**TABLE OF CONTENTS**

**1. Introduction**

**i) Purpose**

**ii) Scope**

**iii) Overview**

**2. The overall description**

**i) Product perspective**

i)product function

ii)hardware interface

iii)software interface

iv)communication interface

**ii) User charactersticks**

**iii) Constraints**

**3. Specific requirements**

**i) Fuctional requirements**

**ii) Non-Functional Requirements**

i)availability

ii)security

iii)reliability

iv)portability

v)maintainability

**1. Introduction**

**1.i) Purpose**

The purpose of this SRS document ia to provide a detailed representation of the ATTENDANCE which is to be used by the faculty staff and students to get their attendance details of corresponding subjects. ATTENDANCE will be built on Electron, a JavaScript framework for building cross-platform apps with the use of HTML, CSS and JS. It will be operating system independent and it will have its own installer for different platforms.

**1.ii) Scope**

The ATTENDANCE is a cross platform application. The application offers various operations like viewing attendance for each subject and red mark for low attendance for the students. For faculty, they can take attendance and also tell the percentage of their subject for each students. These services are conveniently grouped and developed specifically for use on your desktop and Android device. All the operations can be performed when an internet connection is available. Admin of the application can register prospective IIIT Kottayam students & faculty to fully experience all services provided.

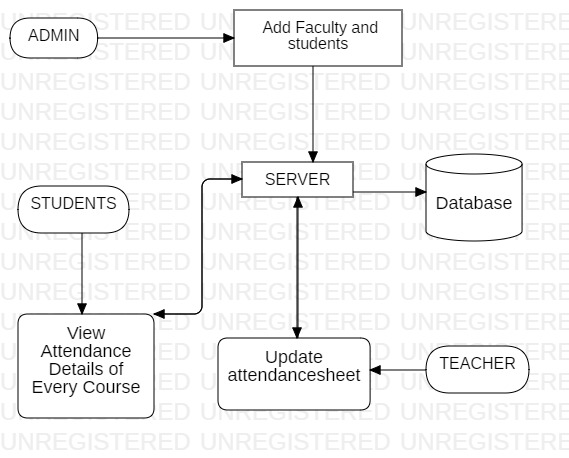
**1.iii) Overview**

The next section, Overall description of the document gives an overview of the functionality of the product. It describe the informal requirements and is used to establish a context for the technical requirements specification in the second section. The third section, Specific Requirements, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product. Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

**2. The Overall Description**

This section describes the functions, aims and objectives of the projects. It also includes the constraints and requirements of the project.

**2.i) Product Perspective**

The ATTENDANCE is a cross platform application project which will be useful for college students and staffs. At present the attendance is taken by the faculty and kept in a register so if students need to check his attendance he has to go to the faculty to ask his attendance percentage. It is very difficult for student to refer to every faculty. To resolve this issue, a desktop and android application is developed and this application is mainly for our college students to get informed about their attendance. In this project the students, faculty & staff are registered by the admin. The faculty of the particular department will upload the attendance in this app by their own login id. By this app the student will automatically get the notification for low attendance of the particular department. This will be more helpful for the students as well as college also. Block Diagram to show various components of the system.

**BLOCK DIAGRAM TO SHOW VARIOUS COMPONENTS OF THE SYSTEM**

**2.i.i) Product Function** The application functions vary with the user using it.

Student:

* On opening the Academic app, a sign-in screen will be displayed. And the student will sign in with the specified details provided by the university at the time of admission.
* Now, user has signed in and a home screen will be displayed with his present attendance in all the subjects individually and his overeall attendance.
* The days on which the student was absent will be displayed in red, and there will be a notification bar to view any low attendance notifications if the student gets any.
* There will be a log out button at the top using which the user can log out of the app.

HOD:

* On clicking the Academic android app, a sign-in screen is displayed. And the HOD will sign in with the specified details provided by the university at the Time of joining.
* Now, user has signed in and a home screen is displayed with 4 drop down lists with Batch number as their label which will produce the corresponding attendance spreadheet of the batch when clicked.
* Total attendance percentage of a student will be showed alongside the spreadsheet.
* There will be a log out button at the top using which the user can log out of the app.

Admin:

Since, admin is a user too. He will sign-in like any other user.

* On clicking the app, a sign-in screen is displayed, and the Admin will sign in with the specified login details.
* Now, user has signed in and a home screen is displayed with the three buttons on it: Add HOD, Add Student & Add Staff.
* Add HOD button will allow the admin to add a new HOD for a particular department.
* Add Student button will allow the admin to add a new Student for a particular batch.
* Add Staff button will allow the admin to add a new Staff for a particular department.
* From, any screen user can switch to home screen and can log out.

**2.1.ii) Hardware Interfaces**

* An android phone or tablet
* Windows, Linux or Mac PC

**2.1.iii) Software Interfaces**

* OS – Windows 7 or above, Linux Ubuntu 16.04 or above, macOS Sierra or above,
* TOOLS – Electron 6.0.2, Android Studio 3.4, XAMPP Control Panel
* PLATFORM: Android SDK Framework
* IDE: VS Code, Android Studio
* EMULATOR: SDK Version 2.2 or Higher
* TECHNOLOGIES USED: JavaScript, Bootstrap, HTML, CSS, Java
* DATABASE- MySQL

**2.1.iv) Communications Interfaces**

IPC renderer module in Electron is used for asynchronous communication between main process and renderer processes.

i) User Characteristics

* The user (student/faculty/staff) should have basic knowledge of using android phones.

**2.2 Constraints**

* The machines having the application should either be connected over LAN or internet.
* Sign-in and password are used for the identification of user.

**3). Specific Requirements**

External Interfaces The User Interface produced by the system are:

**3.1 Functions Requirements**

3.1.1 Module 1: Admin-login

* AIM: Admin can login to get access to the product with the help of unique Login-id & password.
* INPUT: Username & Password
* OUTPUT: Displays a message if username and password does not match otherwise enter into the home page.
* PROCESS: Match username and password from the database.

3.1.2 Module 2: Add Student

* AIM: To add a student(User).
* INPUT: Student name, department, ID, Mobile No, password, email.
* OUTPUT: Student(User) added successfully.
* PROCESS: Save all the data in the database.

3.1.3 Module 3: Add HOD

* AIM: To add a HOD (User).
* INPUT: HOD name, department, ID, Mobile No, password, email.
* OUTPUT: HOD (User) added successfully.
* PROCESS: Save all the data in the database.

3.1.4 Module 4: Add Staff

* AIM: To add a Staff (User).
* INPUT: Staff name, department, ID, Mobile No, password, email.
* OUTPUT: Staff (User) added successfully.
* PROCESS: Save all the data in the database.

3.1.5 Module 5: Update Attendance Sheet

* AIM: To update the attendance record of a batch.
* INPUT: Input to be taken from the filled up spreadsheet after taking attendance.
* OUTPUT: Record updated successfully.
* PROCESS: Update the attendance database.

3.1.6 Module 6: View Attendance:

* AIM: To view the current attendance percentage of a student either overall or subjectwise.
* INPUT: Attendance record from the database.
* OUTPUT: Form showing the subjectwise and overall attendance with details about days present and absent.
* PROCESS: Load data from attendance database.

**3.2) Non Functions Requirements** (Software System Attributes)

Reliability:

This application will run on any android phone running Android 6 Marshmallow or above, PC running Ubuntu 16.04 or higher, Windows 7 or higher, macOs Sierra or above.

Availability:

The application will run 24 X 7 if internet connection is available.

Security:

Security requirements placed restrictions on the use of this application by the student and the faculty of Wireless Lan communicator only, control access to the data, provide different kinds of requirements to different people, require the use of passwords. It requires proper programming techniques.

Maintainability:

This application needs a timely update of information table of the database by the admin. Any other feature as per the requirement can be added any time by the admin.

Portability:

The capability adapted for different specified environments without applying actions or means other than those provided for this purpose in the product. Since, machines are portable, so do the application.