Student Attendance Manager (S.A.M.)

Software Requirements Specification

Version 1.0

Prepared by Group 12

Sahil Kumar	**2016csb1057**
Ram Krishna	**2016csb1053**
Ankan Bal	**2016csb1031**
Arunaksha Talukdar	**2016csb1032**

Indian Institute of Technology Ropar

29 January 2020

1. Introduction

1.1. Purpose

The purpose of this document is to give a detailed representation of the requirements of our software S.A.M. i.e. Student Attendance Manager. Both the functional and non-functional requirements of the software as well as the background and framework is explained to give a comprehensive and complete requirement picture of the software.

1.2. Document Conventions

This Document has been created using the IEEE System Requirement Specification Template.

1.3. Intended Audience

- **Admin**:
  - The Admin will be responsible for creating and adding users and courses to the software database.
  - He will also manage the images uploaded by the app by either deleting them on demand or apply the algorithm for facial recognition and updating attendance record.
- **Faculty**:
  - The Faculty will use our app for taking the snaps from their smartphone camera and either upload them to the server for the attendance of the students or delete them if taken mistakenly.
  - They will also be able to sanction the leave of the students through the app.
- **TA**:
  - The TAs will also be using the app for taking snaps of the students using their smartphone camera and upload them for attendance or delete them if required.
- **Student**:
  - The students will be able to use the app for checking their attendance record for any course they opted.
  - The student may also apply for leave.

ID	Stakeholder	Description
S-1	Admin	Manages the course related data in the database
S-2	Faculty/TA	Uses the app for taking snaps for attendance
S-3	Student	Uses the app for checking the attendance record
S-4	Development Team	Formulates the functional and non-functional aspects of the software and properly develop it
S-5	Q/A	Making test-cases and plans to ensure the quality of the software and fixing bugs

1.4. Product Scope

Our software will provide the android interface for teachers and TAs to take snaps from their camera and upload it to the server to apply facial recognition on the snaps and get their attendance and for students to check their attendance for any specific course. It will also provide a web platform for the admin to create and add users of the software to the server and store, delete or apply the algorithm to the said snap. The scope of this project can also be extended to various other areas and applications such as finding missing individuals mostly children, aiding the blind people by giving them live emotion data of people around, recognize drivers for cars as a medium of locking car doors etc.

1.5. References

- IEEE Software Requirement Specifications template: https://gephi.org/users/gephi_srs_document.pdf
- Ian Sommerville, Software Engineering, 9th Ed https://ifs.host.cs.st-andrews.ac.uk/Books/SE9/
- Clean Code: A Handbook of Agile Software Craftsmanship by Robert C. Martin

2. Overall Description

SAM, a system for attendance management of student will keep track of student attendance in various courses. It will have an android app and a web application. The android app will be used by TA's, professor and students. TA's and professors can upload or delete the image of the class and the app will send the image to the server. TA's can upload multiple images of the class. The students will use this app to check whether their attendance have been marked or not. They may also apply for leave request.

The web application will allow an admin to set up various courses and students information in the system. The web app will invoke face recognition on the images sent by the TA's or professors from the android app. After face recognition it will mark the attendance of the students.

2.1. Product Perspective

SAM is a standalone product that consists of a web app and an android app. Its major functioning is marking the attendance of the students in courses.

2.2. Product Functions

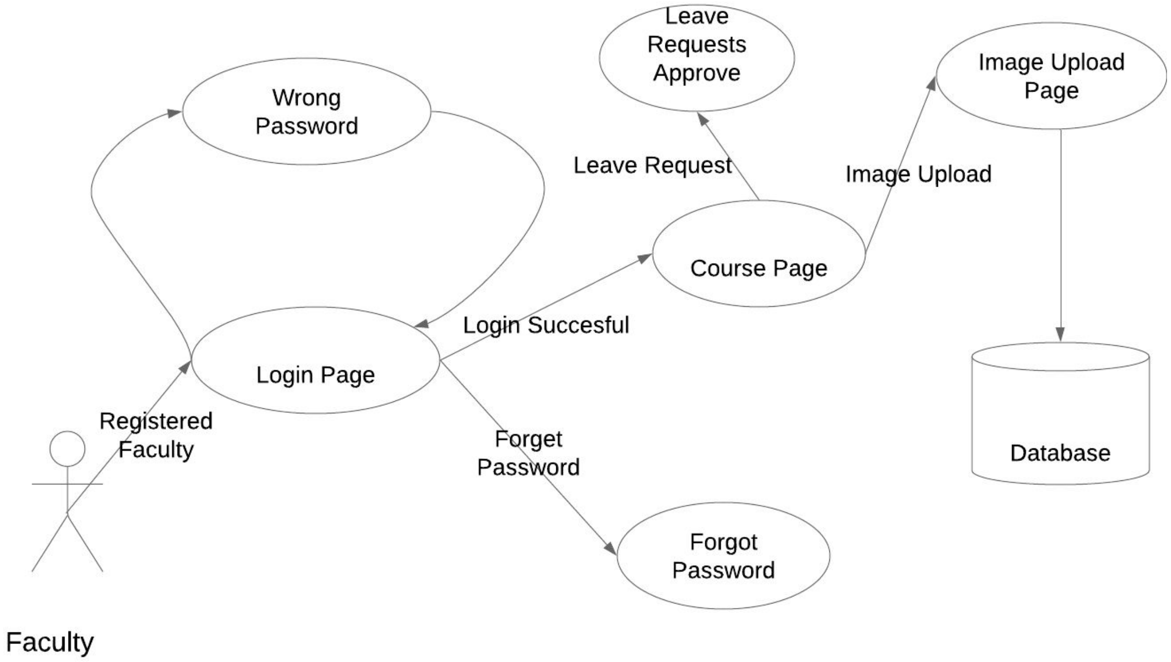
Major functions that our system provides to the users are:

- Functions of the android app
  - Students, TA’s and professors can login
  - Students, TA’s and professors can change password or reset password
  - TA’s and faculty can see the list of courses which they are offering
  - TA’s and faculty can select the course for which they want to mark attendance
  - App will open camera and it will click the pic of the class.
  - TA’s and faculty can upload multiple images to the server
  - TA’s and faculty can delete the images also
  - Students can check their attendance record for courses in which they are enrolled
  - Students can request leave also
  - Professor can approve the leave request

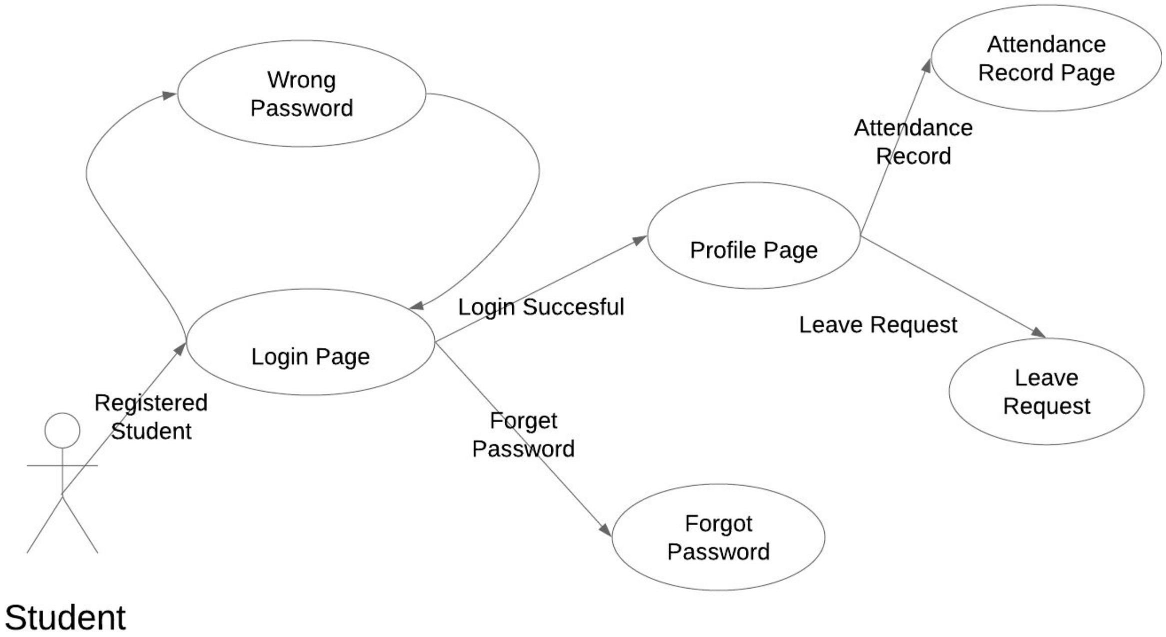
- Functions of the web application
  - Provide unique login details to each students and professors
  - Add students, TA’s and professors to a course
  - Stores the images uploaded from the android app in file system
  - Applies face recognition on the images uploaded
  - Marks the attendance of the students
  - Demarks the attendance of the student in case of image deletion
  - Remove the TA’s or courses once offered

Use Case Diagrams

The use case diagram of faculty is shown below. The use case diagram of TA will be similar to the course instructor.



The use case diagram of students are shown below.



2.3. User Classes and Characteristics

- Admin :
    - He will maintain all the records of attendance of students enrolled in the courses.
    - He will add students, TA’s and faculty in the course.
    - He will handle the web application.
    - He will no longer require the register to maintain the attendance of students.

- Faculty / TA :
    - He will take the images of the class and uploads it.
    - He can delete the images also if required.
    - He can upload multiple images if one image does not contain all the students.
    - He can approve leave requests of the students also.
    - This app can save his teaching time that is usually spent in taking attendance.

- Students:
    - He/She will see the attendance record in various courses.
    - No longer to raise hands.
    - Can see in which course his/her attendance falls below threshold.

ID	**User classes**	**Description**
U-1	Admin	An employee of the academic section who has complete control over the system. He owns expanded rights inside the web application. He performs addition of students, TA’s and professors into courses and maintaining the attendance records of the students.
U-2	Faculty	A user of the android app who has the access of uploading images to the server and deleting the images from the server. He can also approve the leave request of the students.
U-3	Students	A user of the android app who can view the attendance records in various courses and can request leave.
U-4	TA	A user of the android app who has the access of uploading images to the server and deleting the images from the server.

2.4. Operating Environment

The web application will be hosted on any free web host service like github, AWS, etc. The web application will be available on any browsers. The android app will be designed to run on all devices which runs on Android version V5.0 and above. It comprises of more than 90% of devices.

2.5. Design and Implementation Constraints

Our system will use encapsulated and modular design principle. It will follow OOPs and basic software development practices.

The system will contain web application and android app. It will use the following technologies:

- The web application will be implemented in HTML, CSS, NODE.JS
- The database used will be MySQL
- Standard data exchange format will be XML,JSON.
- For android app Java in Android Studio will be used.
- Various modules will depend on each other through well written API.
- Face Recognition API will be used.
- HTTP Protocol will be used for sending data to the server.
- Memory will depend on the number of images stored on the server.
- Internet will be required for operating.

2.6. Assumptions and Dependencies

Admin will provide user id and password to all the students and professors. Students or professors will not sign up by themselves. The admin of academic section should know about databases and web applications. The performance of face recognition api will be enough to identify all the users in the image. Students and teachers must have an android phone and know how to operate it. They should have an active Internet connection in the phone.

Our system is heavily dependent on the performance of the face recognition api. So, in order to perform well face recognition api should perform well.

3. External Interface Requirements

3.1. User Interfaces

Android App:

1. **Login Display:** The users(i.e teacher/TA/student) will be asked to login first in the app. Their login details(username and password) will be verified from the server and the user would be given an option of “forgot password” which if clicked will mail the password reset option to the user. Similarly a logout option will be present.

2. **Home Display:** After login, the user will be shown a list of all the courses that they are offering(for profs/TA) or they have taken(for students) and “No courses yet!” will be displayed if no course has been offered/taken by the user. The list of courses will have all courses as buttons that may be clicked.

3. **Course Display (Prof/TA):** After a course has been clicked, all the course related things will be here like for the prof/TA, they will have the option of taking a photo for attendance, seeing attendance of various students for various days, watching and deleting the photo for attendance of previous days.

4. **Course Display (student):** After clicking a particular course’s button, students can see their attendance record for the course. There will also be some additional information about the courses.

5. **Leave application (for students only):** There will be an option for the students to apply for sick leaves. He will go to a course and there he will apply for the leave along with its reason, this leave application will go the professor of the course who would choose to accept or reject the request.

6. **Leave approval (for professor only):** On the course display page of professors, there would be an option to approve or reject the leave requests of the students of that course.

Web App:

1. **Login:** The admin will have to login to the website. He will also have a “forgot

password” option that will mail him a link to reset password.

2. **Home display:** Here, he will have a specific link for all his tasks. He will have buttons such as add student, add prof, add TA, add course, etc.

3. **Add Student/Prof/TA:** There will be three separate web pages to add student, prof and TA for any course. Those pages will ask the basic details to add the entity to the database.

4. **Add course:** The admin can add a new course, assign professor and TAs to the course. He/She can also add students to a course. These actions will make changes in the database.

5. **Student details:** The admin can view the course and attendance record of any student through this button. He may also modify the attendance if required (eg. in case if the person’s face could not be recognized by the face detection API).
   There maybe some other buttons to view prof/TA/course details.

3.2. Hardware Interfaces

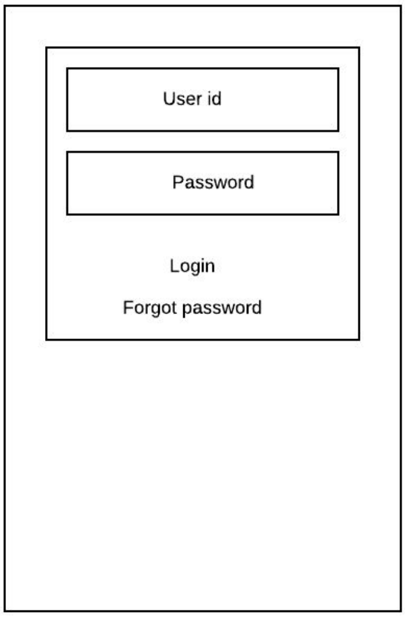
The hardware part of this project will require android devices, one for each professor and TA with functioning cameras. Further a functioning computer would be required by the admin to access the web application.
The web application will be hosted on a web server.

3.3. Software Interfaces

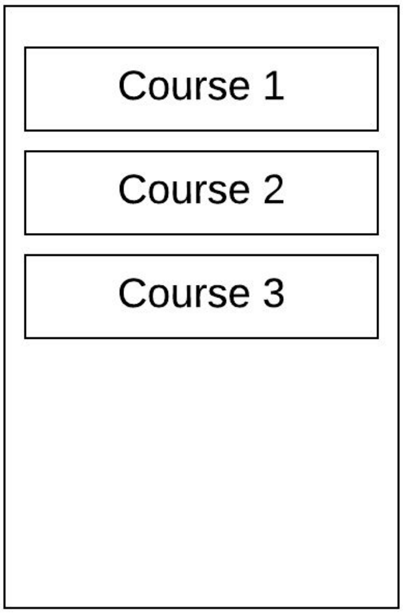
We are planning to use node.js along with a database using MySQL that will be hosted. For android app, we are going to use android studio. Further we’ll have to use a face recognition API that will return the name of students on uploading a photo.

3.4. Communications Interfaces

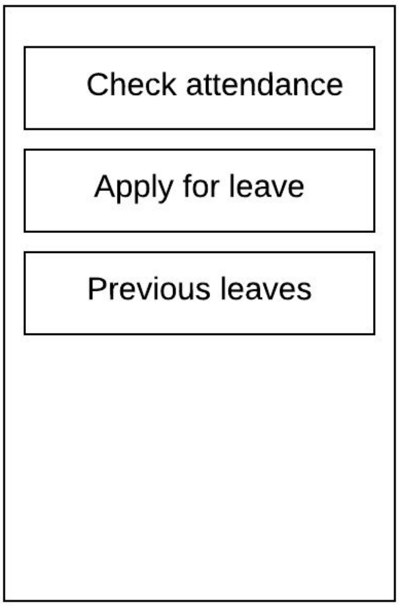
Internet connection will be required by both the android user as well as the admin to access the web application.



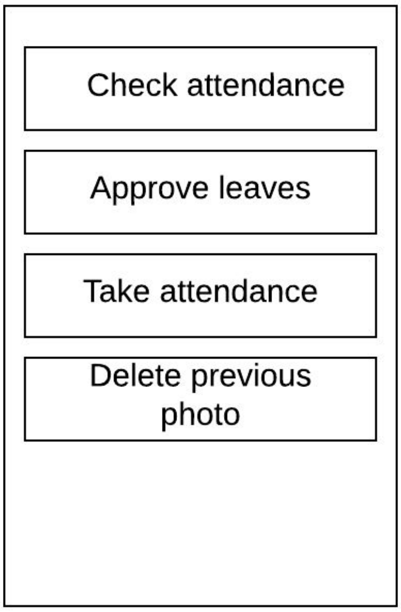
Android app’s GUI(login page)



Android app’s GUI(Courses page)



Android app’s GUI(Particular course’s page-students)



Android app’s GUI(Particular course’s page-Prof.)

4. System Features (Use Cases)

Android application

4.1. Use Case “login”

Purpose	logging in the application
Actor	User of app (Student, professor or TAs)
User input	His/her id and password
Precondition	1. App must have access to the internet 2. Not already logged in the app
Postcondition	Logged in if correct id and password, else prompt for reentry
Basic flow	Login page opens where user enters id and password. This is sent to the server for verification and then depending on correct/incorrect id and password, login complete and main page opens or login page opens up again.

4.2. Use Case “forgot password”

Purpose	Reset password
Actor	User of app (Student, professor or TAs)
User input	Clicks on forgot password on enter user id
Precondition	1. App must have access to the internet 2. User ID must be present in the database with a valid mail id
Postcondition	Mail sent to the user with a link to reset password
Basic flow	User id will be sent to the server with a request to reset password, the server will then find the user’s mail id from the database and send him/her a link to reset password.

4.3. Use Case “Course page”

Purpose	Display all courses taken/offered by the user
Actor	User of app (Student, professor or TAs)
User input	Opening of app/logging in the app
Precondition	1. If the user has already logged in then simply opening app will redirect here as it will be home screen. 2. Else, login of user is required 3. Internet connection is required
Postcondition	All the courses are displayed in the form of button and for further operation of these courses, simply click on the course name. If not logged in then login page will open up.
Basic flow	User id will be sent to the server which will send the information about all the courses of the user. These will be displayed in the form of buttons. In case the user has not logged in, the login page opens up
Alternative flow	If not connected to the internet then list of courses will appear based on his cache(provided login is successful).

4.4. Use Case “Check attendance”

Purpose	Check attendance in a course
Actor	User of app (Student, professor or TAs)
User input	Click on check attendance button present in course page from the previous use case.
Precondition	1. Internet connection is required 2. User must be enrolled (or offering) in the course.
Postcondition	Student gets his attendance in the course. Prof/TA gets the attendance of the entire class in that course.
Basic flow	The app will request server to provide the attendance for the student/students of the course. The response will be displayed.

4.5. Use Case “Apply for leave”

Purpose	The student will apply for leave
Actor	Users(Only students) of the app
User input	Inside a particular course’s page, click for apply leave and input date and reason for leave.
Precondition	Internet connection, login and enrol in course required
Postcondition	The request will go the concerned professor(s) and they will also get notified.
Basic flow	The leave application will be submitted to the server which will send it to the professors of the course.

4.6. Use Case “Approve leave”

Purpose	The professor approves/rejects student’s leave application
Actor	User of the app(Professor only)
User input	Click on approve leave on homepage.
Precondition	Internet connection, login and offering course required
Postcondition	Leave application is approved/rejected and it is reflected in the student’s app. Also, student is notified.
Basic flow	All the leave applications of the students will appear on the professor’s app. He can also see how many leaves the student has already taken in the course. He can approve/reject the application with an option to add comment. This will be sent to the student who asked for leave.

4.7. Use Case “take attendance”

Purpose	click photo of the classroom to take attendance
Actor	User(Prof/TA)

User input	click on take attendance in the course page
Precondition	1. Internet connection, login and offering course required 2. The app must have access to camera and storage permissions.
Postcondition	The photo will be uploaded to server that processes and marks attendance of students involved.
Basic flow	The camera app will open up and a photo maybe clicked by the user. Then the user will click on upload that uploads it to the server. The user may upload multiple photos for the same class. The server will process it and other photos of the day and then mark attendance of the required students.

4.8. Use Case “Delete photo”

Purpose	Delete a photo of a particular day
Actor	User(Prof/TA)
User input	click on delete photo in the course page
Precondition	Internet connection, login, offering course is required.
Postcondition	The photo that was previously uploaded maybe deleted from here and the attendance of the student in the photo will be unmarked if the student is not present in other photos of the day.
Basic flow	The app will request the server to show uploads of a day for some course. Then the chosen photos will be deleted and attendance will be unmarked.

4.9. Use Case “Notification for leave”

Purpose	Notify student/prof about leaves
Actor	Server
User input	Server will send leave request/status
Precondition	1. Internet connection, login is required. 2. Permission for push notification enabled.
Postcondition	When the professor will receive a new leave request, he will be notified through push notification. When student’s leave request will be accepted/rejected he will get notified.
Basic flow	The app will receive data from server when new leave is applied or old one’s status changes. On receiving such data, a push notification/in-app notification will be sent.

Web application

The login, forgot password use cases of admin will be just like the ones mentioned in android application section.

4.10. Use Case “Add/remove new student/course/professor”

Purpose	To add new student/professor/course in the database
Actor	User of the app(Admin)
User input	Click on the add button
Precondition	Internet connection is required. The admin must have logged in.
Postcondition	New add complete. The password of new student/professor assigned and mailed to the student/professor.
Basic flow	The admin will input all the details and this will be sent to the server that will put it in the database. In case of new student/prof, mail will be sent containing user id and password.

4.11. Use Case “Add/remove students/professors/TAs to course”

Purpose	Add/remove students/professors/TAs to course
Actor	User(Admin)
User input	Click on button
Precondition	Internet connection is required. The admin must have logged in. The course must’ve been floated.
Postcondition	The concerned person will be added to the course.
Basic flow	The admin will assign professors and TAs to any course and then will add all the students that want to take up the course. It will submit this query to the server which will make all these changes to the database.

4.12. Use Case “check attendance”

Purpose	Check attendance of the students
Actor	User(Admin)
User input	Choose the student whose attendance is to be checked or feed in some number x(default 75%) to show all the students below x% of attendance.
Precondition	Internet connection is required. The admin must have logged in. The course must’ve been floated.
Postcondition	List of attendances and students will be obtained
Basic flow	The admin will submit name of student/course/x and this request will be sent to the server that will query the database and return the results to the admin.

4.13. Use Case “Photo Processing”

Purpose	Mark/unmark attendance
Actor	User(professor/TA)
User input	upload/deletion of a photo
Precondition	1. Connected to the internet, login required 2. Access to face recognition API required
Postcondition	Attendance will be marked or unmarked for the student(s) involved
Basic flow	Professor/TA will upload/delete a photo, the people in that photo will be identified using the face recognition API. Then after verifying the people are present/absent in other photos of the course for that day, the attendance will be marked/unmarked for the students.

5. Other Non-functional Requirements

5.1. Performance Requirement

- The Computer Hardware should be powerful enough to run the applications.
  - It should have sufficient RAM & Memory.
  - It should have a good resource manager.
- Internet connection is required.
- Decrease the average time the software takes to accomplish a user’s goals,(how many tasks a user can complete without any help, the number of transactions completed without errors, etc.)
- It should be have a Low perceived workload (attempts that are needed by users to accomplish particular tasks)

5.2. Safety Requirements

- The data related to the software will be stored on an online server and a backup compressed image file of the data will be stored on a backup local server of the admin.
- In case of any data loss in the main server, the data can be recovered from the backup image data. The backup data will be periodically updated to ensure minimum data loss.

- In case the backup server fails, then another backup can be created from the online server data.
- For the overall proper function of the software it is required that every student abides by the code of conduct of a student.

5.3. Security Requirements

- The Registration process of students,teachers and TA’s is to be done by an admin.
- Data is to be kept securely on server.
- Access & Usage Policies:-
    - Student is the dominant user Entity due to its membership size.
    - Faculty & TA observe the activity of students.
    - Registration of Faculty & TA is done by administration staff.
- Access & usage will be made available to the authorized students,teachers and teaching assistants by the administrator.
    - The Records are formed on the day they start school and kept till they graduate.
    - The Administrator is responsible for the genuinity of the data which is stored in the database.

5.4. Software Quality Attributes

5.4.1. Reliability

- The Overall Reliability is dependent on the Reliability of various API’s used.
- The API for the facial recognition algorithm will be responsible for the efficiency of the software.
- The Reliability of the Software is enhanced by hosting it online.
- The Reliability of the Software data will improve due to using two servers for storing them i.e. one for the main online server and one as the backup server.

5.4.2. Availability

- Access & usage will be made available to authorized students,teachers and teaching assistants by the administrator.
- Usage not requiring online support are available at times of disconnectivity with internet.

Appendix A: Glossary

Term	Description
S.A.M.	Student Attendance Manager
Node.js	Open Source Server environment using js as server
API	Application Programming Interface
IEEE	Institute of Electrical and Electronics Engineers