

Software Requirements Specification

for

Project 8

Prepared by Harshit Gupta , Harshit Agrawal , Harshit Sharma

IIT Guwahati
Software engineering course

13. February 2018

Contents

1	Introduction	2
1.1	Purpose	2
1.2	Document Conventions	2
1.3	Intended Audience	2
1.4	Product Scope	3
1.5	Definitions and Acronyms	3
1.6	References	4
1.7	Overview	4
2	Overall Description	5
2.1	Product Perspective	5
2.1.1	System Interface	5
2.1.2	Software Interfaces	5
2.1.3	User Interfaces	5
2.1.4	Hardware Interfaces	6
2.1.5	Communications Interfaces	6
2.2	Product Functions	6
2.3	User Characteristics	7
2.4	Assumptions	7
2.5	Dependencies	7
3	Specific Requirements	8
3.1	External Interfaces	8
3.2	Functions	8

3.2.1	Classroom Layout Generator	8
3.2.2	Student's Login	8
3.2.3	Seat Selection	9
3.2.4	State Generator and Allocator	9
3.2.5	Display the Classroom Details	10
3.2.6	Identifying the student Id	10
3.2.7	Students' List (Teacher's View)	10
4	Other Nonfunctional Requirements	12
4.1	Software System Attributes	12
4.1.1	Reliability	12
4.1.2	Availability	12
4.1.3	Maintainability	12
4.1.4	Security	13
4.1.5	Portability	13
4.1.6	Performance	13

Chapter 1

Introduction

The SRS (Software Requirements Specification) aims at developing an Android App for the use of teachers to determine the state of all the students present in the class. The States are indicated by augmenting three different symbols of specific colors given by (Red Cross Mark, Blue Question Mark, Green Tick Mark).

1.1 Purpose

This app is made with the purpose of helping the teacher in knowing the participation of students in the class. By detecting the state of a student, it will help the teacher in determining whether students are following the lecture or not and also provides a feedback to him. This can also be used to mark attendance of the students and allow them to select the seats which they prefer. This app is being developed as a course project for the Software Engineering course of Indian Institute of Technology Guwahati.

1.2 Document Conventions

While creating this document, we followed the format specified by IEEE . The headings of each topic are in bold. Bullet Points are used for sub-headings and key points.

1.3 Intended Audience

This app is intended for:

Android Phone Users :

1. Teachers: It will help the teachers to know the active participation of the students in

the ongoing lecture during the class which will give a true feedback about the lecture. It will also resolve the issue of fake attendance.

2. Students: It will serve the students to mark their attendance and allow them to select classroom seats of their choices beforehand.
3. Others: The administration can keep a check on the performance and participation of students in the classes.

This Software Requirements Document is intended for:

1. Developers: They can review the project and can understand where their efforts should be targeted to further improve this app and also to add new features to it for future development.
2. Testers: Project testers can spot some bugs in the project using this requirements document which can save lots of time as the testing becomes more strategically organized.
3. Others: End Users can read about the features used in this project and can get to know about the utilities of this project.

1.4 Product Scope

This app serves as a tool for both students and teacher, having an android smart phone, helping the teacher to know the state of the students, thereby providing a genuine feedback of the class. It will also help the students in selecting the class seats of their own choice and marking their attendance. Though right now, the app will assign states to students randomly but later on, a function can be added which will determine the states in real-time.

1.5 Definitions and Acronyms

The basic and used keywords in this document are as follows:

- **SDK:** Software Development Kit
- **NDK:** Native Development Kit
- **IDE:** Integrated Development Environment
- **ADT:** Android Development Tools
- **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)
- **Wi-Fi P2P**: Wireless Fidelity Peer to Peer
- **WPA**: Wi-Fi Protected Access

1.6 References

1. Android SDK: <https://developer.android.com>
2. Wi-Fi P2P: <https://developer.android.com/guide/topics/connectivity/wifip2p.html>
3. XML Tutorials: <https://www.w3schools.com/xml/>
4. Android Studio Tutorials: <https://www.tutorialspoint.com/android/>
5. IEEE STD 2004.94445 for peer-to-peer data transport

1.7 Overview

Other parts of the SRS:-

1. Functional Analysis: contains the modular structure of the whole software
2. Software's/Resources Requirements: Softwares which are required to construct the product
3. Interface Description

Chapter 2

Overall Description

2.1 Product Perspective

The perspective of the android based app is to serve the teacher giving the lecture in the classroom. The goal behind developing this app is to develop a mechanism to make the lecturer aware of student's state during the lecture and also displaying their states in an easily interpretable manner. This app also automates the task of taking attendance for the lecturer. Also, selecting seats of the classroom beforehand is provided to the students.

2.1.1 System Interface

The parent system also developed for Android Platform will be directly interfaced with this product and thus won't need any external added APIs for communication. A router will be needed to set up the network to connect teacher's mobile phone with smart phones of all the students.

2.1.2 Software Interfaces

1. The product will be used with help of any Android device with Android 6.0 or above.

2.1.3 User Interfaces

1. The Student Registration page will use form-based interface for letting them enter their details such as Name, Roll No., Webmail, which will allot each student a unique id at the time of registration.
2. The software will be GUI accessible and will use the touch screen interface of the android device to communicate with the user

3. The graphical interface will be rows of boxes each representing a specific seat of classroom.
4. The output will be one of the three different symbols of specific colors(Red Cross Mark, Blue Question Mark, Green Tick Mark) on each box representing the state of the corresponding student sitting on that seat.
5. Each empty box represent no student sitting on that seat.

2.1.4 Hardware Interfaces

1. A typical android smart-phone with the basic peripherals (Touch Screen, Network Connectivity, Wi-Fi) is needed to run and have full control of the product.
2. A router will be needed to set up the network to connect teacher's mobile phone with smart phones of all the students.
3. This app will require 300MB of RAM and 100MB free space and above depending on the strength of the class.

2.1.5 Communications Interfaces

We require a local server access which will be able to connect to the network of devices (teacher's and all students mobile phones) while the lecture is being delivered. All students' android in turn will be connected to the network updating the state of a student sitting on the seat(represented as box).

2.2 Product Functions

1. Functions of the android app are as follows:
2. Random State Generation: Randomly generate and assign states to the present students in the class.
3. Displaying the States: The state of all students in the class is displayed on the app which can be seen by everyone in the connected network.
4. Selecting the Seats: The enrolled students have the choice of selecting their preferred seat before the start of the class. No student can select the occupied seats.
5. The occupied seats will contain the information about the student sitting on that seat in the classroom and also denoting his/her presence/absence in the class.
6. Updating Personal Details: The enrolled students are allowed to update their personal details.

2.3 User Characteristics

The intended users for the product shall have the following characteristics: -

1. The user should be able to afford and use an Android device above the specified API level.
2. The user should be able to understand the functioning and operation of the software on a basic level.
3. The user should be conformable in English language to operate the application.
4. The user should belong to the age group of 18-55 years.

2.4 Assumptions

1. We assume that the student select a seat before coming to the class.
2. We assume that the student will sit on the seat which they selected before coming to the class.
3. The state given to a student is randomly generated and is not accurate with the actual state of the student.
4. We also expect that the students brings his phone in the class and gets connected to the network in the class otherwise he/she will be marked absent.

2.5 Dependencies

1. Local Access Server
2. Network Connectivity
3. Phones Supporting Android Platform

Chapter 3

Specific Requirements

3.1 External Interfaces

The inputs and outputs from the application related to the external environment

Input: The signal received by the central device (teacher's phone) containing the state details from the student's phone act as the input to the application.

Output: The state of the student in form of one of the three different symbols of specific colors(Red Cross Mark, Blue Question Mark, Green Tick Mark). This is displayed in the box representing the state of the student sitting on that seat.

3.2 Functions

3.2.1 Classroom Layout Generator

Input: Number of rows and columns in the classroom.

Output: Graphical Interface of the classroom with the given number of rows and columns is generated.

Description: Each seat of the classroom is represented by a box in the classroom layout. One box represents one specific seat.

3.2.2 Student's Login

Input: Roll No. (representing Username) , password and User type (Teacher/Student).

Output: Either (a) or (b)

- a.) Successfully Logged In
- b.) Error – Entry does not matches.

Description: The user is successfully logged in if the entered details by the user matches with any of the entry in Students/Teachers Database otherwise an error is shown.

3.2.2.1 Edit Personal Details

Input: The Entry that needs to be Updated

Output: The Entry gets updated for the logged in Student

Description: The logged in student can update his personal information and this update gets reflected in the classroom Database.

3.2.3 Seat Selection

Input: Seat which the student wants to pick.

Output: Selected seat will be shown as occupied.

Description: Information about the selected seat i.e. id of the student will get associated with that seat and will be sent to the main server and the seat will be shown occupied.

3.2.4 State Generator and Allocator

3.2.4.1 State Generator

Input: Seat is selected

Output: a random number (1-10) is generated for the selected seat

Description:A number from 1 to 10 is randomly generated and is assigned to the selected seat.

3.2.4.2 State Allocator

Input: the random number generated in the State Generator

Output: One of the three different symbols of specific colors is allocated to the selected seat.

Description:The randomly generated number (input) is classified into three categories based on its value. This input determines the state of the student sitting on that seat as per the following table:

Color Coded Symbol	State
Red Cross Mark	1 to 4
Blue Question Mark	5 to 7
Green Tick Mark	8 to 10

3.2.4.3 State Update

Input: Selected seat with its state information

Output: Update the state of the selected seat in the Classroom Database

Description:The information about the state of the seat is now transferred through connected network, is updated in the database, and is reflected in all devices.

3.2.5 Display the Classroom Details

Input: Classroom Database

Output: : Classroom Layout of the seats along with their respective states.

Description:After updating the classroom database, the layout of the classroom is updated and displays the state of the selected seats in their respective boxes.

3.2.6 Identifying the student Id

Input:Selecting the seat for which students' information is required.

Output:The information about the student sitting on the selected seat.

Description:The information (Roll No, Name, Webmail) can be fetched by clicking the seat if a state has been assigned to it otherwise it displays no student is sitting on the selected seat.

3.2.7 Students' List (Teacher's View)

3.2.7.1 Add a student

Input: Details of new Student (Name, Roll No., Webmail)

Output:New Student is added in the classroom Database and the student is enrolled in the teacher's course.

Description:The teacher has the privilege to add a new entry to the Database while he/she is logged in.

3.2.7.2 Delete an existing student

Input: Roll No. of the student to be deleted

Output:The enrolled student's entry is removed from the Classroom Database.

Description:The teacher can remove an enrolled student (if needed) from the course and hence from the list of the enrolled students.

Chapter 4

Other Nonfunctional Requirements

4.1 Software System Attributes

This section includes all the non-functional requirements for the app:-

4.1.1 Reliability

The software is supposed to work properly till all the mobile phones remain connected to the network throughout the lecture. The software may fail or crash when multiple users try to access the same seat at the same time. The software may also fail when there are network issues. For the software to work smoothly a stable network needs to be maintained.

4.1.2 Availability

The system will be available for use just before the start of a lecture to allow students to select their seats for the lecture and is also available to the teacher during the lecture so that he/she can determine the students' presence. If there are any network issues, then the software shall allow users to restart the application without any loss of data.

4.1.3 Maintainability

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

4.1.4 Security

The system uses a login system for authentication and thus will be highly secure and will prevent any type of unauthorized access to private content.

4.1.5 Portability

The software will be easily transferrable 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.

4.1.6 Performance

The performance of the product depends on the time it takes to update the state of a seat selected by some student which depends on the connectivity of the network. More stable is the network, higher is the performance. In addition, it depends on the hardware (processor) of the devices used.