MB for storing app data.
4. Device must have a working microphone to record sound. It is at a suitable unobtrusive distance from the user such that the input voice is as noise-free as possible.

Dependencies :

1. Android Camera and Microphone API with basic camera operation capabilities.
2. Wifi/Internet Connection in every classroom for the users to connect to a main server.

3. External Interfaces

This section contains the specific requirements that the software need to operate in required manner along with the required features.

3.1 Hardware interfaces

Following are the inputs and outputs associated with the application through which it interacts with the external environment.

- **Inputs**:
  1. Touch-Screen Tap/Drag/Pinch Input from the Android Device.
  2. The Image matrix captured from the front camera, with minimum resolution of 720p and camera of minimum 5.0 MP
  3. Microphone of the Android Device with clear sound quality.
  4. Access to External storage device.
  5. Android Device Clock which provides current time in 24-hr format.

- **Outputs**
  1. Display Screen of the Android Device
  2. Speakers of the Android Device
  3. Vibration Motor of the Android Device

3.2 User Interface

This product uses the following software - user interface :

1. This product can be used on any Android Device running Android OS above the minimum required version.
2. The software will be GUI based and touch screen, capacitive buttons of the android device will be used to communicate with the user.
3. This product will have two types of users, the teachers and the students and therefore there will be two different User Registration page which will have form-based interface.
4. The choice of feature being used will use Menu-Based interface for letting the user select the desired feature. Feature set for different users will be different.

3.3 Communications Interfaces

Local server will be connected to all the devices through a WiFi. All the data collected from the students' devices will be analyzed and accordingly the alerts or pop-ups will be sent to both the teacher and the student through the local server.

4. Functional Requirements

4.1 User Requests

- **Input**: Touchscreen Tap input to invoke a specific request from User’s Device, Corresponding Time from Device Clock, User ID.
- **Output**: A Message to user’s device containing information requested by user.
- **Description**: This is a group of student-specific and teacher-specific functions such as attendance, class report etc.

4.1.1 Start Session

• **Input**:

a. Touchscreen tap input “Start” (from Teacher’s device)
b. Text input for Course ID (from Teacher’s device)
c. Teacher ID

• **Output**:

a. Communication of all devices with local server starts
b. Session Start Time (Saved)
c. “Class has started” alert to all users connected to room Wi-fi

• **Description**: With the teacher’s consent, the local server initialises a class session and begins collecting data from students’ devices. Thereby one cycle of the Alert System kicks-in.

4.1.2 Attendance

• **Input**:

a. Touch-Screen Tap Input for attendance in Student’s device
b. Time of the tap (from User’s device clock)
c. Session Start Time from server storage
d. User ID

• **Output**: Boolean Presence- True/False for the User ID for current session, Attendance Marked/Not Marked message to student.

• **Description:** To record the attendance, every student has to tap a button while they are connected to the classroom WiFi. The main server only accepts attendance when the Tap time is within 15 minutes from session start time.

4.1.3 Class Current Status

•**Input**: Touch-Screen Tap Input from teacher's Android Device, states of all students in the class (from server database).
•**Output:** Image matrix of a Pie Chart or Bar Graph, list of students in best state and worst state.
•**Description:** This function is used by the teacher to see the current status of all students in form of a pie chart or a Bar Graph. On reception of such request the server generates the graph based on last known states of all students present in the class.

4.2 In-Class Alerts

- **Inputs**: User ID, Current and all previous states of the user, Respective Time of Capture
- **Output**: Alert notification on user's device, Boolean- Responsive/Nonresponsive
- **Description**: This function is responsible for all dynamic processing of the states of a user, generating alerts, sending them to corresponding user and capturing their response.

4.2.1 Alert Generator

• **Input**:

a. User ID (from user app)
b. Previous states of the user with capture time (server storage)
c. Last captured state of user with capture time (User device)

• **Output**:

a. Alert Message
b. Alert Priority (Timer value)
c. Alert Type (Screen Flash/Sound/Vibration)

• **Description**: The central server receives given inputs from a user at fixed intervals and analyses this and previous states of the same user, determines if the user is involved or distracted and saves an appropriate alert to send by "Send Alert". Alert priority is high when state has below a critical level for some time.

4.2.2 Send Alert

•**Input**:

a. User to whom alert is to be sent (Server Database storage)
b. Alphanumeric alert message, type, priority (Server Database storage)

•**Output:**

a. Text notification (pop-up) on the concerned users device
b. Sound/Vibration on the concerned users device.
c. Timer on the notification to act upon it.

•**Description:** The server sends an instruction to the user app to whom alert is to be sent over the common Wi-Fi. Based on the received message the user app displays a device notification.The alert type dictates a sound or vibration while the alert priority dictates the timer.

4.2.3 Alert Response

•**Input**: Touch-Screen Tap Input on notification from the user device, Time remaining on the timer of that notification.
•**Output:** Boolean- The alert has been acted on by the student- True/False.
•**Description:** This function waits for the student to acknowledge that he has been alerted during the class within the permitted time and the true/false value of response is saved by the server. If it hasn't been acknowledged then the server inserts it to a blacklist.

4.3 Post-class Action

- **Input**: List of unresponsive alerts in class, Touch Screen tap for "End Class" by Teacher's device.
- **Output**: Meetup Alerts, Email of remarks after discussion.
- **Description**: This module is responsible to ensure that proper actions on alerts sent during class. This is done as:

4.3.1 Post-class Report

- **Input**:
  a. List of unresponsive alerts during class, User ID of all students in this blacklist (from server database).
  b. Touch-Screen Tap Input from teacher's Device.
- **Output**: List to teacher containing names of such students.
- **Description:** After the class ends, the teacher ends the session and the server sends the professor a list containing information of all students with unresponsive alerts.

4.3.2 Meetup Alert

- **Input:**
  a. List of unresponsive alerts during class, User ID of all students in this blacklist (from server database).
  b. Touch-Screen Tap Input from teacher's device.
- **Output**: Alert to all above students to meet the teacher.
- **Description:** When the teacher ends the session, all the students in the blacklist will be sent an alert to meet the teacher immediately.

4.3.3 Performance Discussion

- **Input:**
  a. Touch-Screen Tap Input from teacher's device.
  b. Text input from teacher's device for each student
- **Output**: Email to corresponding student containing remark.
- **Description:** When the student meets the teacher after the class, teacher may leave a remark and also ensure whether the student met him or not. Email will be sent to all the students with the remark.

5. Non Functional Requirements

5.1 Reliability

The software might fail when the WiFi connection is disrupted by large number of active connections to the router. Bandwidth is not as much of a limitation as the number of connections.
It might also fail when a user tries to run multiple features in succession as the Camera API takes a lot of processing & memory. The system shall allow users to restart and reconnect the application after failure with the loss of at most the previous 1 minute of all students state data.

5.2 Availability

The system will be available for use whenever the user (Teacher) deems necessary in extended class hours with the teacher and students present in a classroom with (Wifi) Access to main server.

5.3 Security

The system will use a login system for both the teacher and the students for authentication and thus will be highly secure and will prevent any type of unauthorized access to private contents like remarks, attendance etc.

5.4 Maintainability

The system will be updatable from software patches available through the Google Play Store. Any discrepancies will be addressable by any developer as the coding will be done according to the coding standards of IEEE.

5.5 Portability

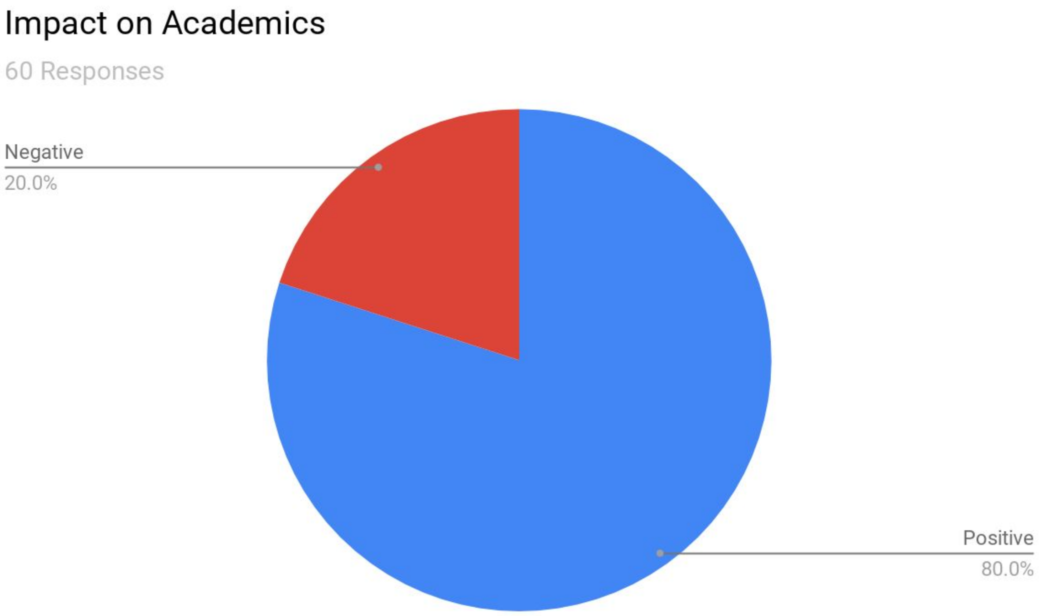
The software will be easily transferable to any Android device satisfying the minimum software dependency requirements as specified in this SRS Document. The software can be installed on an Android device using the same method as any other Android App via the Android App Manager.

5.6 Contextual Enquiry

5.6.1 Survey

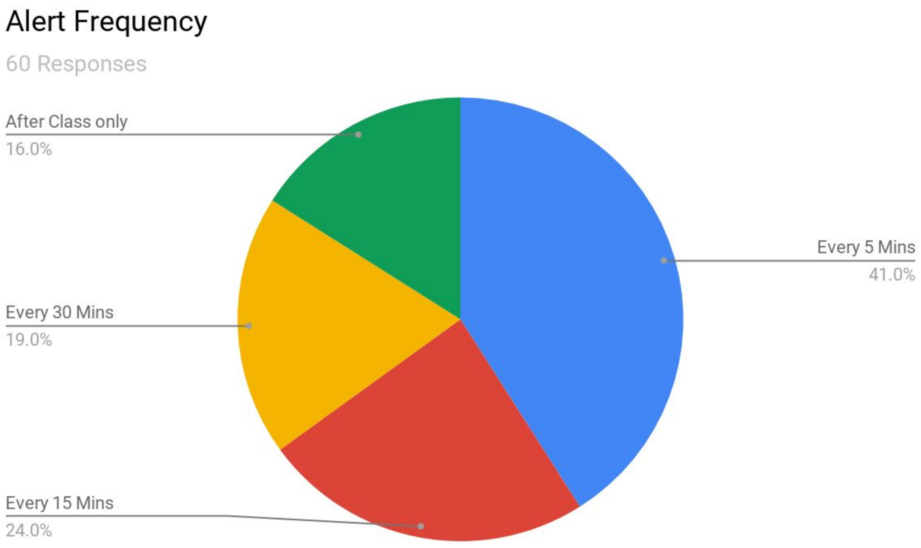
A survey was conducted among students studying at IIT Guwahati. They represent the student user type of our app. For a complete picture, some of our professors and TA's in different courses were enquired to represent the faculty user type.
The questionnaire starts with a brief summary of the app and proceeds to ask a mix of one-liner and option based questions. Some questions are different for students and faculties.

Q1. What impact would the use of this smart warning app have on academics?



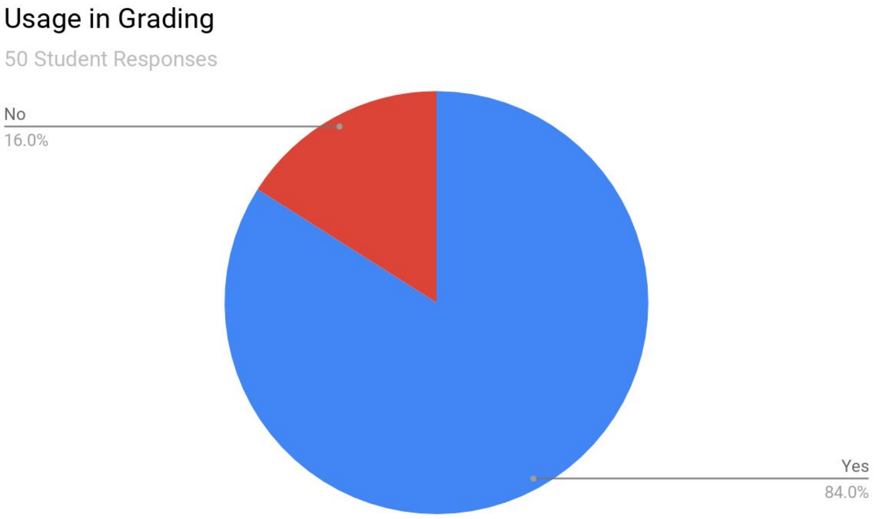
- **Student 1**: [Positive] Lectures would be more productive so less self-study required.
- **Student 2**: [Negative] Constant pressure and alerts to be involved in class would deteriorate concentration.
- **Teacher 1**: [Positive] Would be easier to grade class participation based on concrete information provided by the app.
- **Teacher 2**: [Negative] Require great efforts to reach out to each student and help low performing students.

Q2. What should be the frequency of alerts?

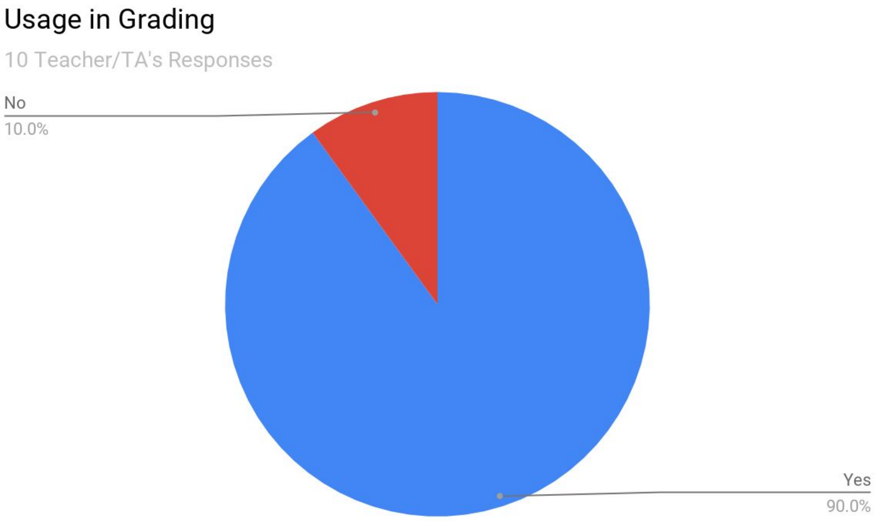

Alert Frequency

60 Responses

Q3. Do you approve the usage of the information generated by the application in course evaluation?

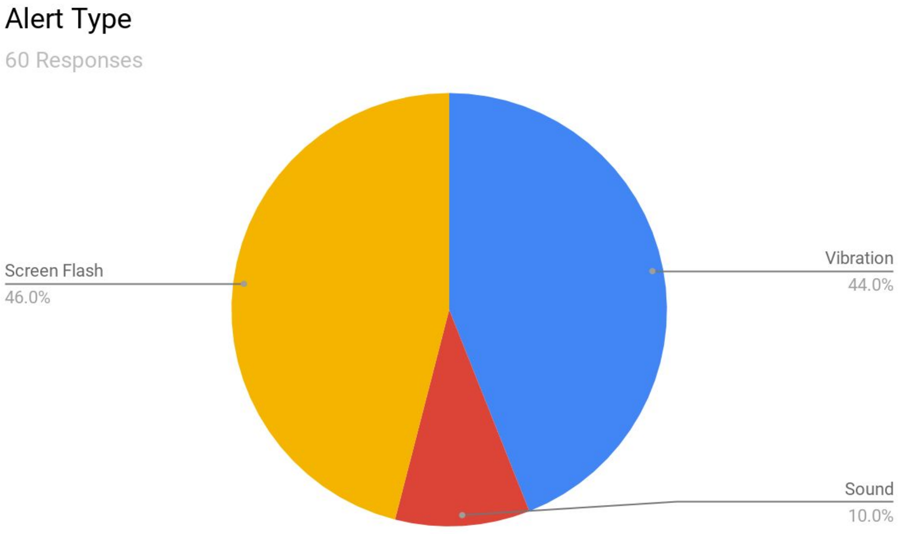

Usage in Grading

50 Student Responses

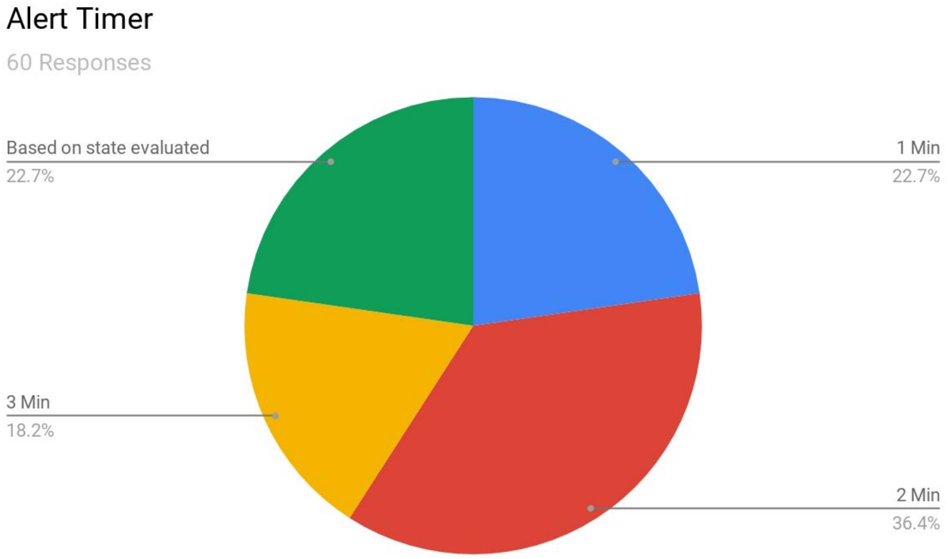

Usage in Grading

10 Teacher/TA's Responses

Q4. What kind of alert notification would you prefer?



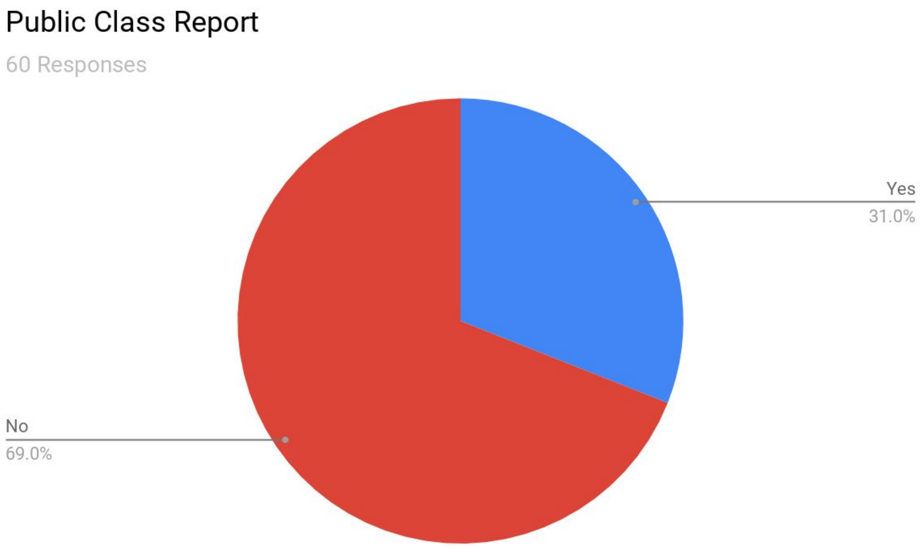
Q5. How much time should a student be allowed to acknowledge the alert notification?



Q6. What mechanism would you prefer to ensure proper action on alerts are taken?

- **Teacher 1 :** Meetup between TA's and students having very less attention state after class hours.
- **Teacher 2 :** Report to DoSAA (or similar authority) in case issues persist after multiple cases for same student.

Q7. Would you want the status report to be made available for all to be accessed?



- **Teacher 1:** Yes, it would promote healthy competition between students and encourage them to participate more.
- **Teacher 2:** No, the status report should accessible only by teacher for grading purpose.
- **Student 1:** Never, I don't want any possible public humiliation.
- **Student 2:** No, I am happy seeing my own alerts and don't want anyone to see my report even if I perform good.

5.6.2 Survey Conclusion

Based on the results of the above contextual enquiry, we implemented the following major features in our project :

- Post Class Report
- Performance Discussion

Functionality of the above functions has been discussed in detail in Section 4.3 of this document.

We save the data of all the classes for future use (Eg. for evaluation) as this was highly encouraged in the survey. The actual frequency and type of alerts implemented in the functions are based on the survey to ensure that the Smart Warning System is unobtrusive during lectures as possible.