Project Report
On
Attendance Taker Application

Submitted By
Miss. Urvi Girish Dedhia

Under the guidance of
Mr. Maunash Jani

Submitted in partial fulfillment of the requirement
for qualifying T.Y.B.Sc. Computer Science
Semester V Examination 2020

DEPARTMENT OF COMPUTER SCIENCE
KISHINCHAND CHELLARAM COLLEGE
Churchgate, Mumbai – 400 020

KISHINCHAND CHELLARAM COLLEGE
Churchgate, Mumbai-400 020

CERTIFICATE

This is to certify that Ms. / ~~Mr.~~ **URVI GIRISH DEDHIA** Seat no. **12** of **T.Y.B.Sc. Computer Science** has completed ~~his~~ / her ~~Windows based~~ / ~~Web based~~ / Android based project entitled **Attendance Taker Application** in partial fulfillment of the degree of **B.Sc. in Computer Science** for **Semester V** under the University of Mumbai for the academic year 2020-2021.

It is further certified that this project had not been submitted for any other examination and does not form part of any other course undergone by the candidate.

Project Guide

Coordinator

Date______________

Examined By________________

ACKNOWLEDGEMENT

I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project.

First of all, I would like to thank our College Principal **Dr. Hemlata K. Bagla,** for being a source of inspiration to do something creative, innovative work for the society.

I would like to thank the Vice Principal and Co-ordinator of the Computer Science Department, **Dr. Shalini Sinha,** for extending her support.

I thank **Mr, Maunash Jani,** my project guide, for his valuable inputs in the making of this project.

My thanks also extends to other teaching and non-teaching staff of the Computer Science Department, for being supportive throughout the year.

I am grateful to my parents and all my friends for their constant encouragement and support for assisting with their input.

INDEX

Sr no.	Topics	Pg no.
I	**Preliminary Investigation**	**1**
i.	Description of Current System	2
ii.	Limitation of Current System	2
iii.	Description of Proposed System	3
iv.	Limitation of Proposed System	3
v	Requirement Specifications	4
II	**System Analysis and Design**	**5**
i.	Event List and Event Table	6
ii.	Class Diagram	9
iii.	Use Case Diagram	10
iv.	Sequence Diagram	11
v.	Object Diagram	12
vi.	State Diagram	13
vii.	Activity Diagram	14
viii.	Component Diagram	15
ix.	Package Diagram	16
x.	Deployment Diagram	17
III	**System Coding**	**18**
i.	Validations	19
ii.	Test Cases, Test Data and Test Results	20
iii.	Screen and Report Layouts	22
IV	**System Implementation and Uploading**	**34**
V	**Conclusion and Future Scope**	**36**
VI	**Future Enhancements**	**38**
VII	**Reference and Bibliography**	**40**

PRELIMINARY INVESTIGATION

DESCRIPTION OF CURRENT SYSTEM

1. In College or any other sector, the attendance is taken manually by checking the name of every member and marking them if they are present.
2. Most of the colleges use a register book for attendance and they store the stack of these books for records.
3. The other alternative way is to ask every student to sign against their name and then the respective teacher register the attendance in an organized sheet.
4. The available attendance taker application for mobile devices still demands the need to mark the attendance manually.

LIMITATION OF CURRENT SYSTEM

1. The current system uses a stack of papers and the work is done manually which is time consuming.
2. It may happen that the teachers may miss to mark the attendance to any of the students or any student may be able to mark proxy attendance of their classmates.
3. It also follows heavy workloads and the risk of managing and organizing is exhausting.
4. It is difficult for one to carry the register book; hence cannot use it at any time of day.

DESCRIPTION OF PROPOSED SYSTEM

1. The Attendance Taker Android Application creates an account for teachers to store their attendance record on cloud.
2. The user i.e. the teacher is able to create or delete a class, add a list of subjects taken in that class by that teacher, add or remove students, provide additional information about students.
3. It shows the list of classes, list of subjects, list of students of the particular class and their past attendance records.
4. The application provides the statistical analysis and graphs to understand the performance of each student.
5. It is flexible as the user is able to use the application, take the attendance or analyze the attendance records of any student anytime.

LIMITATION OF PROPOSED SYSTEM

1. The application does not provide the facility to read the excel sheet of the already taken records other than the app, hence cannot store the previous attendance records in any other format.
2. It does not provide the feature to convert the list of class, members or past attendance in any excel sheet form. Hence there is no way to retrieve the data in any other format.

REQUIREMENT SPECIFICATION

HARDWARE REQUIREMENTS

1. 64-bit Environment
2. 250 GB free disk space
3. Memory - 8 GB RAM
4. Internet Connection
5. Android Device / Emulator of 8 GB RAM -
    - 512 SD card
    - 8 GB RAM
    - CPU/ABI: Google Play Intel Atom (x86)

SOFTWARE REQUIREMENTS

1. Database - Firebase
    - Firebase Firestore version: 21.5.0
    - Firebase Authentication version: 19.3.2
2. Operating System -
    - Linux: Ubuntu LTS 18 or later version
    - Windows: Windows 10
    - Mac: MacOS 10.12+
3. Android Studio -
    - Version: 4.0 or later version
    - Minimum SDK version: 24
    - Target SDK version: 29
    - Minimum Gradle daemon heap size(Java): 2048MB
4. Android Device / Emulator -
    - Hotspot feature
    - Nougat 7.0 or later version
    - API level 24

SYSTEM ANALYSIS

EVENT LIST

1. Teacher -
   - Create an account
   - Add classes, subject and students
   - Take attendance and download the csv file
   - View past records of each attendance
   - Analyse the student performance
2. Admin - The admin controls the operations on the user account.

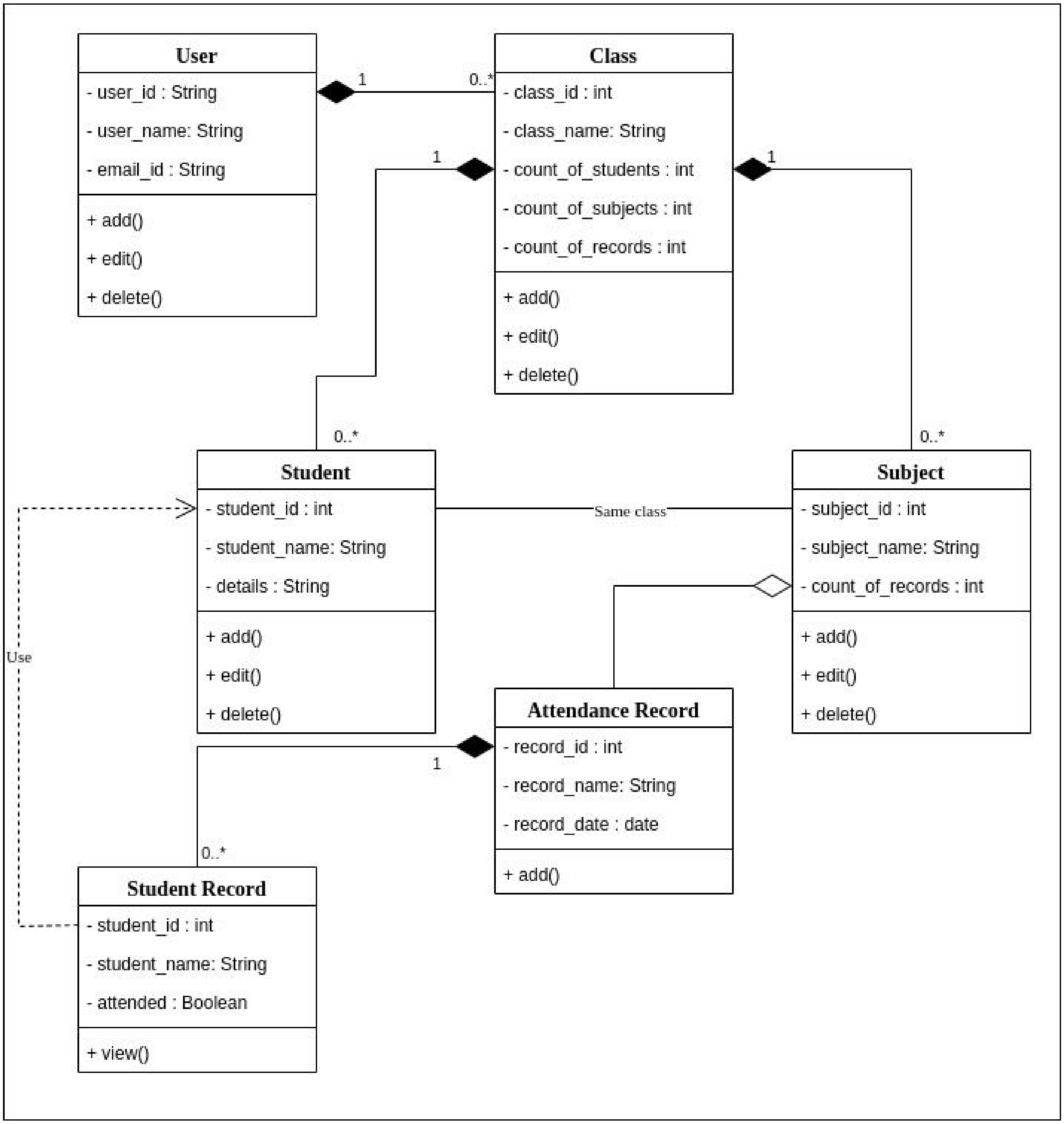
EVENT TABLE

Event	Trigger	Source	Activity	Response	Destination
User Sign-in	Add User	User	Create new user account	User account is created	System
User Login	Login Request	User	User login to the app	Display the home page	User
Enter new Class	Add	User	Create a new class	New class gets added i.e. new class gets displayed on the list	System
Delete a class	Delete	User	Delete the existing class	Existing class gets deleted i.e. class list gets updated	System
Edit a class	Update	User	Update the class name	Existing class name is updated with new name	System
Enter new student	Add	User	Create new student member	New member is added	System

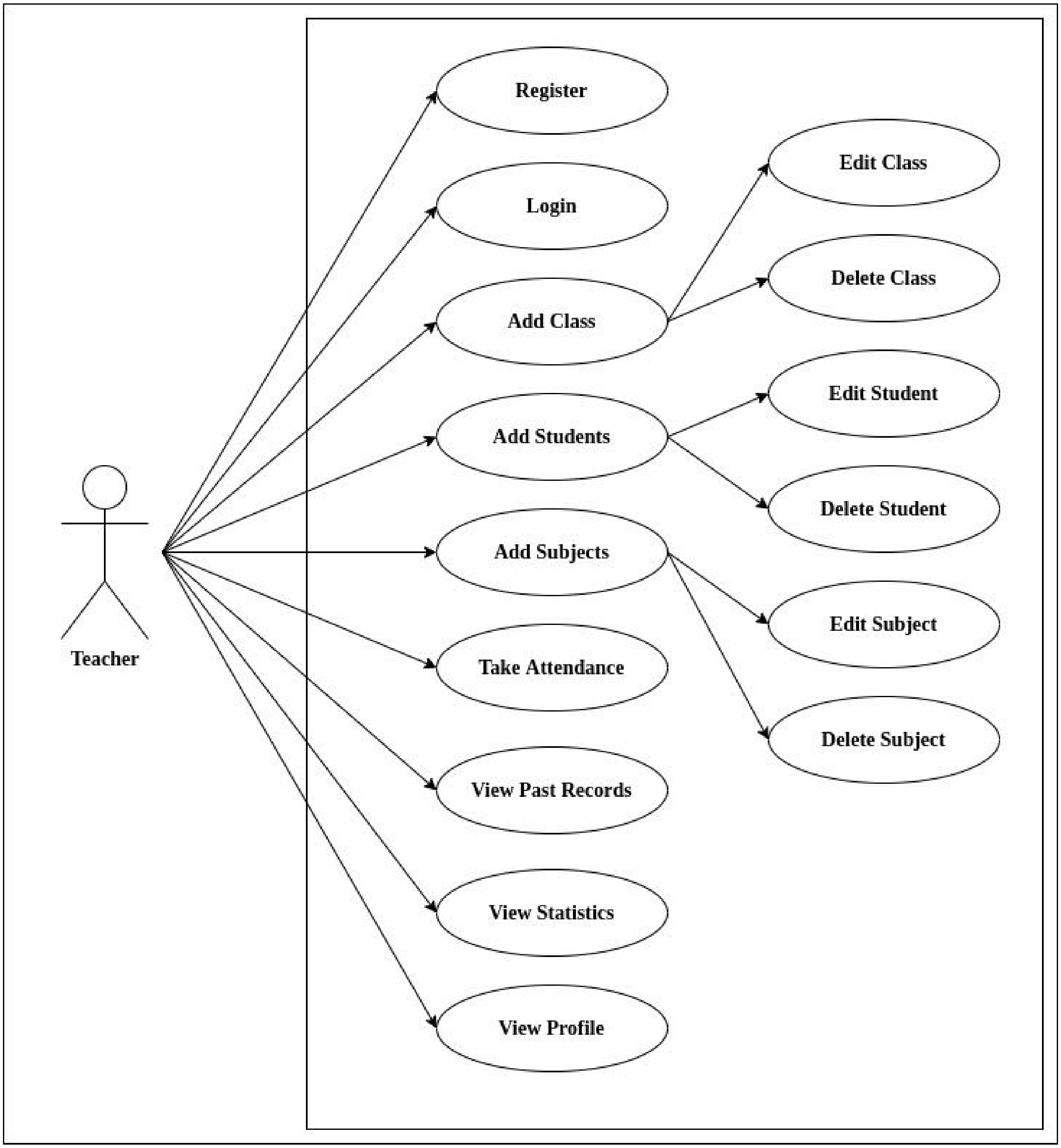
Delete a student	Delete	User	Delete the existing student	Existing member gets deleted i.e. updated student list gets displayed	System
Edit a student	Update	User	Update the name of student	Student is updated with new name	System
Enter new subject	Add	User	New subject is added	Subject List is updated i.e. new subject is added	System
Delete a subject	Delete	User	Existing subject is deleted	Updated subject list is displayed i.e. subject gets deleted	System
Edit a subject	Update	User	Change the name of subject	Existing subject is displayed with new name	System
Take Attendance	Update	User	Mark and update the attendances of students	Attendance of the student is updated and saved	System
View Past Attendance	View	User	Display the list of past attendance	Display the details of the past attendance	User
View Statistical record of student	View	User	Display the statistical details of the past records	Analyze the performance based on statistical	User

			of the students	record of students	
User Logout	Remove	User	User account is logged out	User logout of the app	User

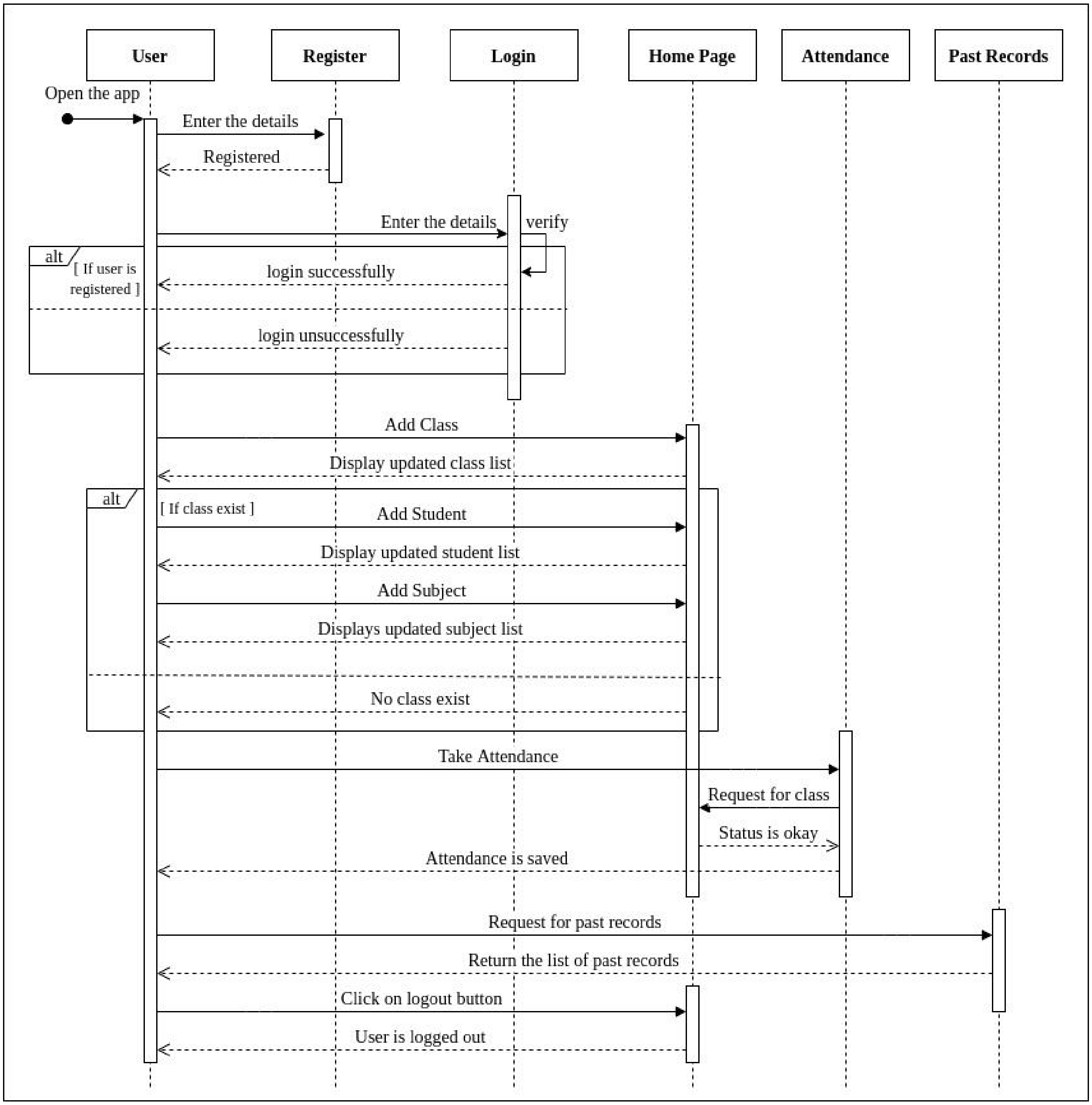
CLASS DIAGRAM



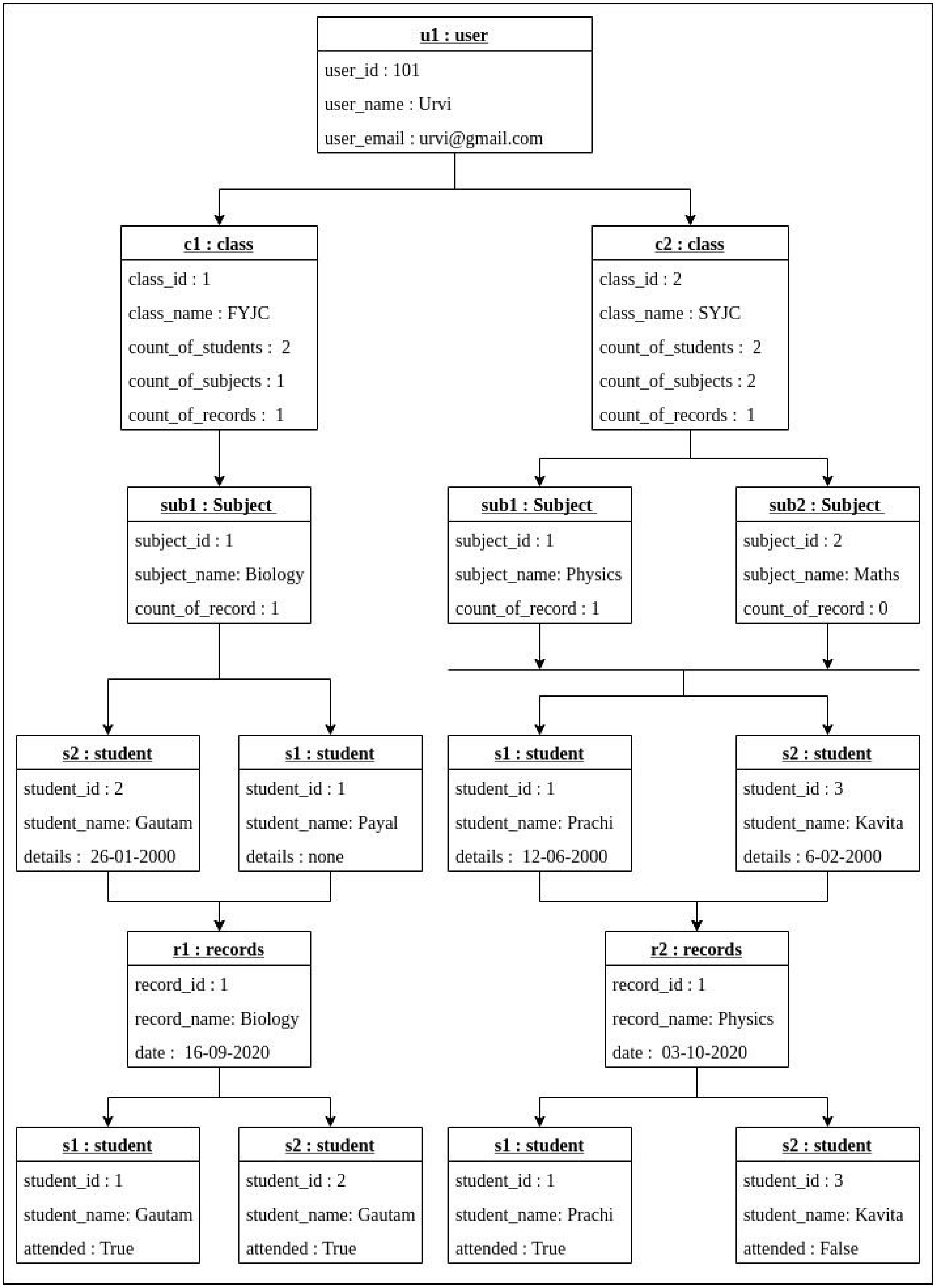
USE CASE DIAGRAM



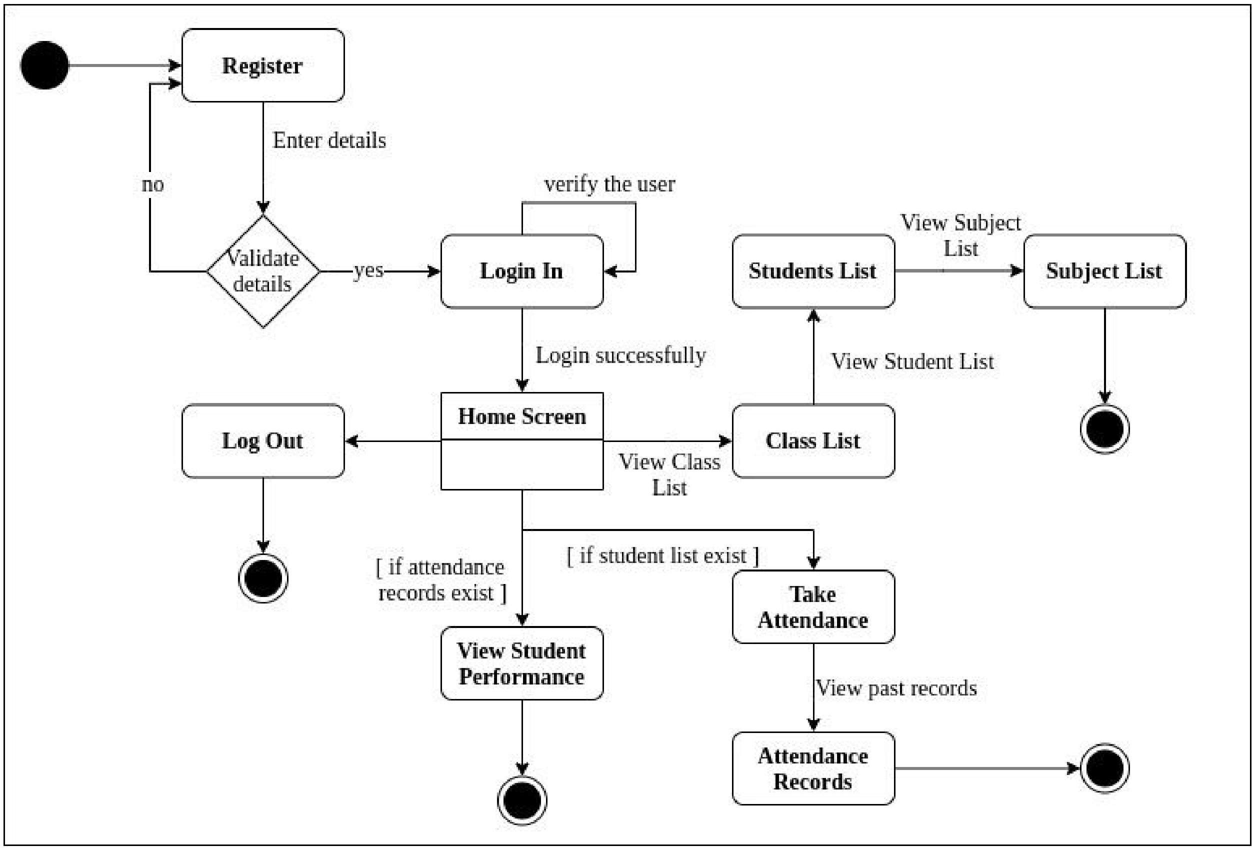
SEQUENCE DIAGRAM



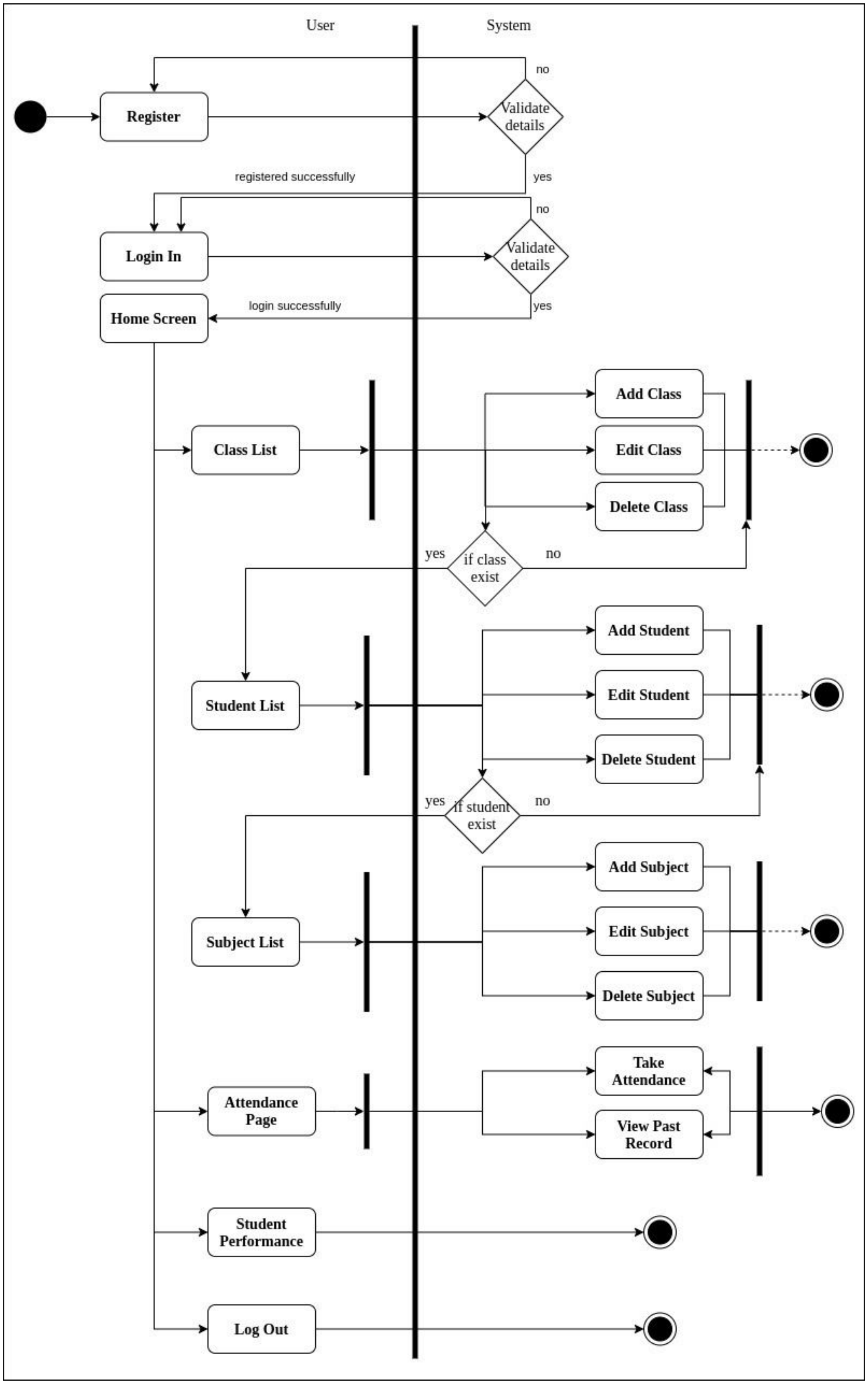
OBJECT DIAGRAM



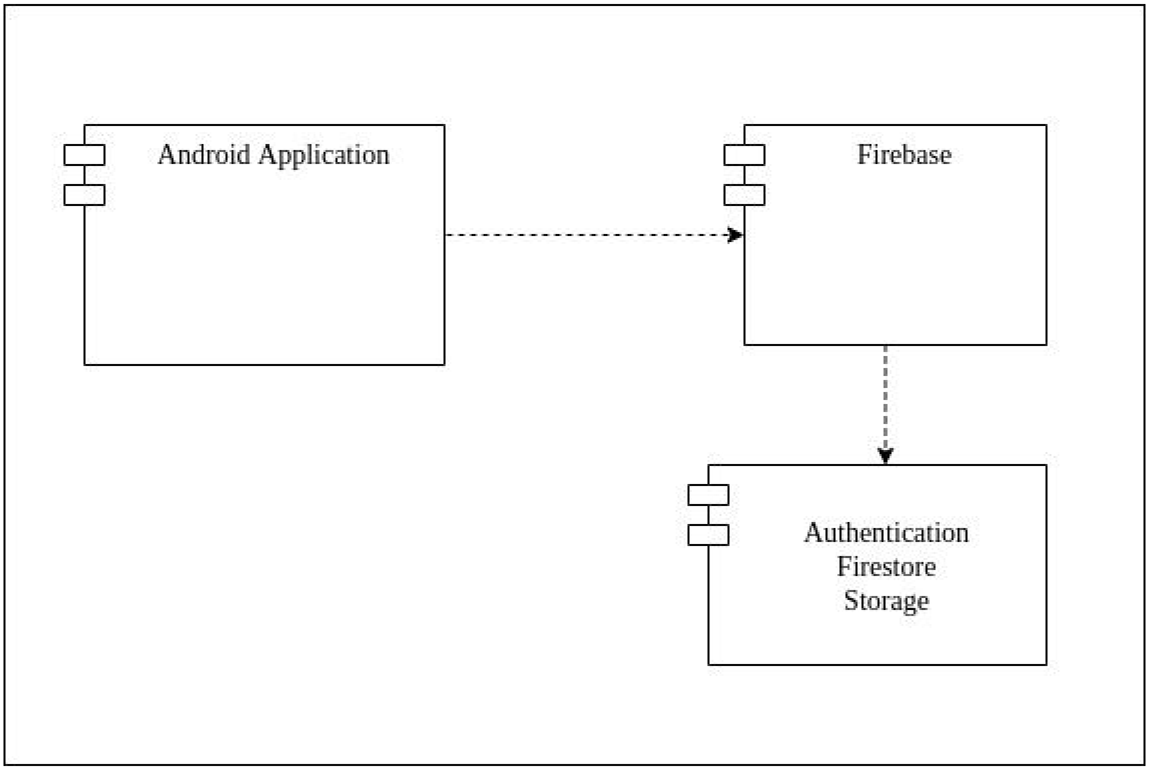
STATE CHART DIAGRAM



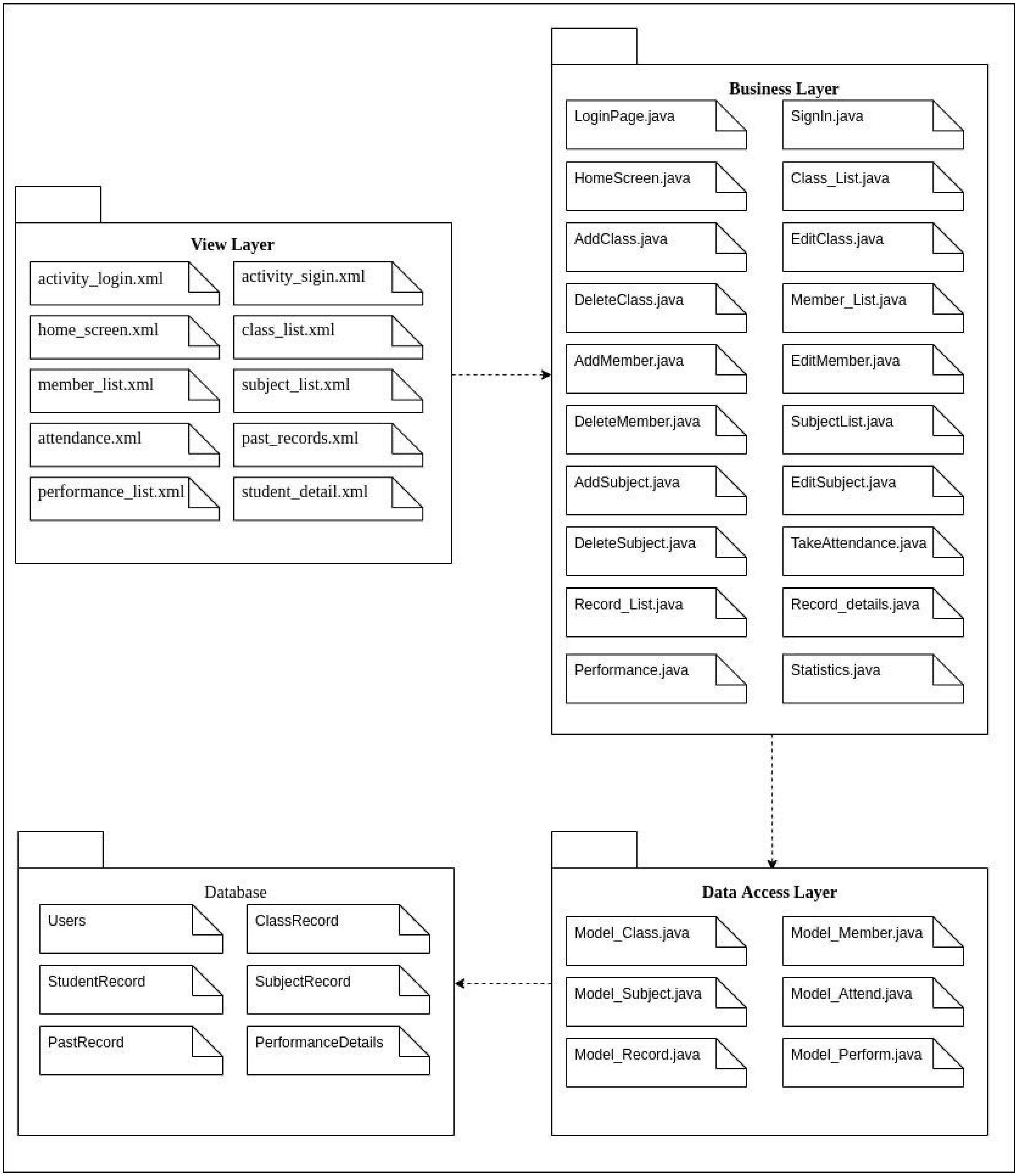
ACTIVITY DIAGRAM



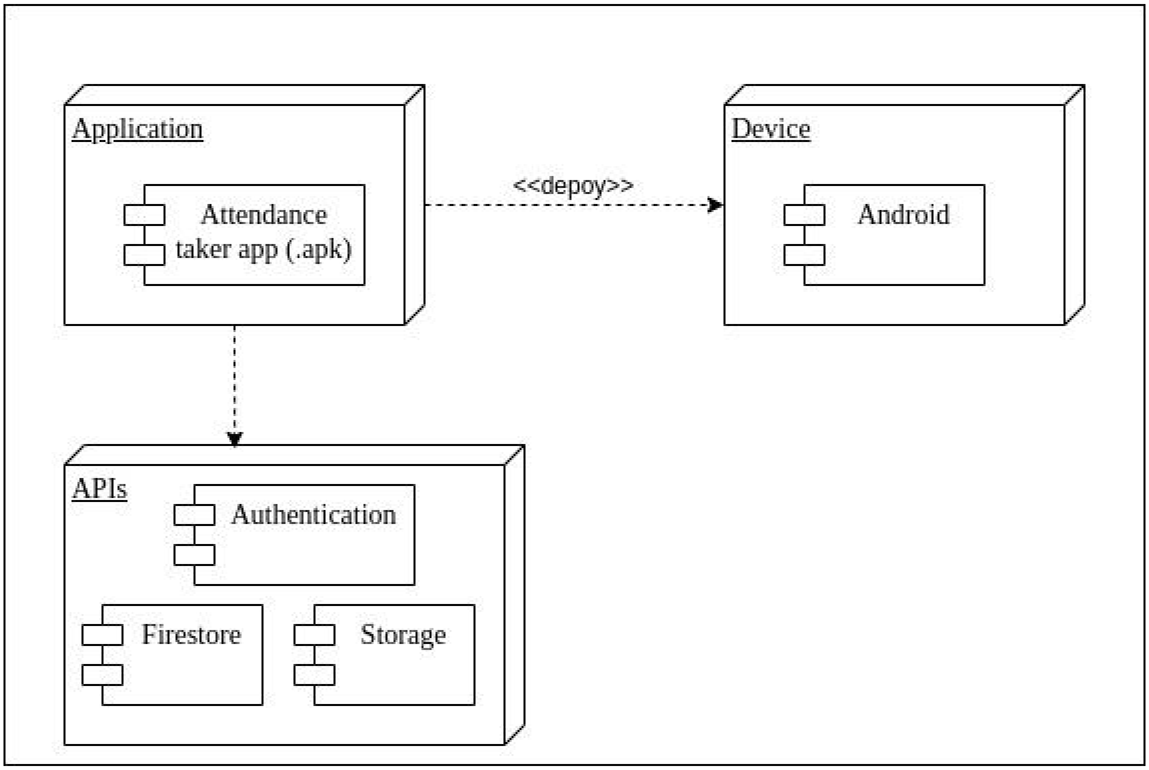
COMPONENT DIAGRAM



PACKAGE DIAGRAM



DEPLOYMENT DIAGRAM



SYSTEM CODING

VALIDATIONS

The following validations are used in the Android application -

1. **Sign Up**
   - It checks that none of the fields are empty.
   - While entering the details, username, email id and password are validated.
   - It validates whether the email id is unique and username does not exist in the database.
   - It checks if the password has a minimum 6 characters.
2. **Login**
   - It checks that none of the fields are empty.
   - It checks whether the given email id exists in the database.
   - It verifies the password given by the user is correct or not.
3. **Forget Password**
   - Forget password does not allow any empty fields.
   - Username and email id of the user is checked against the database.
4. **Add Class, Students, Subject**
   - All the fields must be filled.
5. **Edit Class, Students, Subject**
   - Empty field is not accepted.

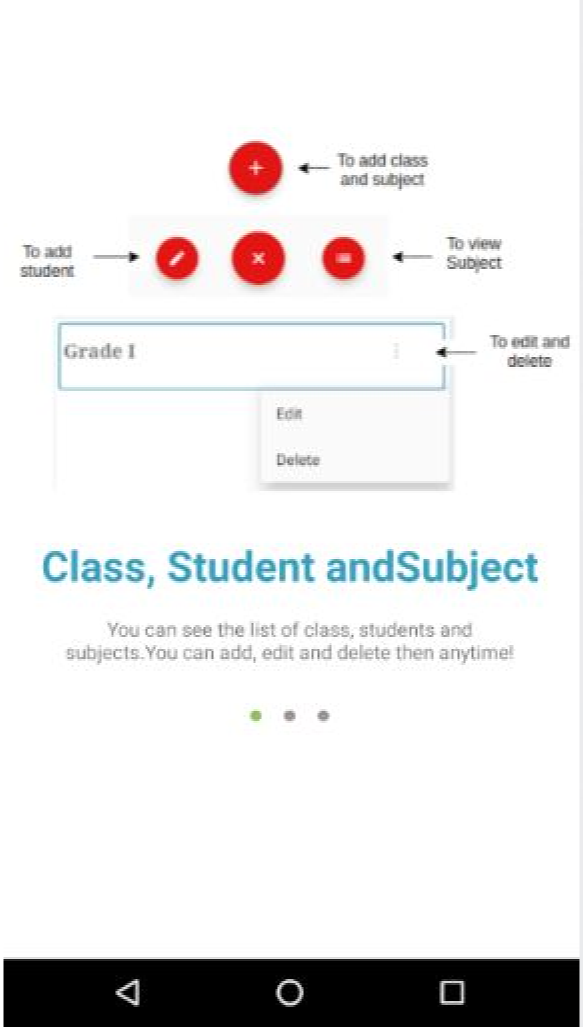
Rule	Parameter	Description
Required	No	Returns FALSE if any element is empty.
Valid Email	No	Returns FALSE if any element contains a invalid email address.
Valid Password	No	Returns FALSE if any element has a password less than 6 characters.

TEST CASES, TEST DATA AND TEST RESULT

Sr no.	Module Name	Form Name	Test condition	Step Procedure	Input test data	Expected Output	Actual Output	Result
1.	Sign up	sign.java	Sign up button	Enter username	username : abc	Enter all the fields correctly	Enter all the fields correctly	Pass
				Enter email id	Email id: abc	Enter a valid email id	Enter a valid email id	Pass
				Enter password	Password : abc	Should contains minimum 6 characters	Should contains minimum 6 characters	Pass
2.	Login	login.java	Login button	Enter email id	Email id: fm@gmail.com	User with fm@gmail.com does not exist	User with fm@gmail.com does not exist	Pass
				Enter password	Password: kmc	Wrong password	Wrong password	Pass
3.	Login	login.java	Google Sign in button	Select the account	Account: none	Select a gmail account	Select a gmail account	Pass
4.	Forget Password	resetpassword.java	Yess button	Enter email id	Email: fm@gmail.com	User with fm@gmail does not exist	User with fm@gmail does not exist	Pass
5.	Add Class	add_class.java	Add button	Enter class name	Class name:	Enter the class name	Enter the class name	Pass
6.	Edit Class	edit_class.java	Edit button	Enter class name	Class name:	Enter the class name	Enter the class name	Pass

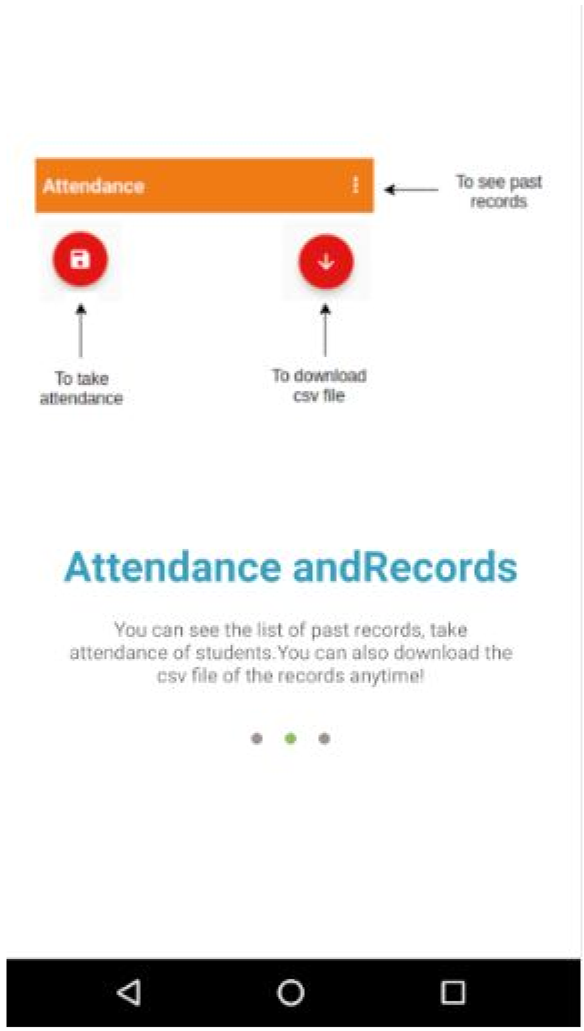
7.	Add student	add_member.java	Add button	Enter student id	Student id:	Enter all the details	Enter all the details	Pass
				Enter student name	Student name:	Enter all the details	Enter all the details	Pass
				Enter student details	Student details:	Enter all the details	Enter all the details	Pass
8.	Edit student	edit_student.java	Edit button	Enter student id	Student id:	Enter all the details	Enter all the details	Pass
				Enter student name	Student name:	Enter all the details	Enter all the details	Pass
				Enter student details	Student details:	Enter all the details	Enter all the details	Pass
9.	Add Subject	add_subject.java	Add button	Enter subject name	Subject name:	Enter the subject name	Enter the subject name	Pass
10	Edit Subject	edit_subject.java	Edit button	Enter subject name	Subject name:	Enter the subject name	Enter the subject name	Pass

SCREEN AND REPORT LAYOUTS



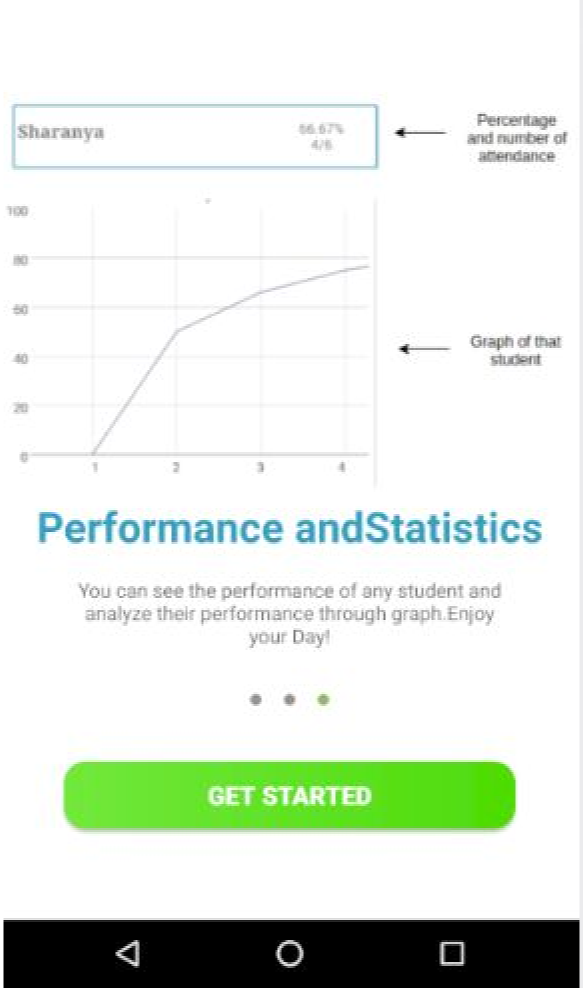
Class, Student andSubject

You can see the list of class, students and subjects.You can add, edit and delete then anytime!



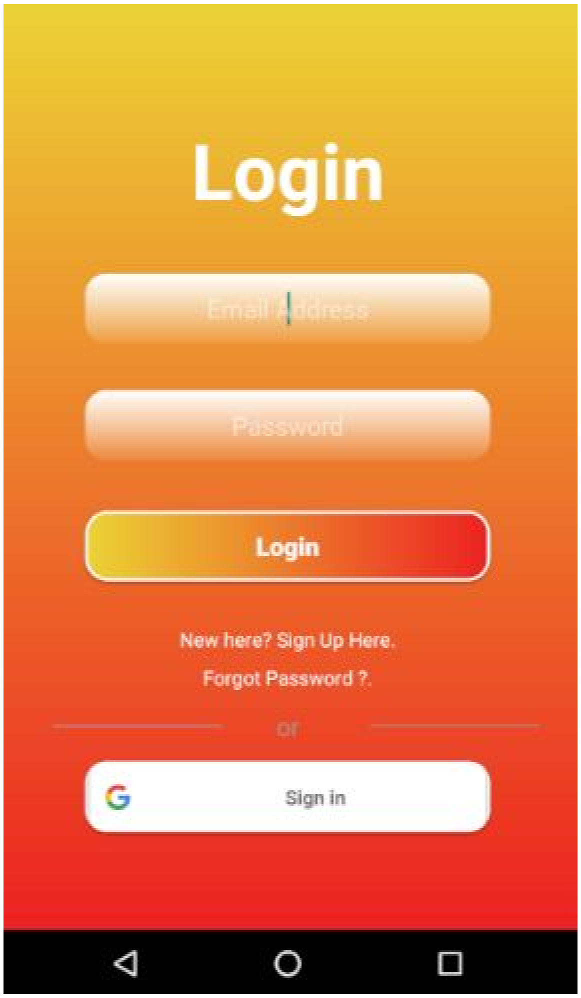
Attendance andRecords

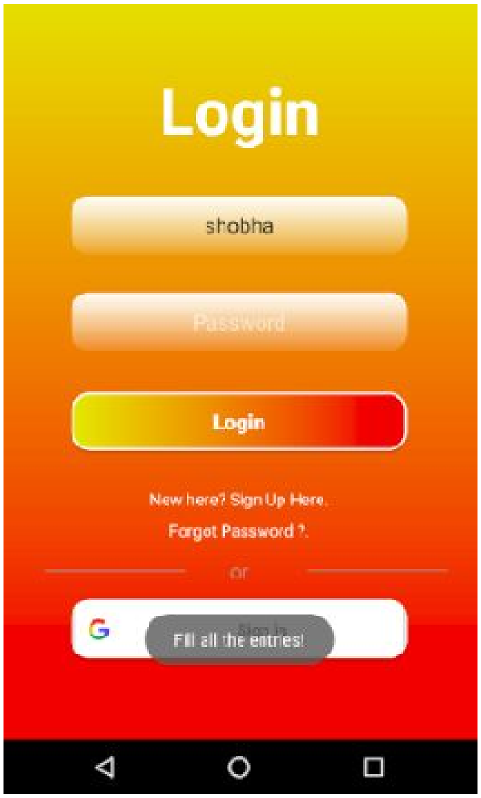
You can see the list of past records, take attendance of students.You can also download the csv file of the records anytime!



Performance andStatistics

You can see the performance of any student and analyze their performance through graph.Enjoy your Day!



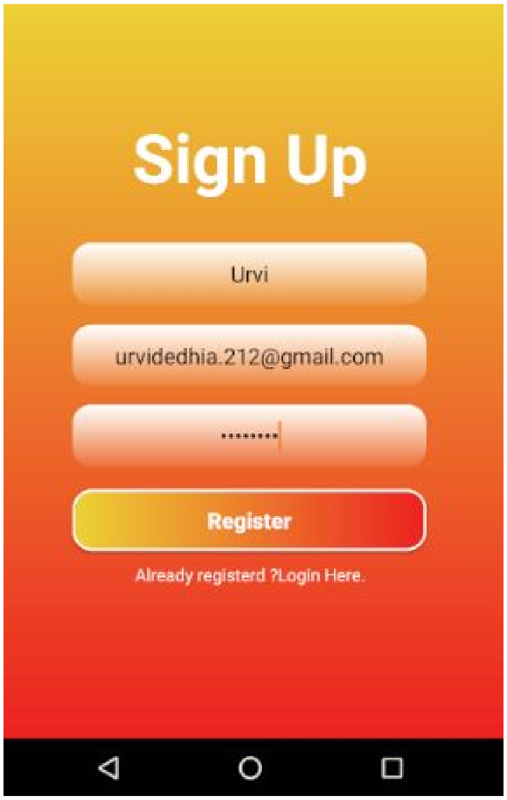


Login

shobha

Password

Fill all the entries!

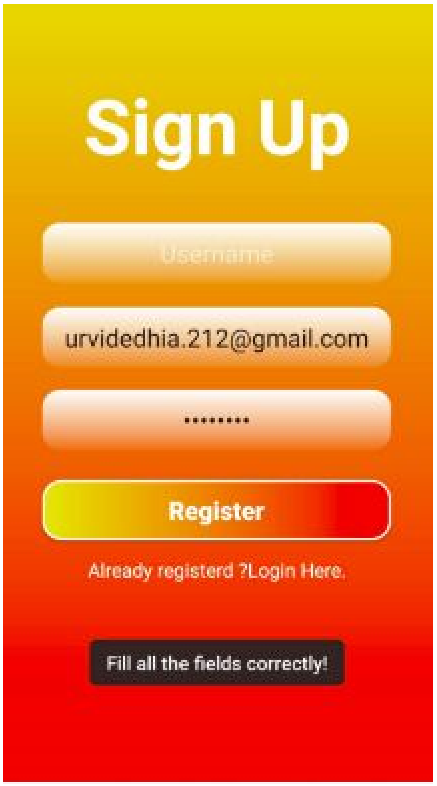


Sign Up

Urvi

urvidedhia.212@gmail.com

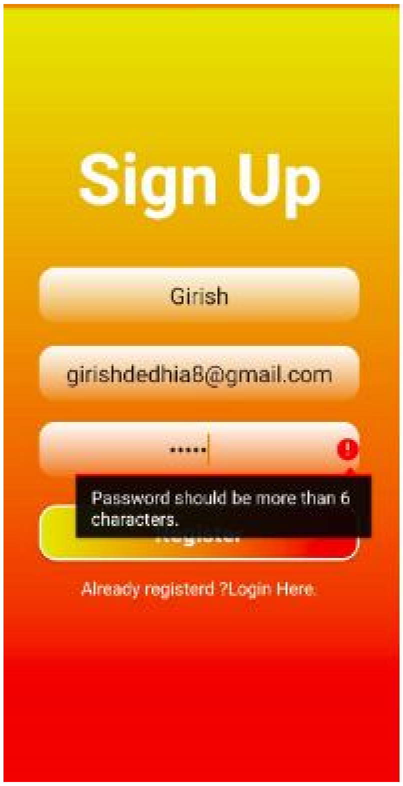
....



urvidedhia.212@gmail.com

•••••••••

Fill all the fields correctly!

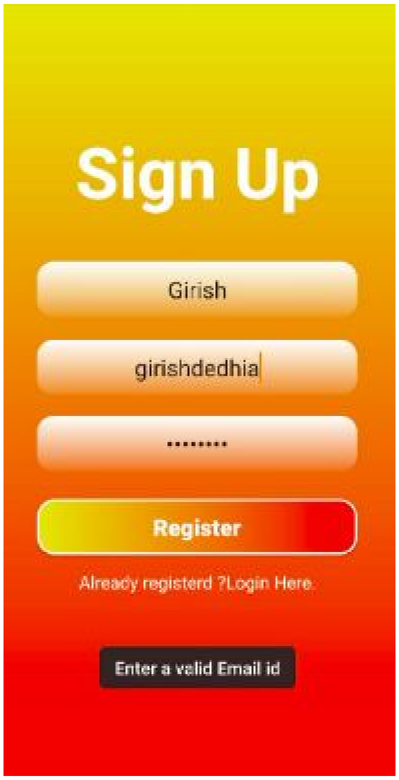


Girish

girishdedhia8@gmail.com

••••••

Password should be more than 6 characters.

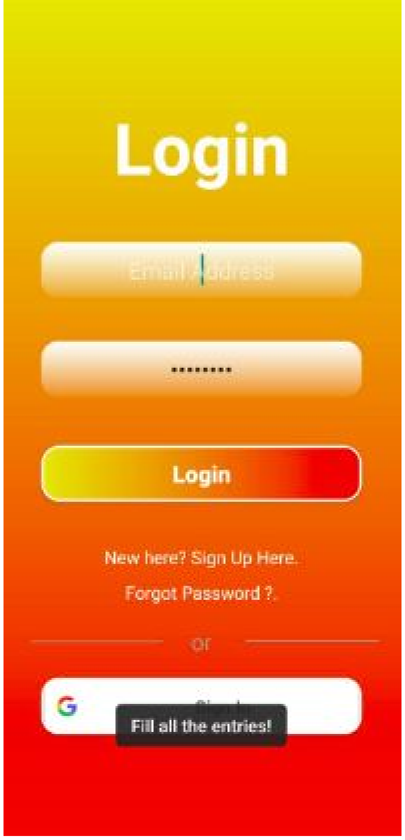


Girish

girishdedhia

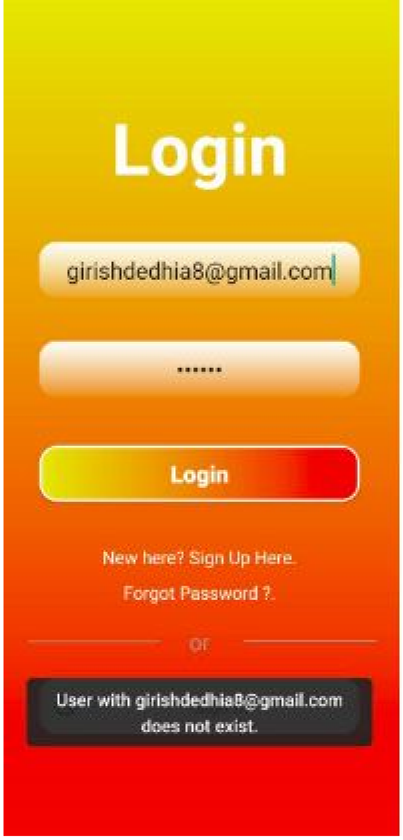
•••••••••

Enter a valid Email id



•••••••••

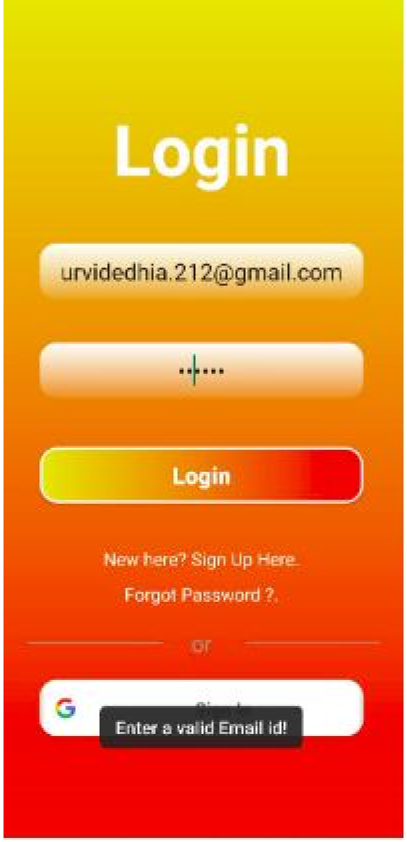
Fill all the entries!



girishdedhia8@gmail.com

••••••

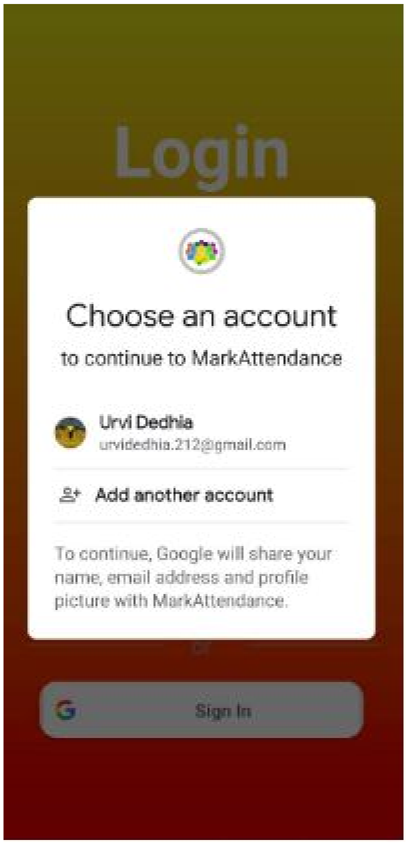
User with girishdedhia8@gmail.com does not exist.



urvidedhia.212@gmail.com

••••••

Enter a valid Email id!



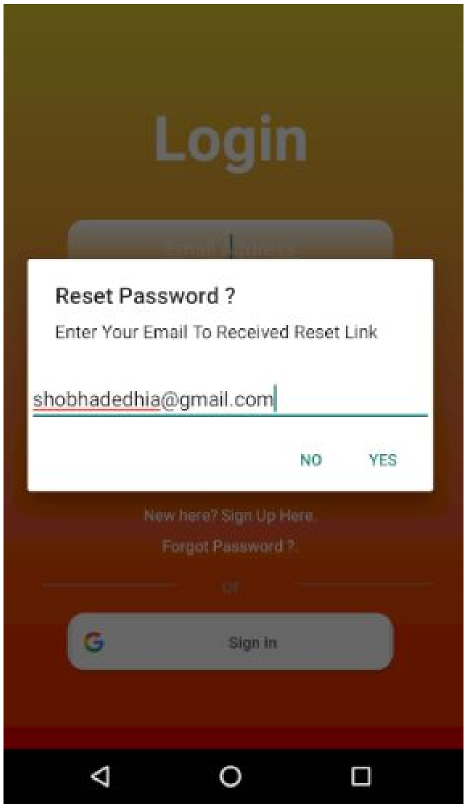
Choose an account

to continue to MarkAttendance

Urvi Dedhia
urvidedhia.212@gmail.com

Add another account

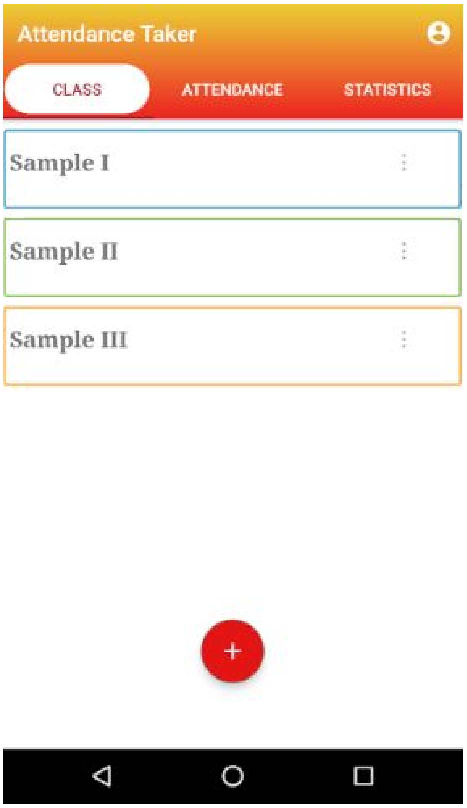
To continue, Google will share your name, email address and profile picture with MarkAttendance.

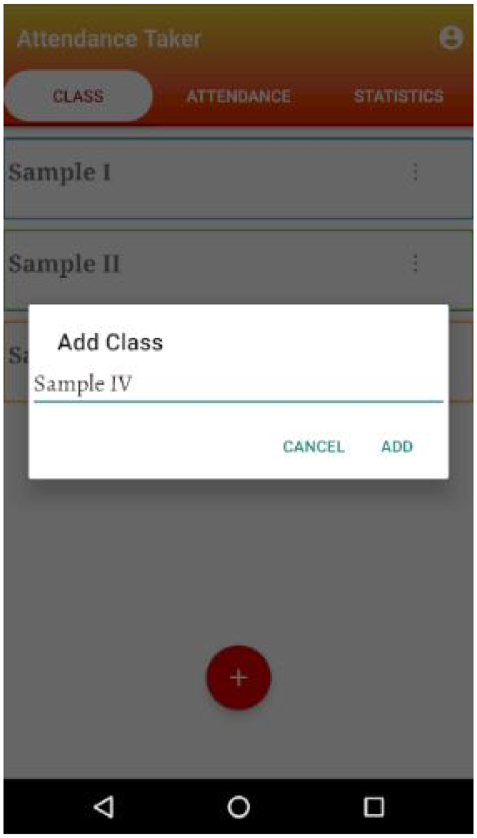


Reset Password ?

Enter Your Email To Received Reset Link

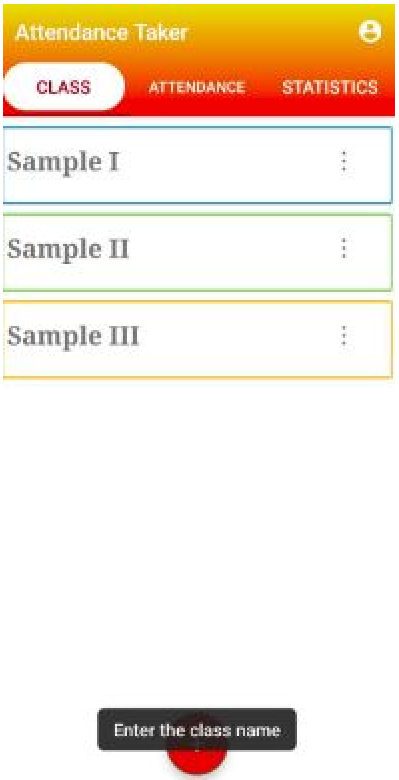
shobhadedhia@gmail.com



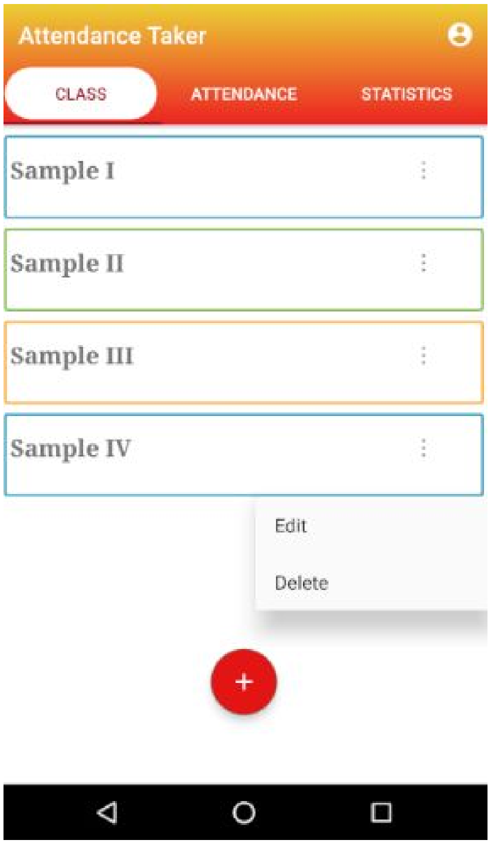


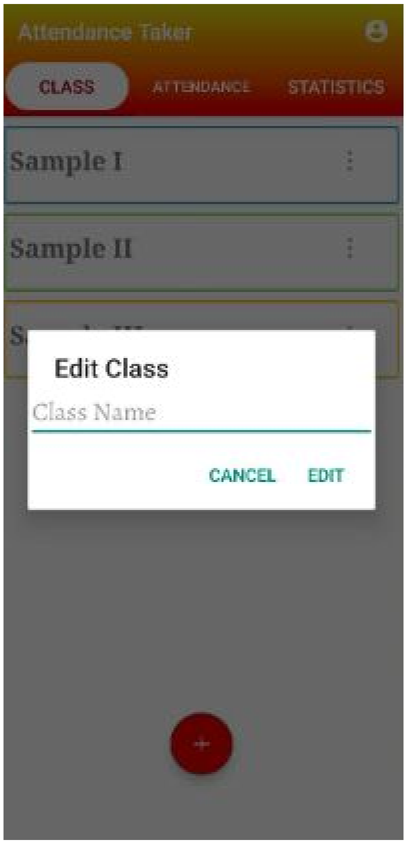
Add Class

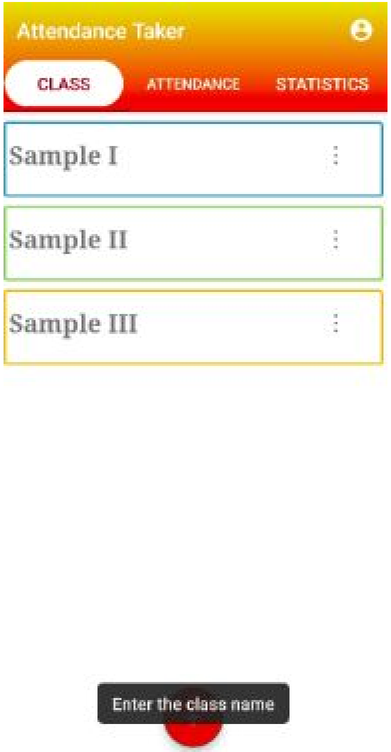
Sample IV



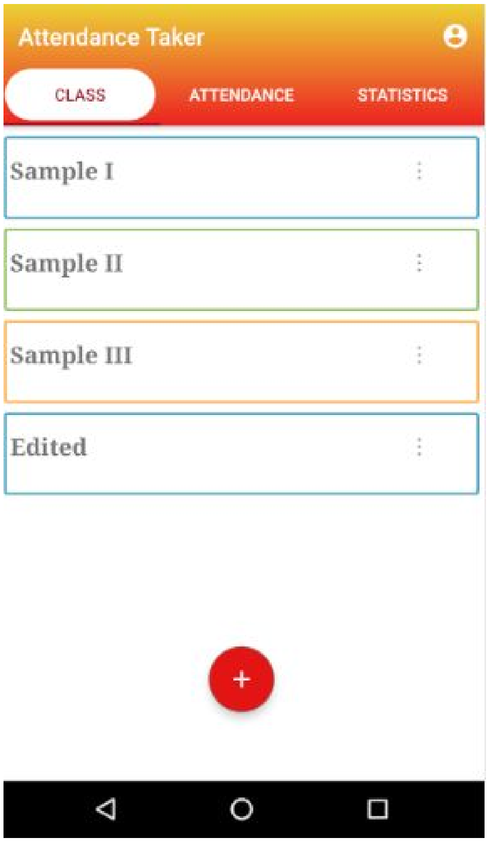
Enter the class name

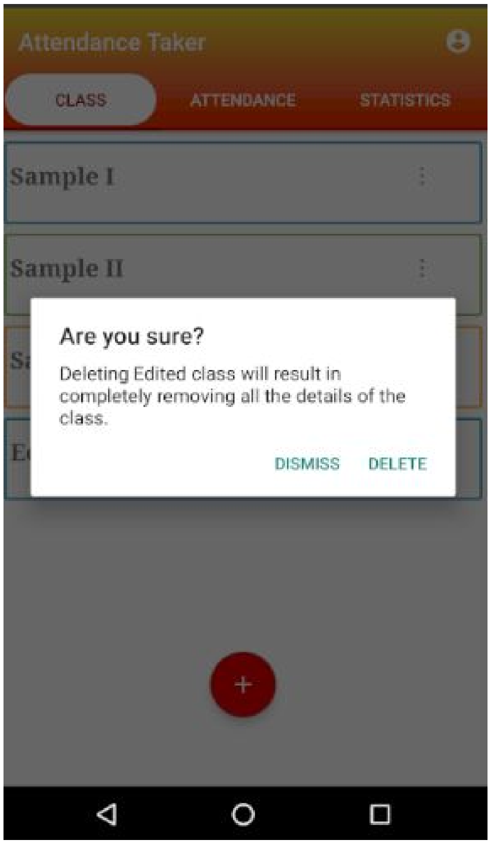






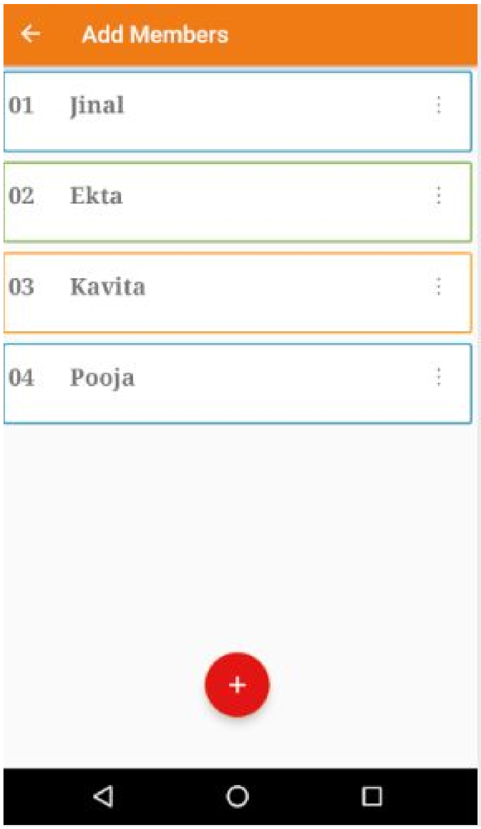
Enter the class name





Are you sure?

Deleting Edited class will result in completely removing all the details of the class.



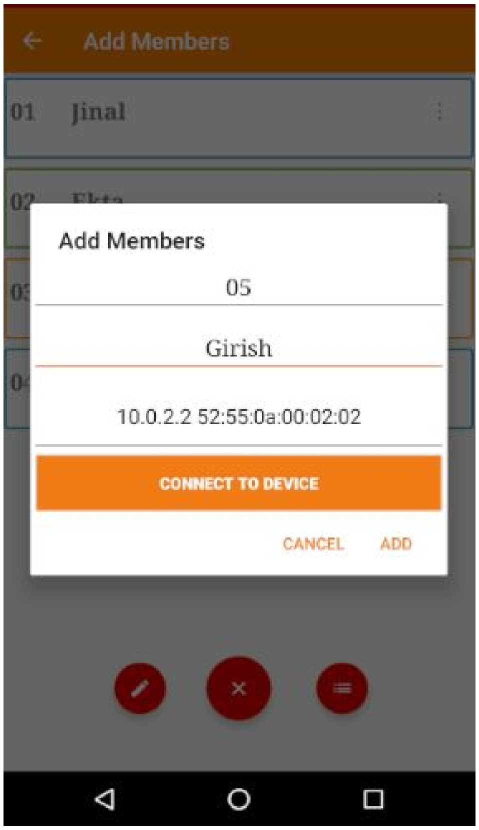
Add Members

01 Jinal

02 Ekta

03 Kavita

04 Pooja



Add Members

01 Jinal

Add Members

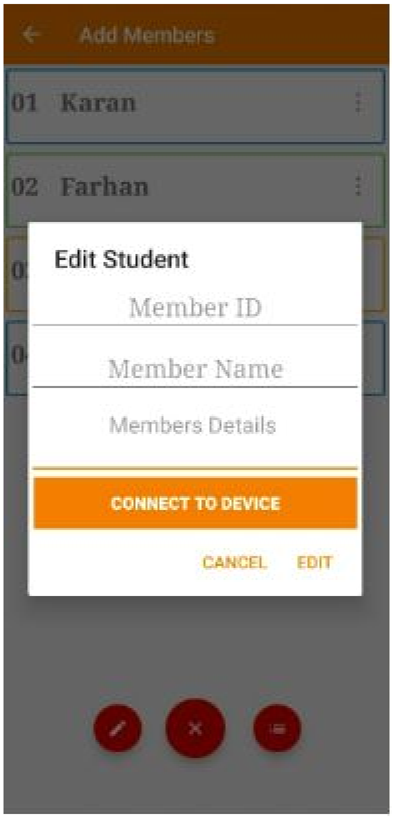
05

Girish

10.0.2.2 52:55:0a:00:02:02

CONNECT TO DEVICE

CANCEL ADD



Add Members

01 Karan

02 Farhan

Edit Student

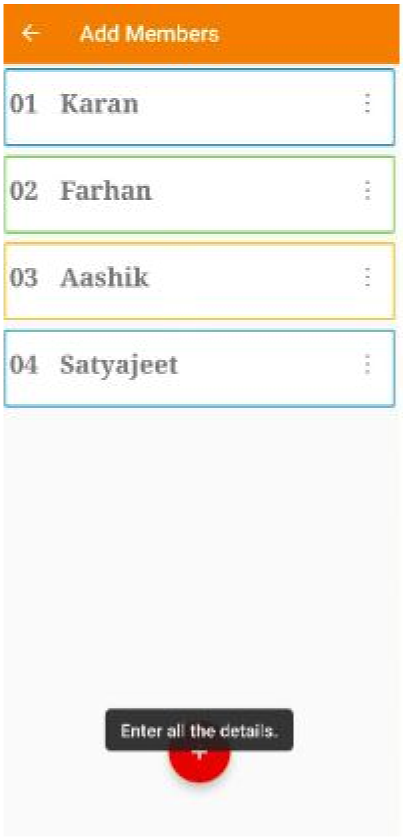
Member ID

Member Name

Members Details

CONNECT TO DEVICE

CANCEL EDIT



Add Members

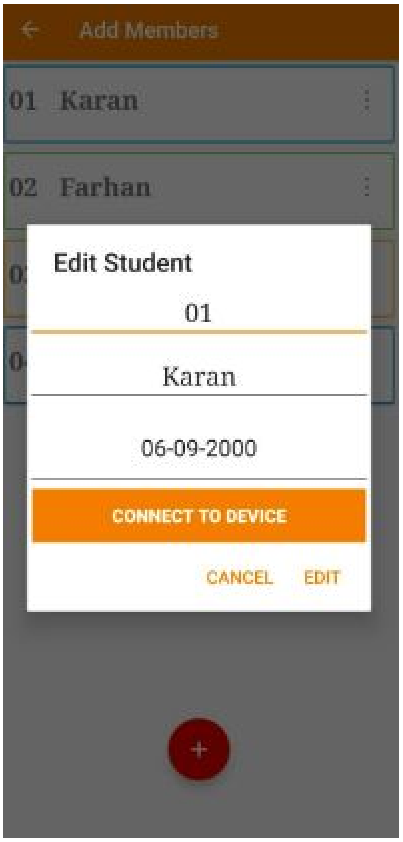
01 Karan

02 Farhan

03 Aashik

04 Satyajeet

Enter all the details.



Add Members

01 Karan

02 Farhan

Edit Student

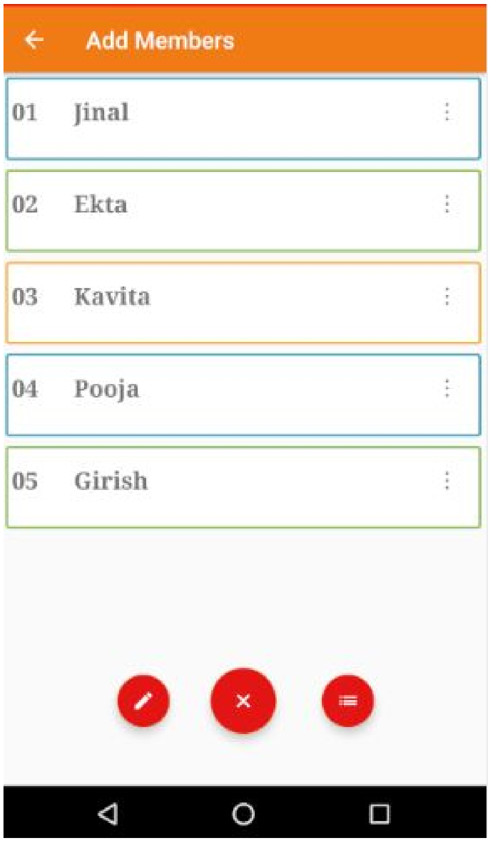
01

Karan

06-09-2000

CONNECT TO DEVICE

CANCEL EDIT



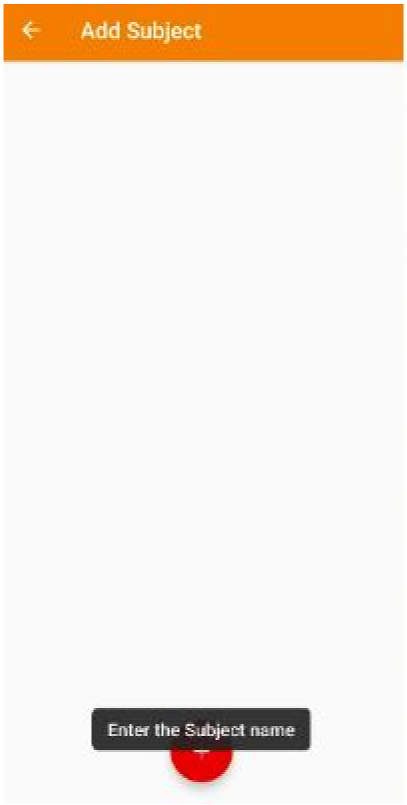
01 Jinal

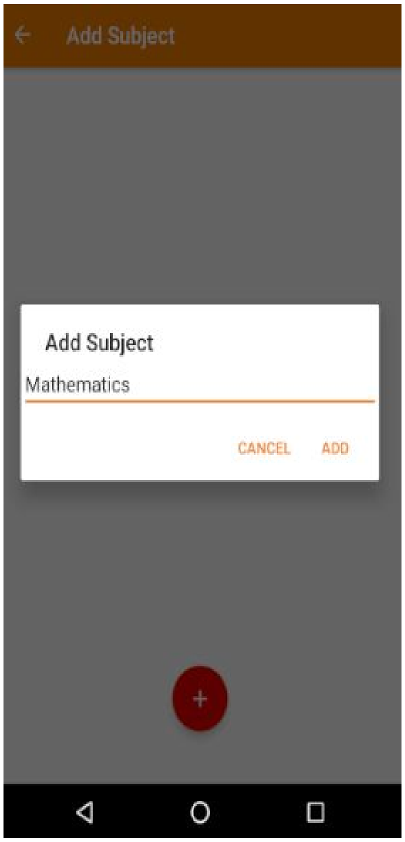
02 Ekta

03 Kavita

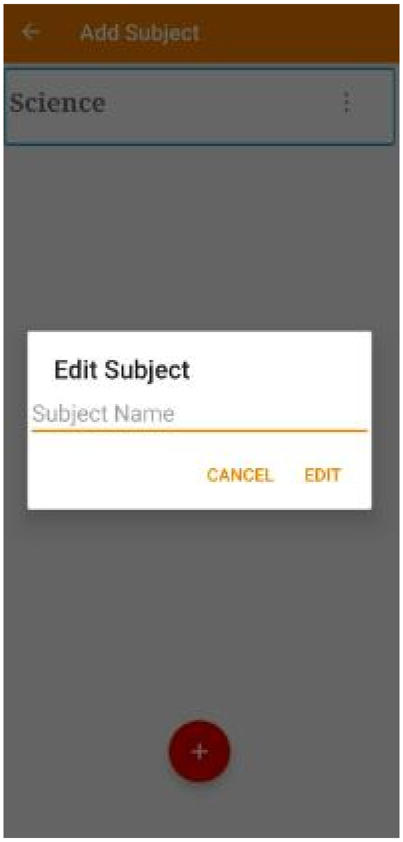
04 Pooja

05 Girish

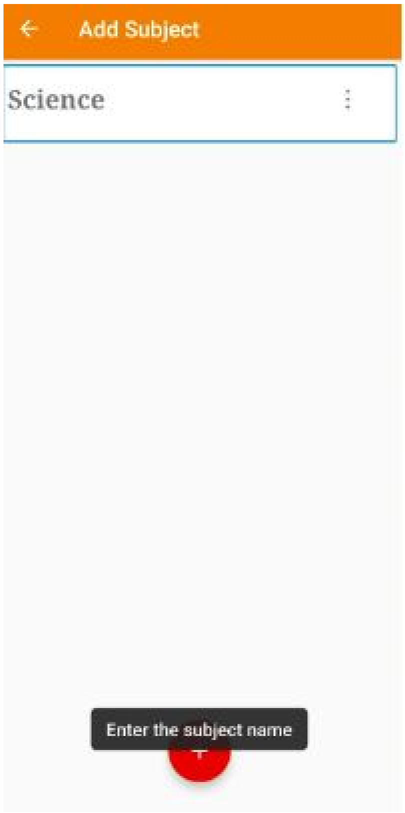




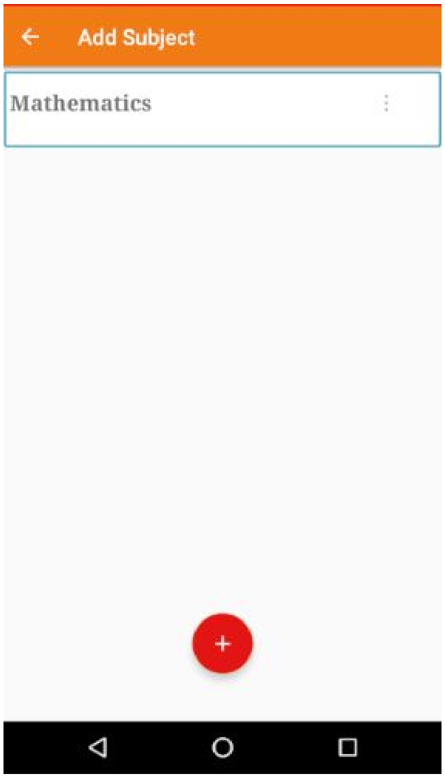
Mathematics



Science

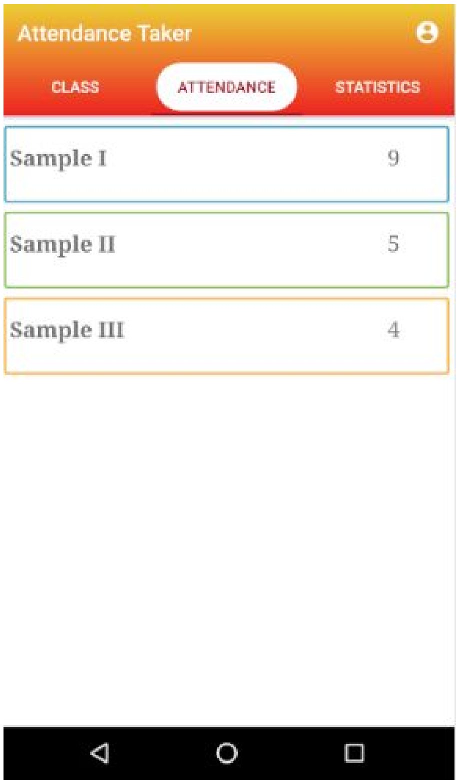


Science



Add Subject

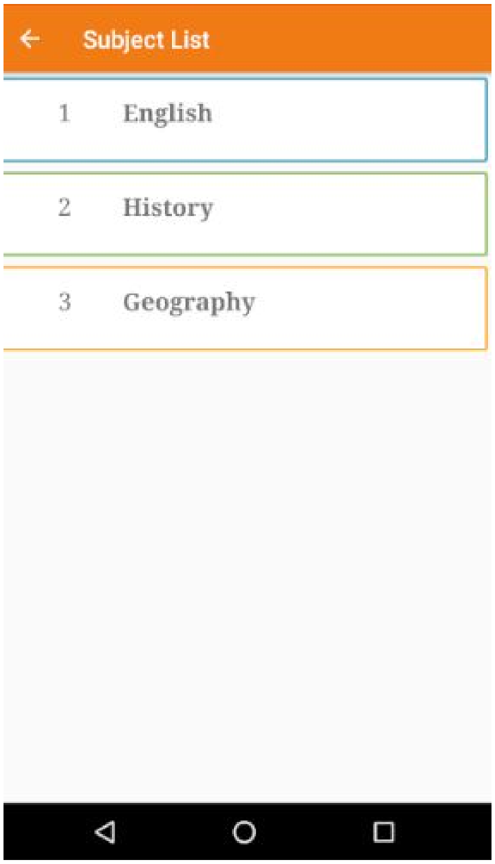
Mathematics



Attendance Taker

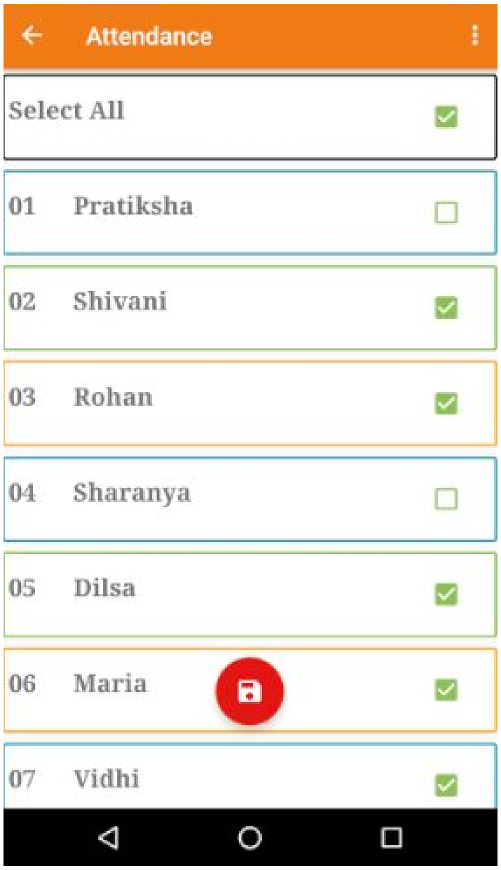
CLASS ATTENDANCE STATISTICS

Sample I	9
Sample II	5
Sample III	4



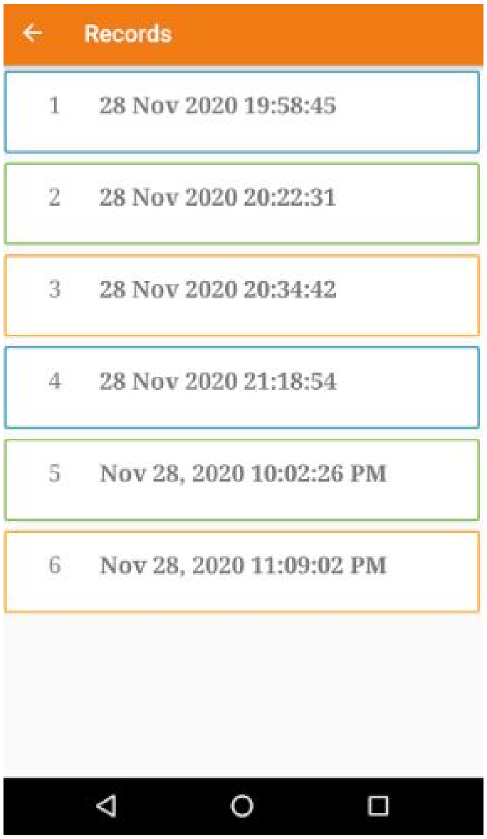
Subject List

1	English
2	History
3	Geography



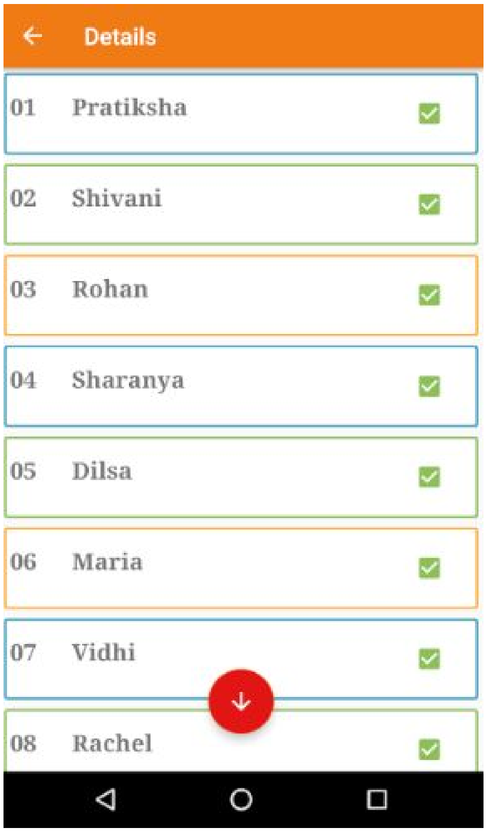
Attendance

Select All		☑
01	Pratiksha	□
02	Shivani	☑
03	Rohan	☑
04	Sharanya	□
05	Dilsa	☑
06	Maria	☑
07	Vidhi	☑



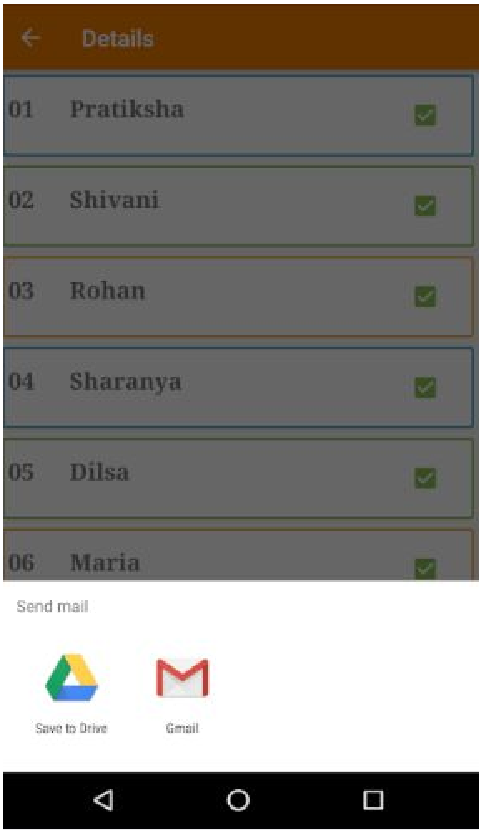
Records

1	28 Nov 2020 19:58:45
2	28 Nov 2020 20:22:31
3	28 Nov 2020 20:34:42
4	28 Nov 2020 21:18:54
5	Nov 28, 2020 10:02:26 PM
6	Nov 28, 2020 11:09:02 PM



Details

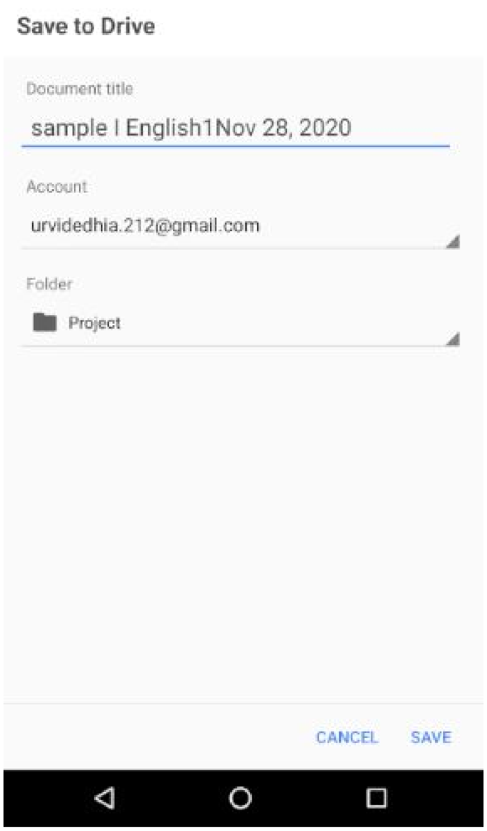
01	Pratiksha	☑
02	Shivani	☑
03	Rohan	☑
04	Sharanya	☑
05	Dilsa	☑
06	Maria	☑
07	Vidhi	☑
08	Rachel	☑



Details

01	Pratiksha	☑
02	Shivani	☑
03	Rohan	☑
04	Sharanya	☑
05	Dilsa	☑
06	Maria	☑

Send mail

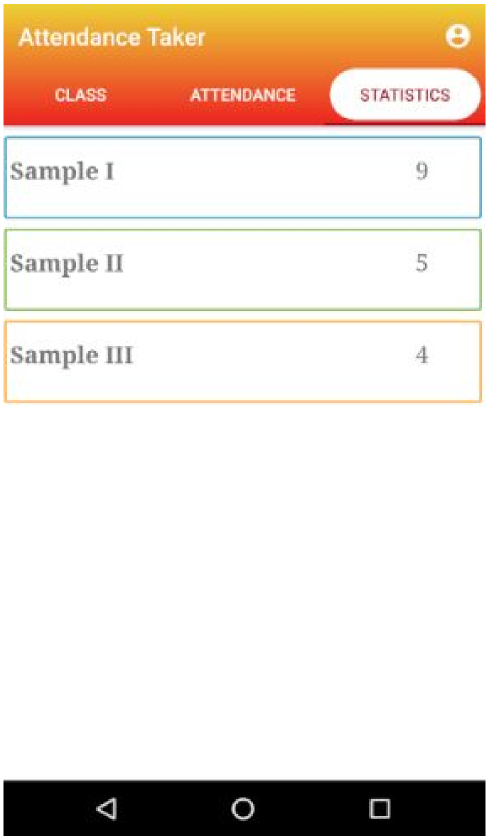


Save to Drive

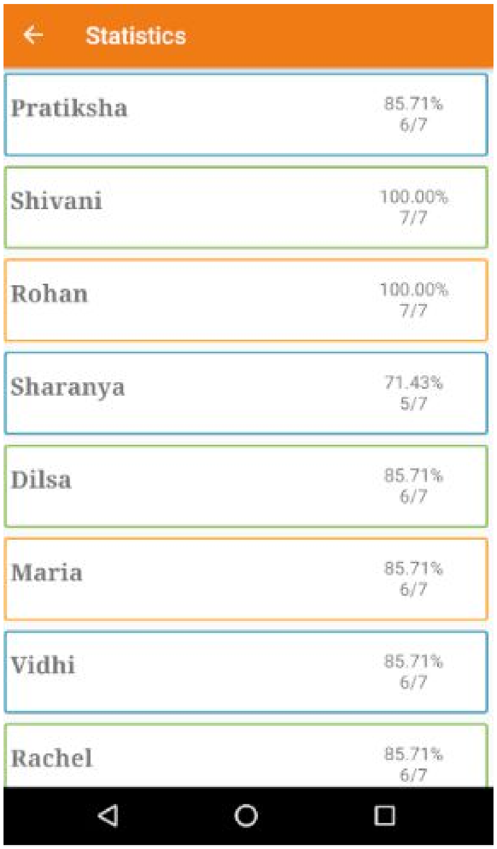
Document title
sample I English1Nov 28, 2020

Account
urvidedhia.212@gmail.com

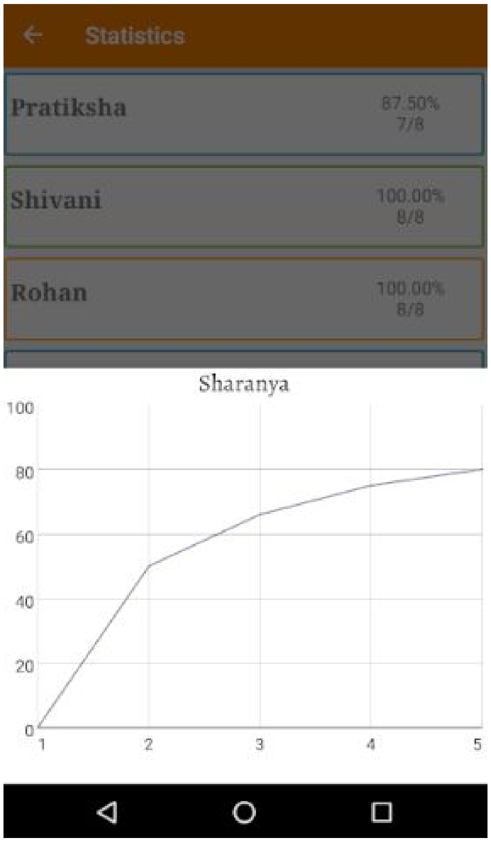
Folder
Project



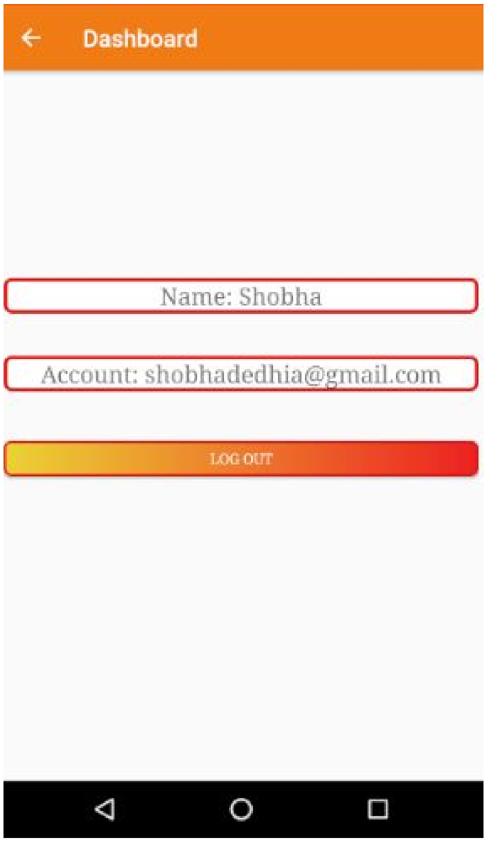
Sample I	9
Sample II	5
Sample III	4



Pratiksha	85.71%	6/7
Shivani	100.00%	7/7
Rohan	100.00%	7/7
Sharanya	71.43%	5/7
Dilsa	85.71%	6/7
Maria	85.71%	6/7
Vidhi	85.71%	6/7
Rachel	85.71%	6/7



Pratiksha	87.50%	7/8
Shivani	100.00%	8/8
Rohan	100.00%	8/8



Name: Shobha

Account: shobhadedhia@gmail.com

SYSTEM IMPLEMENTATION AND UPLOADING

ANDROID STUDIO

1. Project in Android Studio contains one or more modules with source code files and resource files. Types of modules include:
   - Android app modules
   - Library modules
   - Google App Engine modules
2. All the build files are visible at the top level under **Gradle Scripts** and each app module contains the following folders:
   - **manifests**: Contains the AndroidManifest.xml file.
   - **java**: Contains the Java source code files, including JUnit test code.
   - **res**: Contains all non-code resources, such as XML layouts, UI strings, and bitmap images.

FIREBASE

1. To use firebase, one needs to have a gmail account.
2. Connect your android project directly from IDE to firebase.
3. Create a new Project and select the platform (Android/iOS/Website).
4. Follow the steps of registration.
5. Connect firestore, authentication, storage sections to firebase.
6. Run the application to verify the integration.

PUBLISHING THE APPLICATION

1. The following is a high-level overview of the steps you might need to take to sign and publish a new app to Google Play:
   - Generate an upload key and keystore
   - Sign your app with your upload key
   - Opt in to Play App Signing
   - Upload your app to Google Play
   - Prepare & roll out release of your app

CONCLUSION AND FUTURE SCOPE

1. I have created an android application for teachers to take their attendance easily and efficiently.
2. The class list, student list and subject list details are stored on the cloud, hence its memory efficiency and makes it easier to track the records and details.
3. Attendance is taken and the file is stored on the cloud and when the teacher requires the hardcopy or .csv file of that record, they can download it anytime.
4. Student attendance performance is measured in percentage wise, also the number of attended lectures out of total attendance is displayed.
5. For better understanding of the performance, graphs are used in the application.

FUTURE ENHANCEMENT

1. Soon Admin Application will be developed to directly interact with users and databases.
2. In future this application will be used at the college level, where the account must be created as college and teachers under that account shall take attendance.
3. It will be set to make this application also available for iOS based phones and will be operated on a website.

REFERENCE AND BIBLIOGRAPHY

1. https://firebase.google.com/docs/firestore
2. https://firebase.google.com/docs/auth
3. https://firebase.google.com/docs/storage
4. Youtube channel - https://www.youtube.com/c/CodinginFlow
5. Stackoverflow.com
6. https://medium.com
7. https://developers.google.com/identity/sign-in/android/start-integrating#next_steps
8. https://www.youtube.com/watch?v=HtwDXRWjMcU&list=PLgqDEp0rc2fjdhm33sZ4IRWk3B2U4d7Hp&index=5&t=1192s
9. https://www.youtube.com/watch?v=umCX1-Tq25k&list=PLgqDEp0rc2fjdhm33sZ4IRWk3B2U4d7Hp&index=8&t=607s
10. https://www.youtube.com/watch?v=VDAwbgHoYEA&list=PLgqDEp0rc2fjdhm33sZ4IRWk3B2U4d7Hp&index=12&t=110s