CS223
Team No. 22
Project No. 8
Software Requirements Specification Document

1.Introduction

This SRS aims at developing a Android App for the general use of lecturer capable of generating the class view grid representation according to the position of students augmented with specific symbol corresponding to state value of that student .It will illustrate the purpose and complete declaration for the development of system.

1.1 Purpose

This app is being developed as a software project for the software engineering course of Indian Institute of Technology Guwahati. The intended audience would be professors who use Android Application to track the state of the student present during the lecture .The app also allows students to keep track of their state.

1.2 Product Scope

Features:

1. Secured login ensuring security for both instructor and student, instructor is capable of adding new courses database and student is allowed to expand database by registering in courses
2. Locates the position of the student by identifying them by matching the student Id from our database.
3. Augments the state value of the student represented by special symbols(three different symbols of specific color).
4. Represents the students along with their state symbols in classroom representation.
5. Smart Warning system by email if a student does not attend the class.

This Software when completed will be usable by all users who can read and understand English Language and can operate an Android Application efficiently. Though the intended audience is supposed to be the lecturer who will use this app for analyzing the student's state in classroom.

1.3 Definitions, Acronyms, and Abbreviations

API: Application Programming Interface
API level: A measure of the version of Android device being used.
API levels 8~ Android 2.2 (Ice Cream Sandwich)
23~ Android 6.0 (Marshmallow)
ADT: Android Development Tools
MP: Mega Pixel
FPS: Frames Per Second
QR code: Quick Response Code
Beacons: Device used to create signal that can be detected by android device (bluetooth ,WiFi,LiFi etc)

1.4 Document Convention and References

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.Font used is 'Arial'. The headings of each topic are in bold. Bullet points are used.

1.5 Overview

Other parts of the SRS :-
1. Functional Analysis : contains the modular structure of the whole software
2. Softwares/Resources Requirements: Softwares required to construct the product
3. Interface Description

2. The Overall Description

This section provides a background for the requirements, which are defined in section three and thus presents an overall view of the specific requirements.

2.1 Product Perspective

This product is meant to serve as a android application which provides professors a class view and track class conveniently. The app if combined with a external state generator for student can help professor to know each student's class participation and also class strength .

2.1.1 System Interface

The parent system also developed for Android Platform will be directly interfaced with this product so it will not need any external API for communication.However external scanner API for reading QR codes will be required.

2.1.2 User Interfaces

The product will be used with help of any Android device supported by the final product.The app will have a touch screen interface. The class view will be a GUI where user can touch on student position to get information.A "HELP" button will be present to guide user through the functionality of the app.

2.1.3 Hardware Interfaces

A typical android smartphone with the basic peripherals (Touch Screen, Rear camera,WiFi connectivity) is needed to run and have full control of the product with at least version of CPU: Dual Core 1.2 GHz Krait.

2.1.4 Software Interfaces

This system can operate on Android versions 4.4 and above.The software will be connected to a main server .

2.1.5 Communications Interfaces

Local server access to be able to connect to the network of devices connected while the lecture is progressing. All students' Android in turn will be connected to a main interface on the server.

2.1.5 Memory Constraints

Minimum memory storage space of 100Mb.
Camera: 30 FPS: 1.0 MP

2.2 Product Functions

Functions included in the final product with brief summary are as follows, detailing will be done in Functional Requirement :-

1. **User Registration**

   This function creates a new user profile,the user may be a student or an instructor depending on the registration key provided user profile is made.

2. **User Login**

   This function identifies the user who is using the app and depending on the user profile the functionality of the app changes.

3. **Database Enhancing**

   This function registers new courses ,new student in a course or updates a course's information.

4. **Student position detection**

   This function deals with providing the system the information about each student row and column number providing his/her relative position in the classroom coordinate system.

5. **Classroom matrix**

   This function will store all the students position by mapping their coordinates in a matrix (grid like structure/2D array).

6. **Class view generation**

   This function will be responsible for creating GUI of classroom with each student represented as dot, box, image preview,smiley,etc

7. **State Augmentation**

   This function will augment the state which will be generated randomly to the student depending on their state value by different symbols of different colour.

8. **Attendance**

   This function will send warning mail to student absent during the lecture.

2.3 User Characteristic

There will be two type of users, one student and other will be the course instructor.The student will be able to scan the QR code and will be able to see his/her performance and his/her attendance.The instructor can see the class view with the augmented state.The professor can access the record of all the student.

2.4 Assumption And Dependencies

1. User have access to device supporting android 4.4 or higher
2. Device must have a camera to scan the QR code.
3. Every sit in the classroom is associated with unique QR code.
4. Device should have a minimum of 100 mb space for storing the application.
5. The classroom we show is rectangular in shape allowing us to give coordinate representation to each seating position.

6. Our app puts a upper bound on the number of rows and column possible which is equal to 99.

2.5 Apportioning of Requirements.

The function responsible for providing the state of the student is currently random in nature ,in the future versions specific parameters can be introduced on the basis of which a student's state can be evaluated incorporating these function will make our app more useful.
The app till now only takes in consideration a classroom that is rectangular in shape future updated versions can be developed that can represent classroom of any shape and size.The upper bound on the number of row and column can als be increased but in real life scenario a classroom of row/column size >99 is quite unrealistic.

3. Specific Requirements

The specific requirements for the software to work in the desired manner and provide all required features are described in the following section.

3.1 External Interfaces

Input:

1.QR code:- The app requires QR code by the student who has logged in to identify the seating position of the student.

Output:

1.Seat location:-It provides the position of any student present in the classroom.
2.Attendance:-It also provides information about the number of students present and their identity.
3.State of the Student:- Provides graphical representation of the students' state to the professor.
4.Warning Mail:- Sends mail/notification to the student registered but absent during the course.

3.2 Functions

1. User Registration

Input : New user profile ,profile unique id (student roll no, instructor employee no)

Output :Creates a new user profile in the main server.

Description : This function is used by any new instructor or student that is not present in our main server to create a new profile to login.The id entered by the user must be unique failing to do so will create a prompt message stating user with same profile exists.

2. User Login

Input : User credentials (Username , password, user profile)

Output : Redirects to the user account

Description : This function verifies the user credentials and logs the user into the system.At this stage user provides whether he/she is instructor or student, depending upon the user profile user is redirected to his/her profile.It is to be noted that for a student only one login from each android id is possible.

3. Database Enhancing

Input : Depends on the user profile instructor inputs includes new course addition,classroom location etc. and student inputs include course registration , updation of student profile eg phone number, webmail id etc.

Output : All the new entries are rendered to the main server from where further function takes their corresponding input.

Description : This function is responsible for updation of database any new course floated ,new student registered ,time table changes are incorporated in the main server.

3.1 Course Addition

User Profile : Instructor

Input : Course name ,Course code,Room information etc.

Output : New course added to main server registration opens ,the user is redirected to the main profile window.

Description : Allows instructor to add new courses with appropriate information.

3.2 Course Updation

User Profile : Instructor

Input : Change to be reflected (Timing change,Room information changes etc.)

Output : Updates the course in the main server and redirects to the main profile window.

Description : Allows instructor to make changes to the course information of existing courses.

3.3 Course Registration

User Profile : Student

Input : Course Id and enrollment key

Output : Updates the main server enrolling the user with current user Id in the course and then redirects user to the main window.

Description : This function registers the student with the appropriate enrollment key in the course.The main server also

increases the number of students expected to present during lectures .

4. Student Position Detector

We have proposed two methods to identify student position in classroom.Out of these two one will be implemented

Method 1:

1.Position by QR code

Input : QR code

Output : Seat position of the user

Description : This function identifies the seat of user base on information provided by QR code.

4.1.1 QR code Scanner

Input : Entry code,QR code from camera

Output : Unique key extracted from QR code

Description : QR code is a 4 digit encoded number. This number contains row and column information about the seat.
But the QR scanner opens only when the correct entry code for the day,s lecture is provided this ensures no proxies are given as the code is given by the instructor at the start of lecture.

4.1.2 QR code decoder

Input : Key extracted from QR

Output : Row number and column number for seat.

Description : The first two digits of unique code is the row number and last two are the column number.The intersection of the coordinates gives us the exact position of the student.

Method 2:

2. Position by Reduced Signal Strength

Input : Wifi signal strength from all beacons

Output : Relative position of each student

Description : This function locates the nearest device (i.e. student) to our fixed wifi devices by sorting wifi signal strength calculated in each device. Then this nearest device also becomes a beacon and again a new device is found relative to the now existing beacons.

Once relative position of all user is known we can evaluate the row and column where student is seated.The function is called recursively till all the students' location is obtained.

4.2.1 WiFi Enabler

Input : List of all the device Id

Output : Turns on wifi of each device

Description : This function is used to turn on wifi of all the devices to scan our beacons .This function will run only once during one class initially when our two beacons will run at the start.

4.2.2 Signal Strength Calculator

Input : Wifi Id

Output : Wifi signal strength

Description : This function calculates wifi strength so that we can calculate relative position of student device with respect to the beacon corresponding to which the function is called.The function further sorts the devices on the basis of signal strength comparing which we get the position.

4.2.3 Beacon Toggle

Input : Device Id

Output : Device becomes a beacon

Description : This function is used after one recursion of the wifi strength calculator is completed which gives us the position of some more devices .All devices whose position is determined become beacon (by turning on hotspot).

5. Classroom Matrix

Input : Position of every student

Output : A 2-D grid is stored in database maintained in the main server.

Description : Each student is represented as a cell corresponding to their row and column in class by storing it in a 2D array/matrix with the class being bifurcated in grid structure giving a coordinate system structure to each seating position.

6. Class View Generation

Input : Student Position grid

Output : Classroom visualisation map

Description : It generates a graphical view of class according to students seating position.

6.1 Classroom Map

Input : Total rows and columns

Output : A classroom map

Description : The classroom is graphically represented in form of grid structure with each grid representing a sitting position.

6.2 Class positioning

Input : Student seating position grid

Output : Student mapped in grid according to there sitting position

Description: For classroom visualisation a graphical map is generated with the present students represented as dot, box, image preview,smiley,etc while the empty seats are left vacant.

Process

7. State Augmentation

Input : No of students present from server

Output : Augmented view of classroom

Description : This function generates a state augmented view of classroom by augmenting a state symbol n each user according to their state value.

7.1 Random State Generator

Input : No of students

Output : Integer in [1,10]

Description : This function generates random state value for each student in range 1 to 10 **.**Based on these values a symbol is augmented for each student

7.2 State Symbol allocation

Input : State value (an integer between 1 - 10)

Output : State symbol i.e. Tick mark, Question mark or cross

Description : This function decodes the state value of student and provides a state symbol based on table below.The state symbol is then augmented on the student in the class view

Color Coded Symbol	State Value
Red Cross Mark (×)	1 to 4

Blue Question Mark (?)	5 to 7
Green Tick Mark (✔)	8 to 10

8. Attendance

Input : Student position grid

Output : Updates attendance record in server

Description : This function is marks the student present and also notifies the absent students about their missed class.

8.1 Absentee Identifier

Input : Student position grid

Output : List of absent student

Description : Registered Students who didn't log into the app for the ongoing class are marked as absent.

8.2 Missed Class Notifier

Input : Absent Student list

Output : Notification about missed class

Description : Every student who misses the class is notified about their attendance through the app.

3.3 Software System Attributes

This section includes all the non functional requirements for the software:

3.3.1 Reliability

Method 1:

The app uses QR code for identification of position of student ,QR Codes are very reliable, once a QR Code is generated or printed it will not

degenerate or lose the data it holds. It is only if the image becomes corrupt that data can be lost. In the case of a QR Code printed on a physical item (such as paper or a magazine), it is only when that item becomes damaged that data could be lost but can be easily renewed by printing again. Even if part of a QR Code was to become unreadable, QR Codes can have error correction built in that helps ensure a probability of the QR Code still being readable even after being damaged.

Method 2:

The reduced signal strength method can be obstructed by the surrounding objects like wall,chairs,benches etc. However it is evident the disturbance can be taken uniform so relative order of strength for small distances as required by our app will remain same.
Secured login during attendance ensures proxy restriction.

3.3.2 Availability

The system will be available for use whenever the user deems necessary 24/7.The system shall allow users to restart the application after failure with the loss of at most the last operation.

3.3.3 Security

The system will use a login system for authentication and thus will be highly secure and will prevent any type of unauthorized access to private content. The app will be able to differentiate between professor and student allowing the student to access only his/her profile and status whereas allowing the professor to extract the information about class as whole as well as any individual student.

3.3.4 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.

3.3.5 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 using the same method as any other Android App via the Android App Manager.

3.3.6 Contextual Enquiry

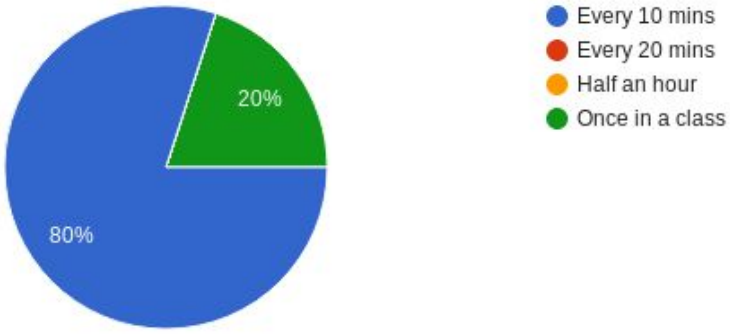
Contextual Enquiry is a user centred design method composed from few hours one on one interaction during which the developer watches the course of users , normal activity, opinions and discusses those activities with the users.
It can be performed in two ways:
1.ACTIVE CI:The user in actively involved in the developing team.
2.PASSIVE CI:The user mere provides his observation and experiences to be recorded.

We have performed passive CI.Following questions were asked to different users and their responses were recorded :

USER PROFILE 1:
INSTRUCTOR

How frequently you will like the app to refresh the state of student ?

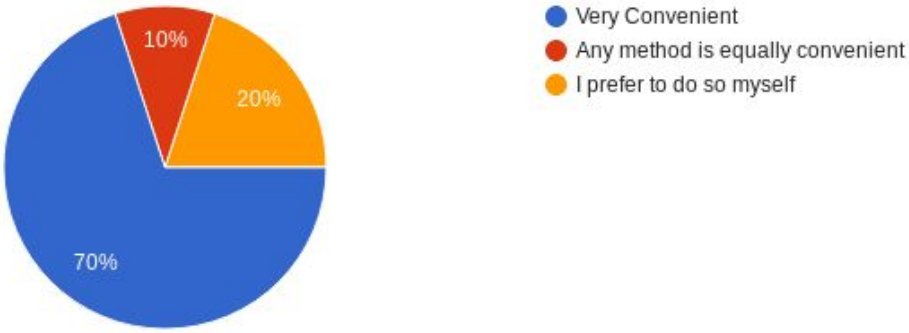
10 responses



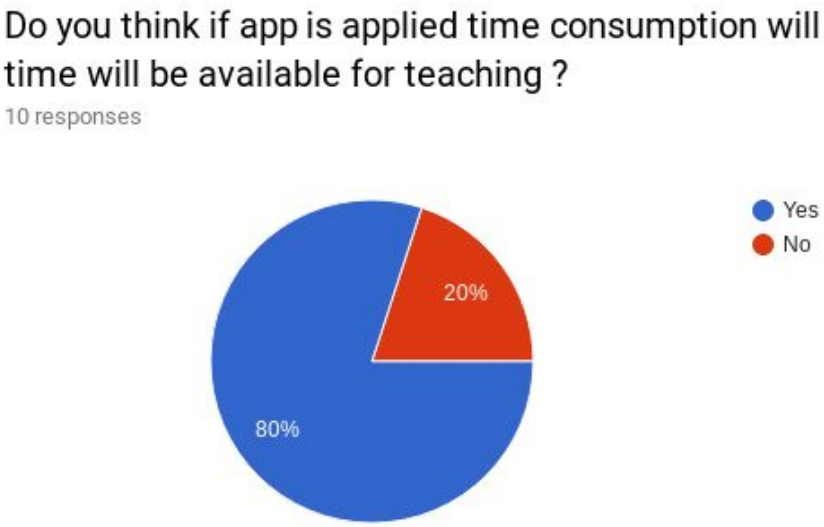
Observation : Most of the instructor wanted the refresh time gap to be as small as possible so that they can monitor student in a better manner.

How convenient you find an app to monitor the attendance and state of your student?

10 responses



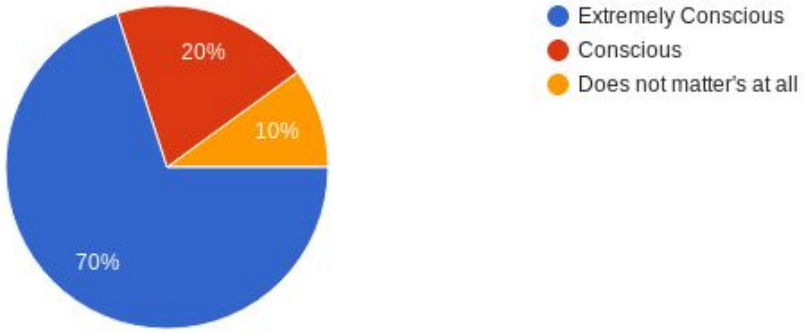
Observation The users found it convenient to use app for monitoring attendance. So we have implemented this feature using **Attendance** function



Observation : Most of the instructors believe that this app could save time during lectures .

How much concerned are you about issue of proxies?

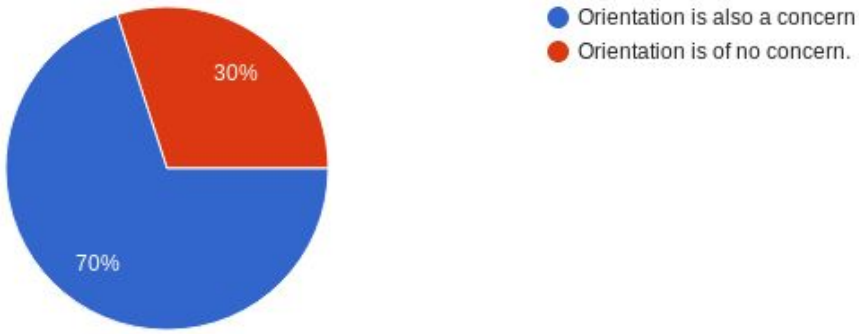
10 responses



Observation : Proxy was a great concern for instructors there were very few for whom it was of no concern.

Do you intent to gather only information about state of the student or class orientation is also a concern?

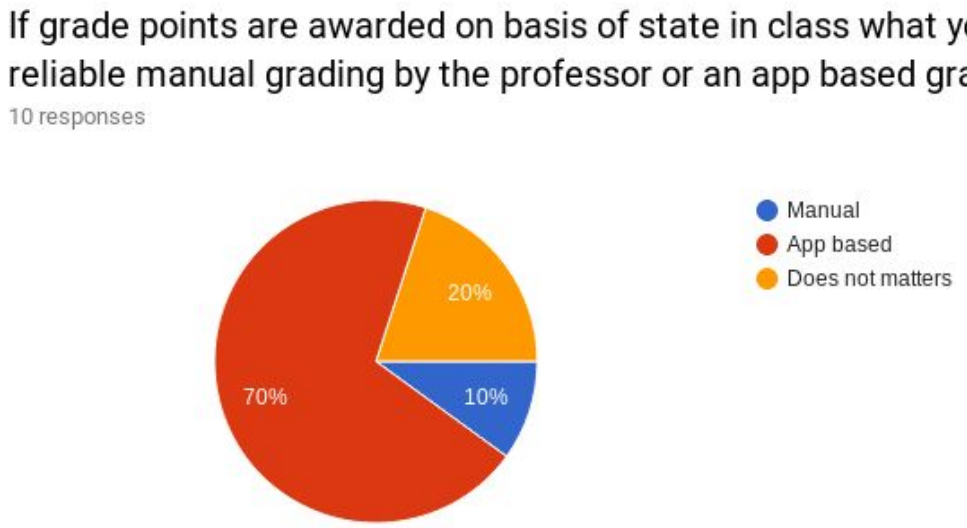
10 responses



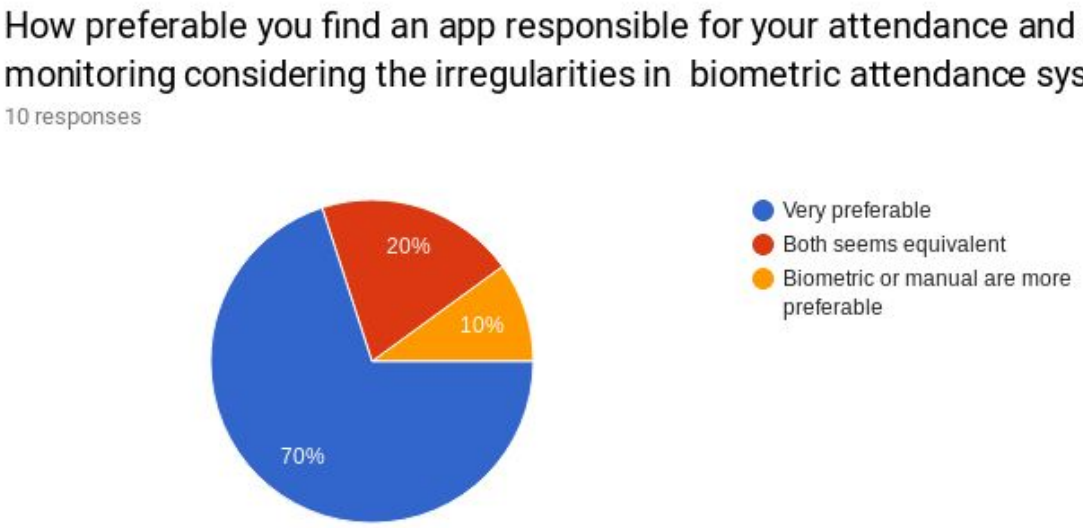
Observation : Yes from the feedback we got the instructor wants both the state of student as well as class orientation so that they can

keep in track whether student with bad states always sit only on the back benches or not. This feature is implemented in our app using **class view generation** function.

USER PROFILE 2
STUDENTS:



Observation : Most of the users find it convenient to use a mobile based app for grading. So our vision clearly matches with the users.



Observation : Most users are dissatisfied with the current attendance system so our application could provide a better alternative to them

CONCLUSION :

1. The responses we got from both the instructors as well as the student recommended us to develop an app using android as the platform.
2. The majority of users believed the accuracy of app will be quite high as compared to the existing manual systems.
3. It will be time efficient and help to increase teaching time.
4. Also the time to refresh state should be small.