ವಿಶ್ವೇಶ್ವರಯ್ಯ ತಾಂತ್ರಿಕ ವಿಶ್ವವಿದ್ಯಾಲಯ, ಬೆಳಗಾವಿ
VISVESVARAYA TECHNOLOGICAL UNIVERSITY - BELAGAVI

A MINI PROJECT REPORT ON

"DROID ASSIST"

Submitted to Visvesvaraya Technological University in partial fulfilment of the requirement for the award of degree of

Bachelor of Engineering

In

Computer Science and Engineering.

Submitted by:

Mr. Mohammed Moin Mirza	4JN18CS048
Mr. Zaheer Abbas	4JN18CS128
Mr. Chethan B A	4JN19CS403

Under the Guidance of:

Dr. Chetan K .R Ph.D.
Associate Prof., CSE

Department of Computer Science & Engineering
Jawaharlal Nehru New College of Engineering
Shivamogga - 577 204

August 2021

National Education Society ®

Jawaharlal Nehru New College of Engineering

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the Mini-project Report entitled

DROID ASSIST

Submitted by:

M.MOIN MIRZA 4JN18CS048
ZAHEER ABBAS 4JN18CS128
CHETHAN B A 4JN19CS403

Students of 6th semester B.E under the supervision and guidance towards the partial fulfilment of the requirement for award of degree of the Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belgaum during the year 2020 – 2021.

Signature of Guide and HOD:

Dr. Chetan K R Ph.D.	**Dr. Poornima K M** Ph.D.,
Associate Prof., CSE	Professor & Head, CSE

Name of Examiners	**Signature with Date**
1.	
2.	

ABSTRACT

In our project we will be creating a student friendly application called Droid Assist. The ideology behind the application is accumulating all the resources, important links, keeping track of attendance, scores and timetable at a single place to create a hassle free online education experience. The student is allowed to login using Email id, USN and password. Those without an account can register into the application by providing the required details present in the sign up form. We will be using Firebase as our Backend for services such as Authentication and cloud database Firestore. As of now the following features will be made available to the user: Home Dashboard of the user where it will tell the classes for that day. Timetable for a particular class. Resources for a particular subject. Also the user can track his attendance for all the subjects that he has enrolled in. He also can track his scores of all the official assessments done during the semester and has access to the scores he secured during the assessment. All the data will be stored using the Firestore cloud database.

ACKNOWLEDGEMENT

We would like to acknowledge our profound gratitude to all those who have helped in implementing this mini project.

We are grateful to our institution **Jawaharlal Nehru New College of Engineering** and **Department of Computer Science & Engineering** for imparting us the knowledge with which we can do our best.

We would like to thank our beloved guides **Dr. Chetan K R, Associate Professor** and **Dr. Jalesh Kumar, Professor, Dept. of CS&E, JNNCE** who have helped us a lot in making this project and for their continuous encouragement and guidance throughout the project work.

We would like to thank **Dr. K.M Poornima, HOD of CS&E Dept.** and **Dr. Manjunatha P, The Principal, JNNCE, Shimoga** for all their support and encouragement.

Finally, we also would like to thank the whole teaching and non-teaching staff of the Computer Science and Engineering Department.

Thanking you all,

Project Associates,

Mohammed Moin Mirza 4JN18CS048

Zaheer Abbas 4JN18CS128

Chethan B A 4JN19CS403

Table of Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

1.1 Overview

Online classes are being conducted all over the world. Offline classes in colleges and schools are shifting towards online mode because of Covid-19. A lot of issues have risen to this situation. The main thing is for the students to stay in contact with their faculties and attend the online classes. Online classes come with their own set of problems. The main problem is that all the resources such as class links, class notes and class videos are in a scattered place. And also to keep the students accountable to attend the classes, the attendance of each student must be tracked.

Thus, there is a requirement of a software that can solve the above problems mentioned. The idea is to build a mobile app that supports a good set of devices and can be accessible on any device. Droid Assist, the app our team has built solves all these problems and is available for all mobile devices.

1.2 Applications

Droid Assist provides students with the necessary tools that are necessary for them to attend online classes without any problems. Thus Droid Assist application is helpful to each and every student that is taking online classes. Also it helps all the necessary resources provided from the college staff keep things at a single place.

Thus the main application is in the Education Industry, where the online classes are being taken. This app helps the students and the college staff as a tool, so that they can focus on their work rather than worrying about the resources. This also makes the online class experience hassle free. In addition to the resources, a student can check the attendance and scores of each subject. Using the data of the attendance and the marks scored by the students the app can help students to attend classes given the links of the online classes.

This does not apply just for a single college or a school. The app can be expanded further to help more students of all the colleges and schools, as the app provides all the general features that are required by all school and college students.

1.3 Problem Statement

Online classes come with their own set of problems. As all colleges are adapting the online class mode. The following problems arise, students not being able to find the resources of the subjects provided from their teachers at a single place. Students not being able to keep accountable for attending the classes. Also there is a problem of students not being able to keep track of their scores that are scored during their tests conducted. These are the main issues that are to be concerned. Other problems include teachers not being able to connect with the students virtually on a single tool to share the resources of the classes they conduct. It becomes a bit difficult to keep the students accountable for not being punctual to the class.

1.4 Objectives

- Build a student dashboard kind of app to help students attend classes on a daily basis.
- Help students with the resources required for each class to make online classes more adaptable.
- Help keep track of each student's attendance.
- Help keep track of scores scored by each student in each of their enrolled subjects.
- Keep track of the time table of each class a student is enrolled in.

1.5 Overview of Android

Android is an open source, Linux-based software stack created for a wide array of devices and form factors. It follows a layer of stack structure of components. Fig 1.1 gives an overview of the Android Architecture. Main components of Android and its features include the following:

- **The Linux Kernel -**
    - The foundation of the Android platform is the Linux kernel. The Android Runtime (ART) relies on the Linux kernel for underlying functionalities such as threading and low-level memory management.

- Using a Linux kernel allows Android to take advantage of key security features and allows device manufacturers to develop hardware drivers for a well-known kernel.

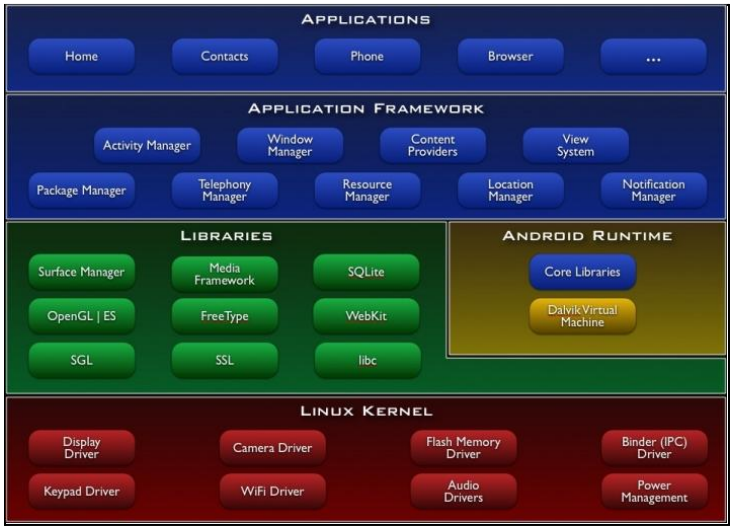

Fig. 1.1 Android Architecture

- **Android Runtime and supporting Libraries -**
  - Prior to Android version 5.0 (API level 21), Dalvik was the Android runtime.
  - Devices running Android version 5.0 (API level 21) or higher, each app runs in its own process and with its own instance of the Android Runtime (ART). ART is written to run multiple virtual machines on low-memory devices by executing DEX files.
  - Many core Android system components and services, such as ART, are built from native code that require native libraries written in C and C++.

- **Application Framework -**
  - The entire feature-set of the Android OS is available to you through APIs written in the Java language. These APIs form the building blocks you need to create Android apps by simplifying the reuse of core, modular system components and services.

- **System and User Applications** -
    - Android comes with a set of core apps for email, SMS messaging, calendars, internet browsing, contacts, and more. Apps included with the platform have no special status among the apps the user chooses to install.
    - The system apps function both as apps for users and to provide key capabilities that developers can access from their own app.

1.6 Overview of Kotlin

Kotlin is a statically typed, general-purpose programming language developed by JetBrains. Kotlin is object-oriented language, and a “better language” than Java, but still be fully interoperable with Java code. Kotlin is sponsored by Google, announced as one of the official languages for Android Development in 2017.

1.6.1 Kotlin Features

1. Expressive and Concise - Kotlin has an intuitive syntax. Kotlin's modern language features allow us to focus on expressing our ideas and write less boilerplate code.
2. Extension functions - Extension functions allow us to add extra features to the existing components present in the language.
3. Interoperable - Kotlin is 100% interoperable with the Java programming language, we can have as little or as much of Kotlin in your project as you want.
4. Safer code - With @Nullable and @NonNull included in its type system, Kotlin helps us avoid NullPointerExceptions.
5. Structured concurrency - Kotlin coroutines streamline asynchronous programming, making common tasks like network calls and database updates simple and performant.

1.6.2 Advantages Over Java

1. Readability - Kotlin has more readable and precise code which makes it easier to understand the program.

2. Null-Safe - Kotlin is null-safe by default. It does not allow variables to be assigned with a null value.
3. Using Getters and Setters - Koltin assigns setters and getters by itself if we use data classes
4. Immutability - In Kotlin variables are defined using val or var to help developers easily understand which values can be reassigned.

1.6.3 Role in App Development

In 2019 Google announced that Android development will be increasingly Kotlin-first. Android apps written in Kotlin are much faster compared to those written in Java. Using Kotlin for Android development, you can benefit from:

- Less code combined with greater readability. Spend less time writing your code and working to understand the code of others.
- Mature language and environment.
- Kotlin support in Android Jetpack and other libraries. KTX extensions add Kotlin language features, such as coroutines, extension functions, lambdas, and named parameters, to existing Android libraries.
- Interoperability with Java. You can use Kotlin along with the Java programming language in your applications without needing to migrate all your code to Kotlin.
- Code safety. Less code and better readability lead to fewer errors. The Kotlin compiler detects these remaining errors, making the code safe.

Chapter 2

DESIGN AND IMPLEMENTATION

2.1 Functional Requirements

Droid Assist supports the following functionalities:

1. Student Register and Login.
2. Student Dashboard.
3. Class Timetable.
4. Class Resources.
5. Student Attendance.
6. Student Internal Assessment Score.

2.1.1 Student Register and Login

Droid Assist allows new students to create their account. Firebase Authentication using email and password is used to create new student accounts on the platform. Students have to fill in theri email, name, usn, password, branch, semester and section to register on Droid Assist.

Once Registered, the student can login using the same credentials such as his email, usn and password to access the different functionality of the app. Registered credentials such as email, name, usn except for password are all stored in Firestore database.

2.1.2 Student Dashboard (Home)

Once the student logs into Droid Assist, he can access his Dashboard. The Dashboard contains information related to the student. It shows the Name, USN and the department the student belongs to. Navigating to different sections of the app is through the Dashboard screen itself. Dashboard screen provides the classes that the student has for that particular day.

All the data present in the Firestore about the current user is fetched. And only the required details are displayed on the screen. The data present in the Firestore is Name,

Email, USN, Semester, Branch, Section of the User. This data is collected when a user Registers on the App. The user can log out from the dashboard screen.

2.1.3 Class Timetable

Timetable screen can be accessed from the Dashboard screen. The timetable screen shows the timetable of the entire week. There is a drop down from which the user can select the day of the week. The selected day's timetable is shown with the subject code and time of the class.

The timetable depends on the class, semester and branch of the current logged in student. Timetables of all the classes are stored in the Firestore. Based on the current user, the Timetable is fetched and displayed.

2.1.4 Class Resources

Resources screen provides the class link, notes link and the videos link of a particular class. These resources depend on the semester, section and branch of the current logged in user. The user can click on the present button to access these resources. The resources help students keep all the things related to a particular class in a single place.

The Resources of all classes are stored in the Firestore and are fetched based on the current user and displayed to the user. The notes link will be a link to a remote folder such as drive or onedrive.

2.1.5 Student Attendance

Attendance screen provides the list of the following fields for each class. Class attendance percentage, Class subject code, classes attended by the user and the total classes taken. This helps the student keep track of his attendance.

A student's attendance is stored in Firestore. The fetching details depend on the current user's USN. This data is collected when the student registers for the first time on the App.

2.1.6 Student Internal Assessment Score

The Internal Assessment screen shows the score obtained by the current user in a particular assessment. There is a drop down from which the user can select the Assessment details he wants to view. It shows the score followed by the subject code.

The Internal Assessment scores depend on the current user's semester, section, branch and usn. The details are fetched based on this and displayed to the user.

2.2 Design

The Wireframe was made using figma. Figma Link - Design Link . The link provides more insights of the colors and the icons used in our project.

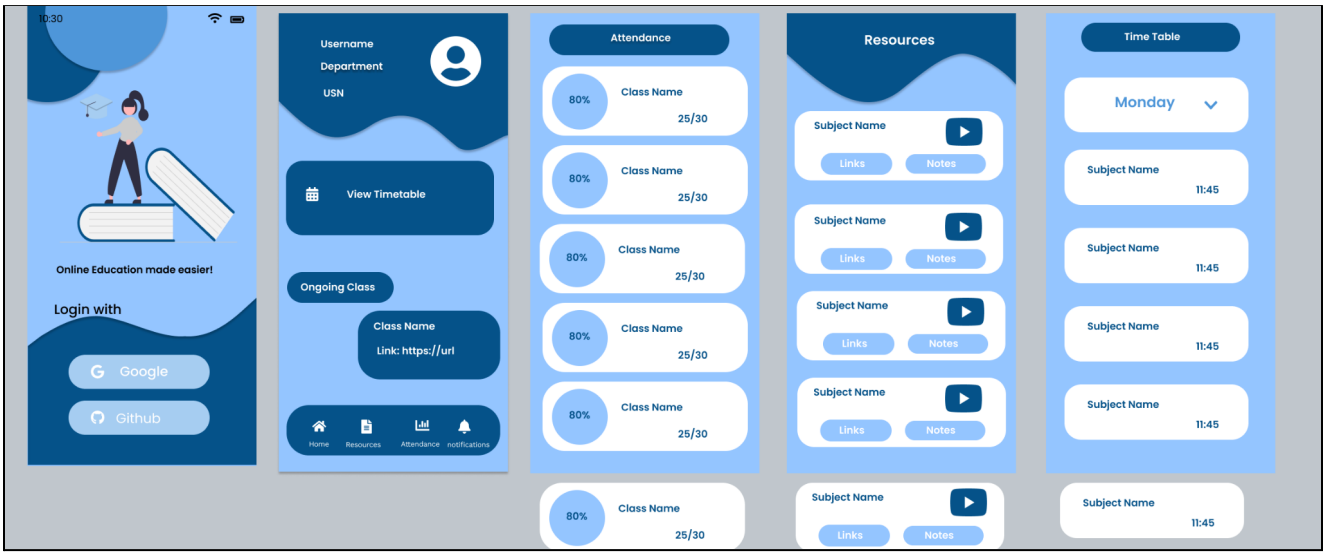

Fig. 2.1 Droid Assist Complete Design

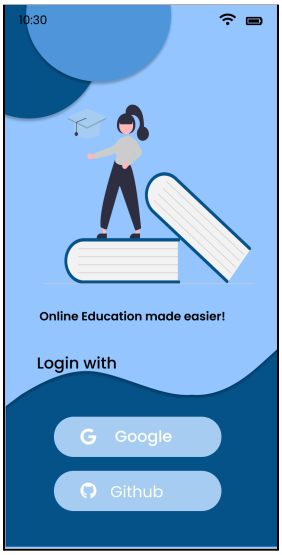

Fig. 2.2.A) Login

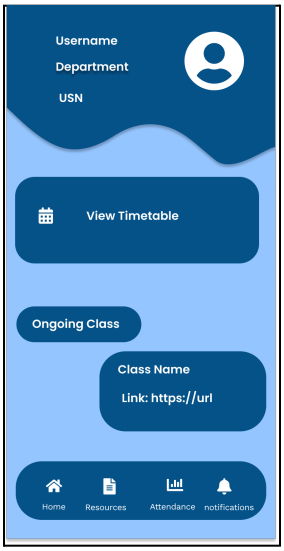

Fig. 2.2.B) Home

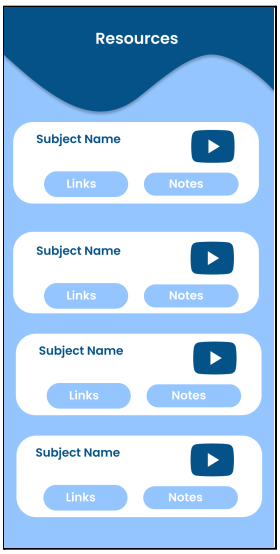

Fig. 2.2.C) Resources

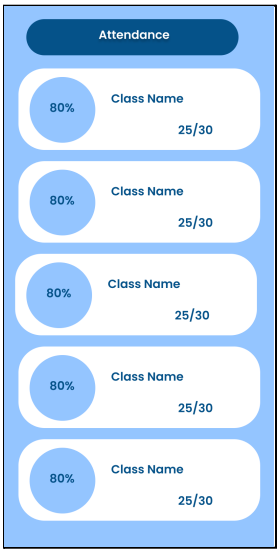

Fig. 2.2.D) Attendance

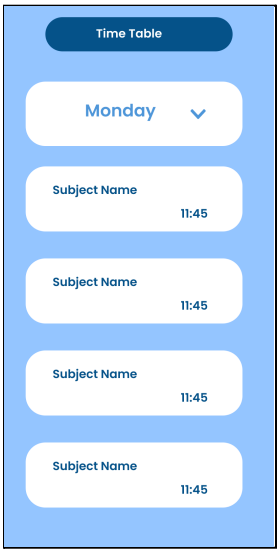

Fig. 2.2.E) Time Table

Fig. 2.2 Design of Each Activity

2.3 Firebase and Firestore

This section provides information related to Firebase and Firestore. Droid Assist uses Firebase as a backend to manage the user and Firestore to store and manage the user's data.

2.3.1 Overview of Firebase

Firebase is a Backend-as-a-Service (Baas). It provides us solutions such as

1. Authentication - Firebase provides us with a ready-to-integrate Authentication system. Users can be authenticated using their Phone number or Email or any other third party Auth provider.
2. Firestore (NoSQL Database) - Firestore is a scalable NoSQL cloud Database to store and sync data.
3. Storage - Firebase provides us a way to store images and other user data in the form of buckets and folders.
4. Machine Learning - Firebase also provides a Machine Learning Kit to help with the work of machine learning.

Droid Assist Uses Authentication and Cloud Firestore Services from Firebase. Authentication is just used as a registration process and the data is stored in Firestore Database.

Features of Firestore:

- Cloud Firestore is a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud.
- It keeps your data in sync across client apps through real time listeners and offers offline support for mobile and web.
- Flexibility - The Cloud Firestore data model supports flexible, hierarchical data structures. Store your data in documents, organized into collections. Documents can contain complex nested objects in addition to subcollections.
- Allows expressive queries to retrieve individual, specific documents or to retrieve all the documents in a collection that match your query parameters. Queries can include multiple, chained filters and combine filtering and sorting.

- Cloud Firestore uses data synchronization to update data on any connected device. However, it's also designed to make simple, one-time fetch queries efficiently.
- Designed to scale - Cloud Firestore brings you the best of Google Cloud's powerful infrastructure: automatic multi-region data replication, strong consistency guarantees, atomic batch operations, and real transaction support

2.3.2 Firestore Structure

Droid Assist has five collections in the Firestore, which are explained below in detail along with their field attributes:

1. **Users Collection -**

The users collection stores the information of the user when the user registers in the app. Each user has his own document in the users collection. The USN of the user is used as the ID for that particular document. Each document has the following fields:

- branch: stores the department of the user as a string.
- email: stores the email of the user as a string.
- imgUrl: stores the image url of the user, empty by default.

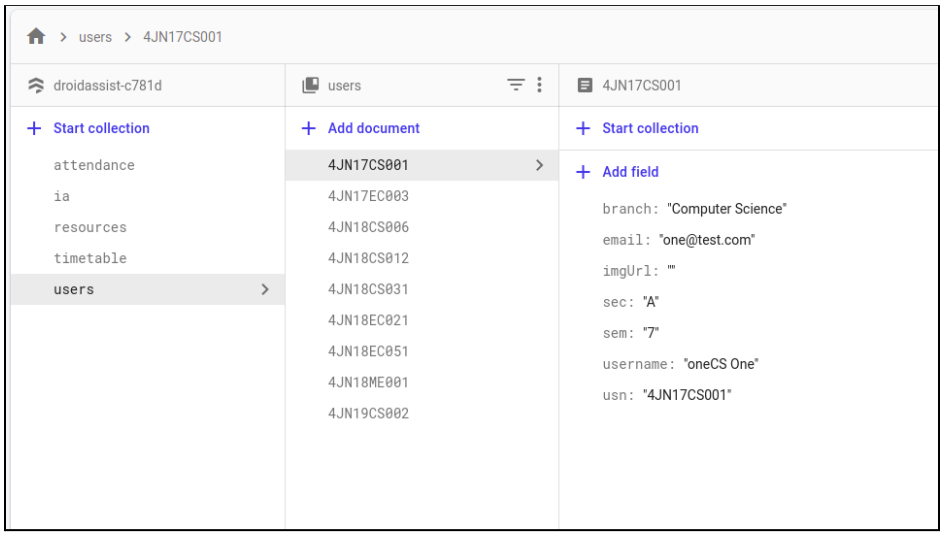

Fig. 2.3 Users Collection

- sec: stores the section of the user.
- sem: stores the semester of the user.
- username: stores the username of the user.
- usn: stores the university seat number of the user.

2. Timetable Collection -

The timetable collection stores the information of an institution's student time table. Each document is a slot for a particular day, section, semester and branch. As each section has its own timetable. Each document has the following fields:

- branch: stores the branch of the timetable slot.
- day: stores the day of the week.
- sec: stores the section to which the timetable belongs.
- sem: stores the semester to which the timetable belongs.
- slot: stores the particular time of the subject.
- subjectCode: code of the subject.

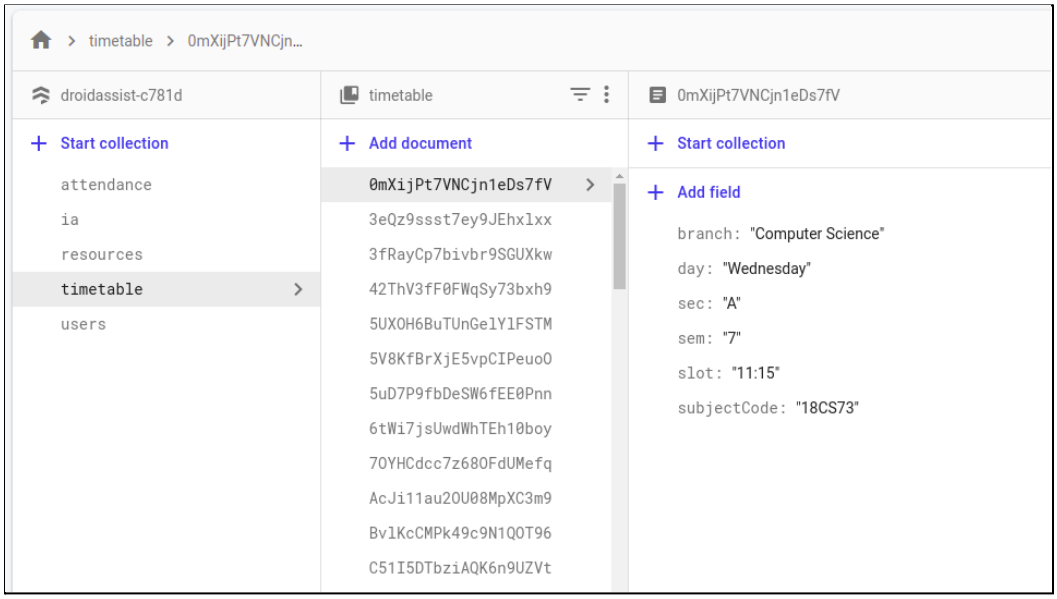

Fig. 2.4 Timetable Collection

3. Resources Collection -

The resources collection stores the information of a particular semester's subject resources such as the class link, class notes and video playlist url. The

collection has a document for each subject of all the semesters. Each document contains the following fields:

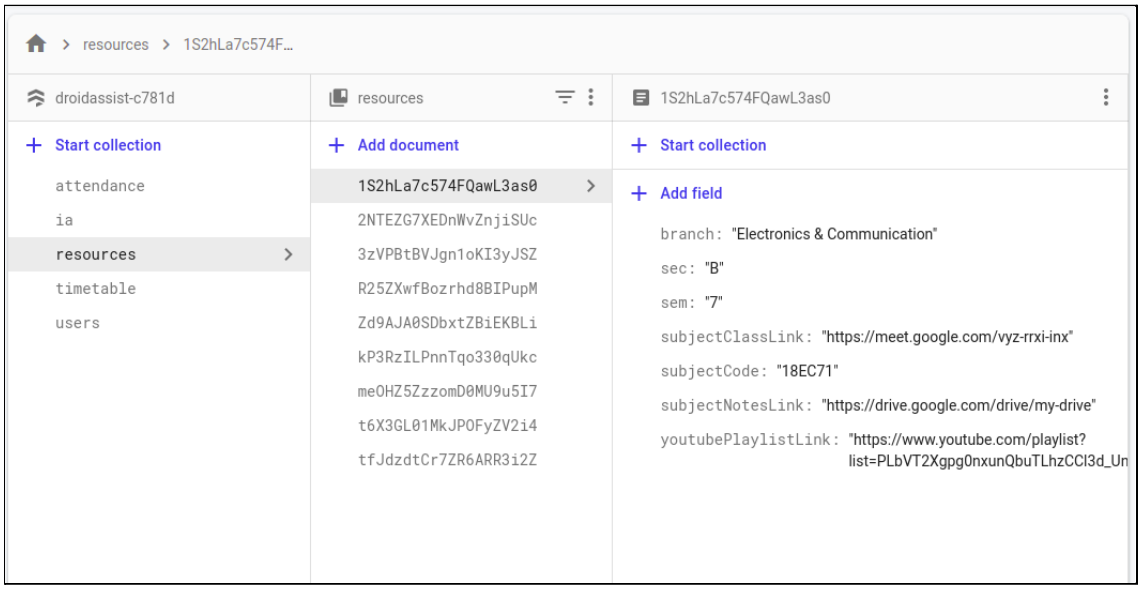

Fig. 2.5 Resources Collection

- branch: the branch of the user.
- sec: the section of the user.
- sem: the semester of the user.
- subjectClassLink: meeting link of the particular subject.
- subjectCode: code of the subject.
- subjectNotesLink: notes of a particular subject
- youtubePlaylistLink: link of the youtube playlist of the subject.

4. Internal Assessment Collection -

Internal assessment stores the scores of each of the assessments taken by the student. The collection contains documents of subjects and scores. For a student he would have a number of subjects multiplied by the Number of Assessments done. Each document has the following fields:

- branch: stores the branch of the user.
- score: the score achieved by the user in a particular subject.

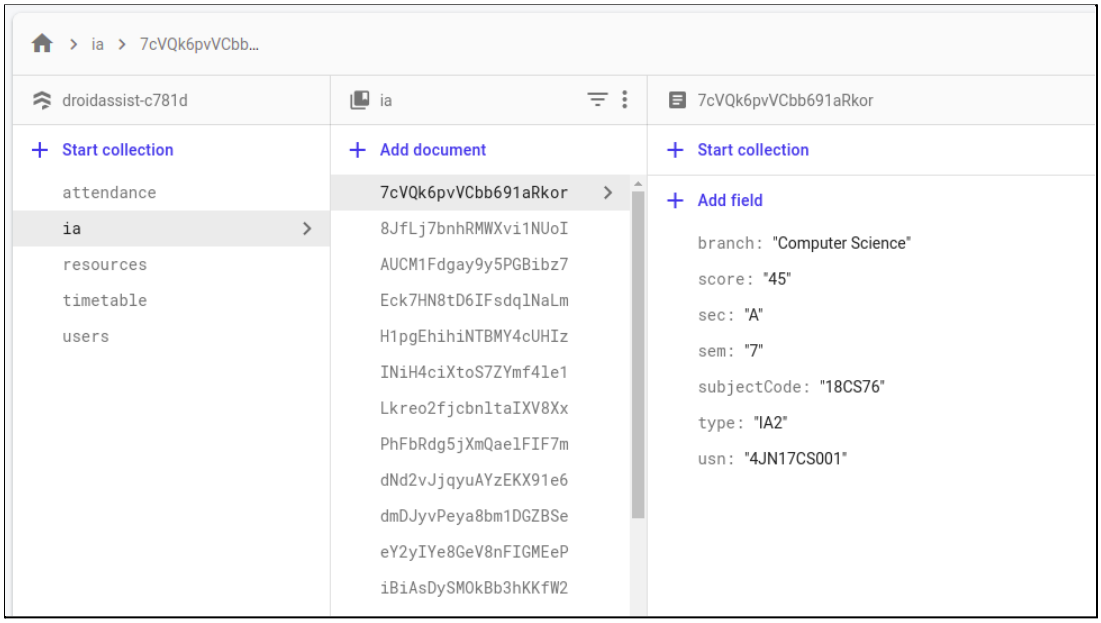

Fig. 2.6 Internal Assessment Collection

- sec: section the user belongs to.
- sem: semester the user belongs to.
- subjectCode: subject of the assessment taken.
- type: type of the assessment conducted.
- usn: USN of the user.

5. Attendance Collection -

The Attendance collection stores the attendance of each student. Each document is the attendance of a single subject of the particular student. Thus a student would have multiple documents of the subjects he has enrolled in a particular semester. Each document has the following fields:

- attendedClasses: number of classes a student has attended.
- branch: branch the student belongs to.
- sec: section the student belongs to.

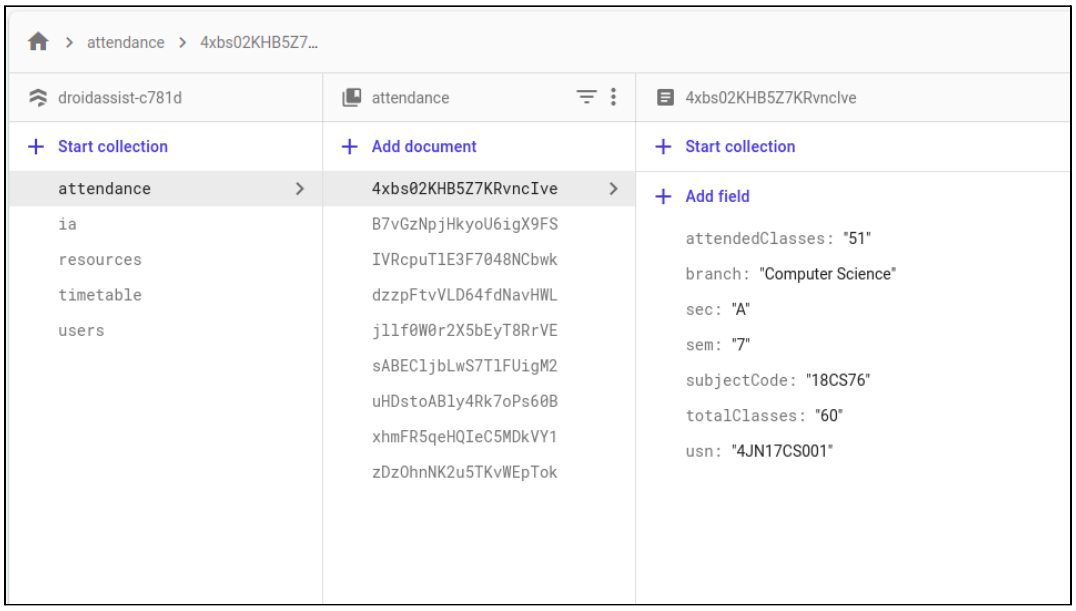

Fig. 2.7 Attendance Collection

- sem: semester the student belongs to.
- subjectCode: subject the attendance belongs to.
- totalClasses: total number of classes conducted.
- usn: USN of the student.

2.4 Implementation

Droid assist has mainly 8 Activities that handle the overall functionality of the app. The list of activities include:

1. Splash Activity.
2. Signup Activity.
3. Login Activity.
4. Home Activity.
5. Timetable Activity.
6. Resources Activity.
7. Attendance Activity.
8. Internal Assessment Activity.

All the activities are explained using a flowchart. Further explanation, if required will be present above or below the flowchart. All the code implemented in the project can be found on GitHub using the following link:
Droid Assist Codebase (https://github.com/nk4456542/DroidAssist).

1. **Splash Activity**

Splash Activity is the first screen that the user is shown when the app is started. The user can register if it is his first time or he can login if he already has credentials.

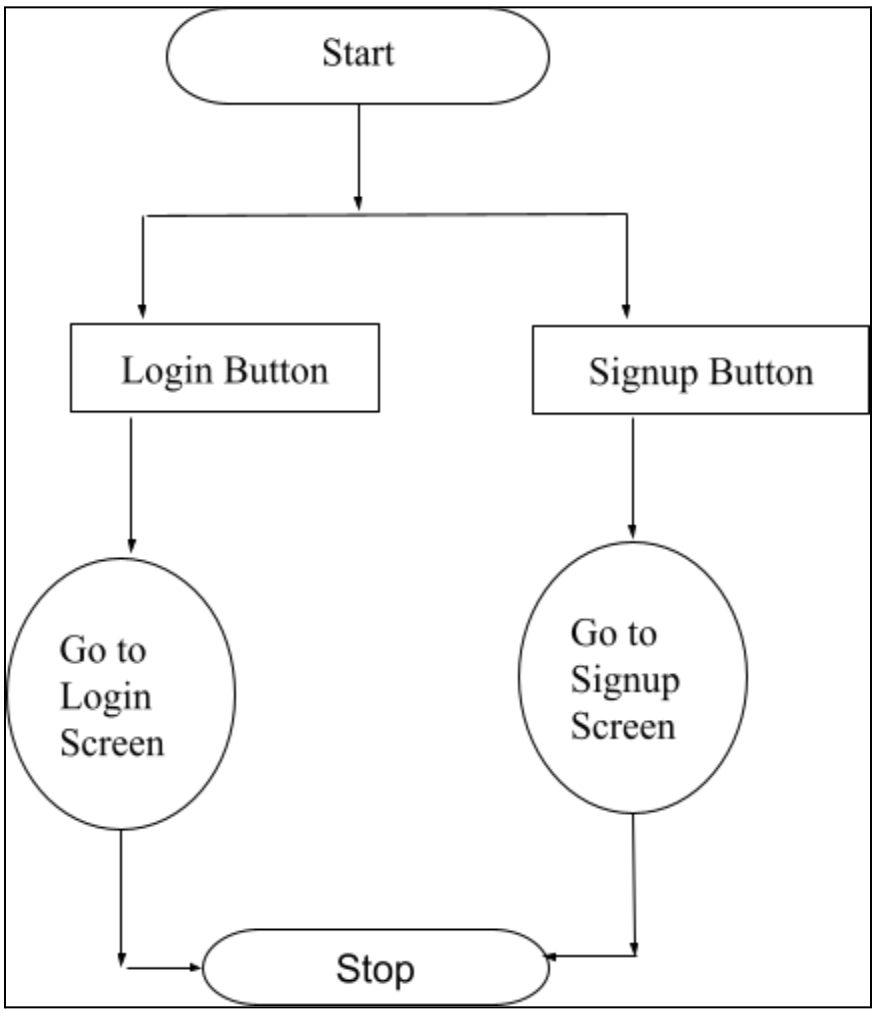

Fig. 2.8 Splash Activity Flowchart

2. Signup Activity

The user can register on Droid Assist. The following flowchart shows the implementation details of how the registration process works in Droid Assist.

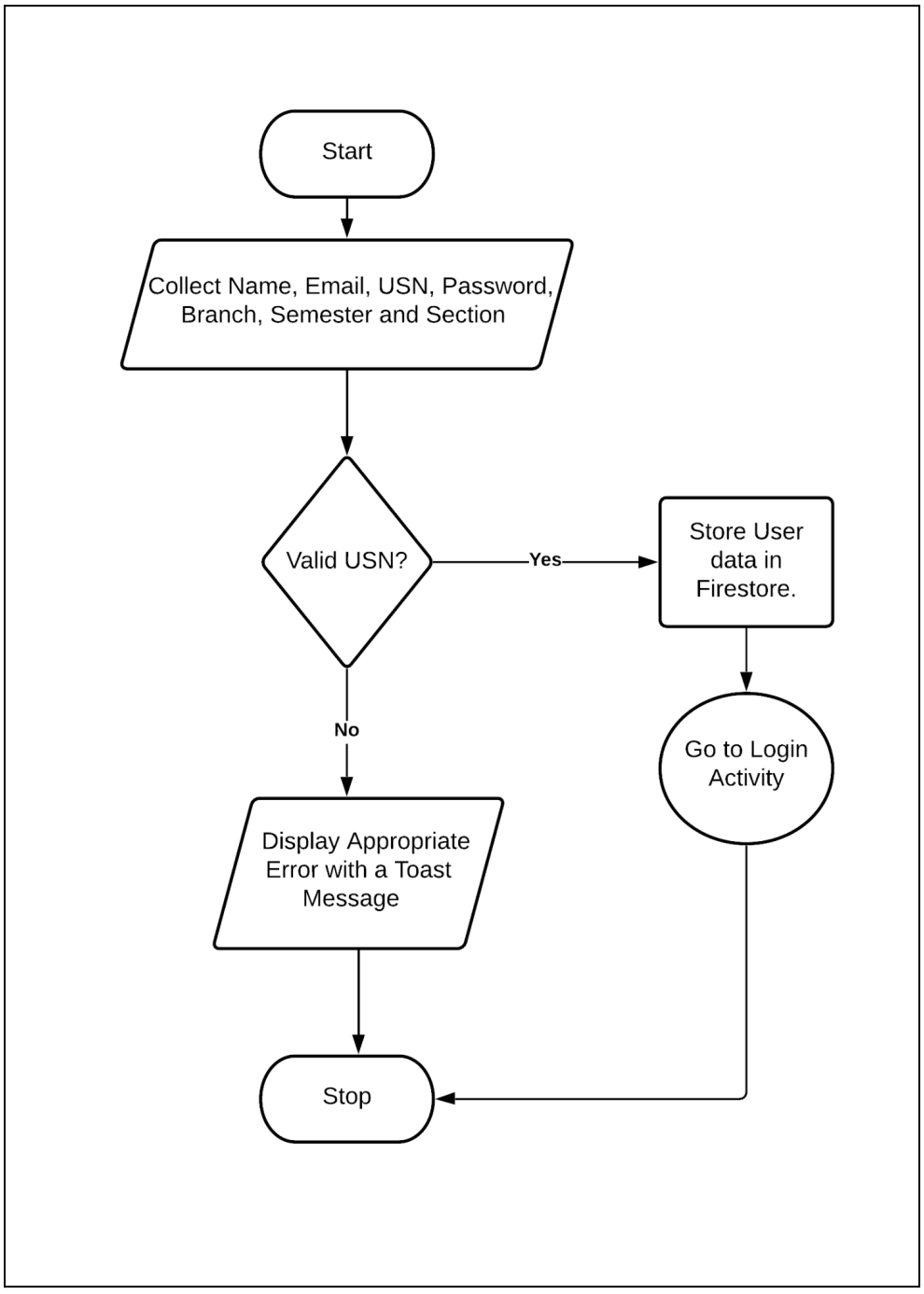

Fig. 2.9 Signup Activity Flowchart

3. Login Activity

Once the user has registered, the user can login using the same credentials he used during registration. Once the user logs in, he can avail all the functionalities of Droid Assist. The following flowchart shows the implementation details of Login Activity.

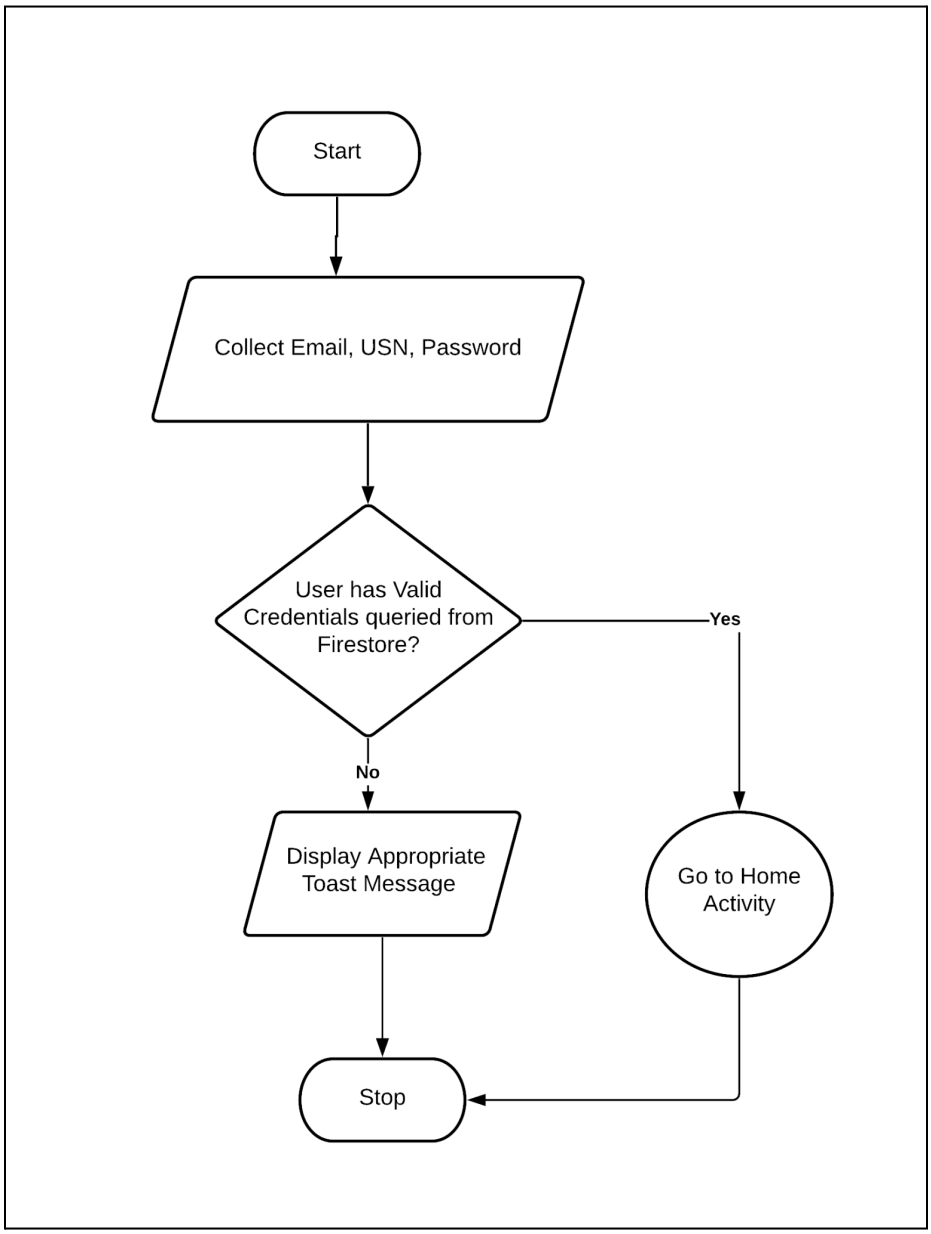

Fig. 2.10 Login Activity Flowchart

4. Home Activity

The user is provided a personal dashboard where he can see the today's classes and navigate between different activities. The flowchart for the same is shown below.

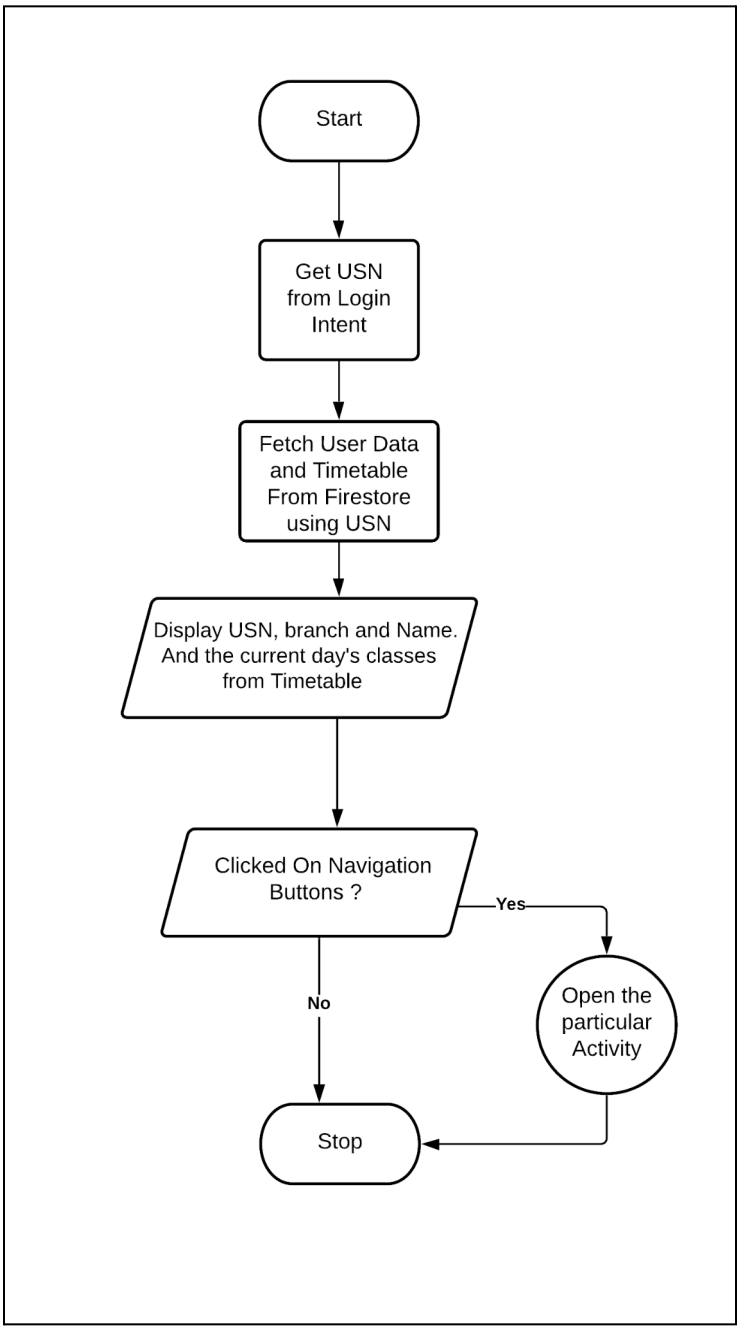

Fig. 2.11 Home Activity Flowchart

5. Timetable Activity

The timetable screen shows the timetable of a particular user. The details fetched from Firestore are displayed on screen using a RecyclerView. The following flowchart shows the implementation details of the Timetable Activity.

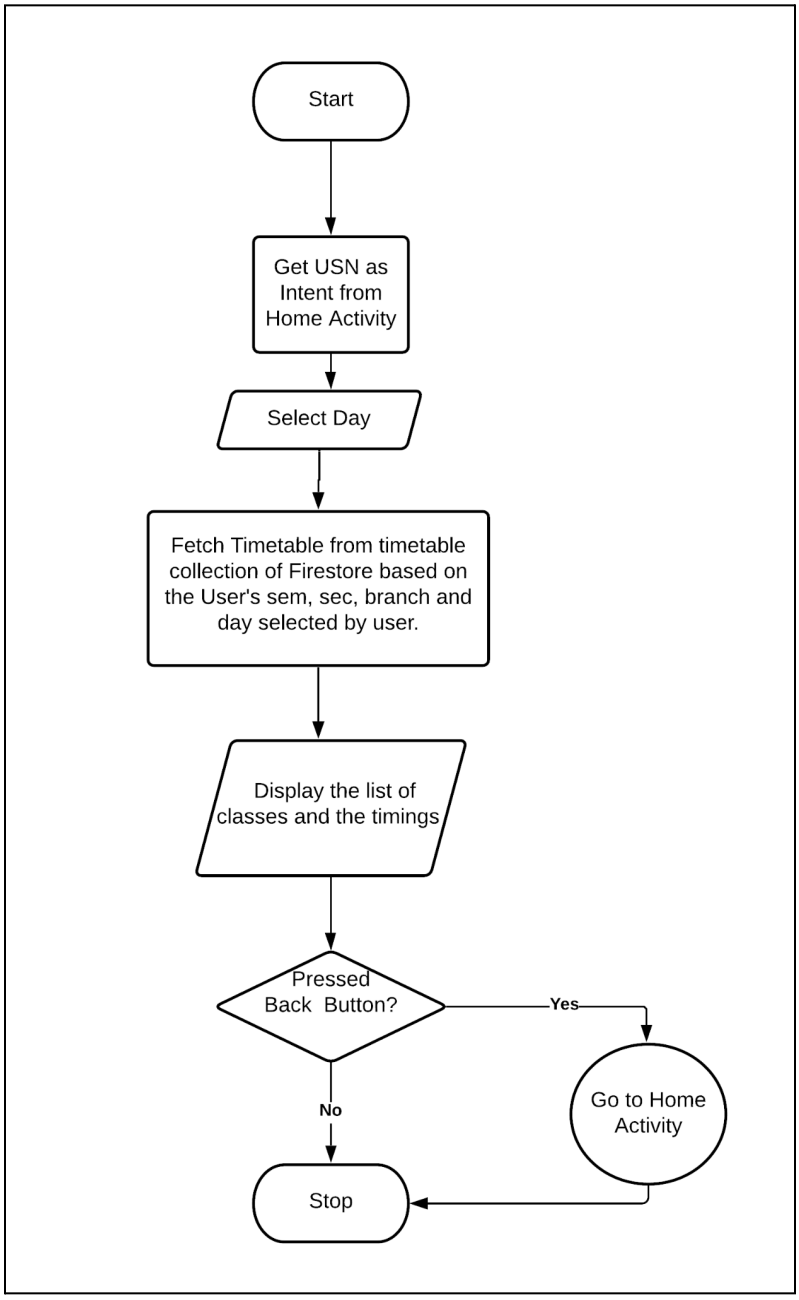

Fig. 2.12 Timetable Activity Flowchart

6. Resources Activity

Resources of each subject are fetched from Firestore and are shown in a RecyclerView. This helps students keep all the resources of each subject at a single place. The following flowchart shows the implementation details of the Resources Activity.

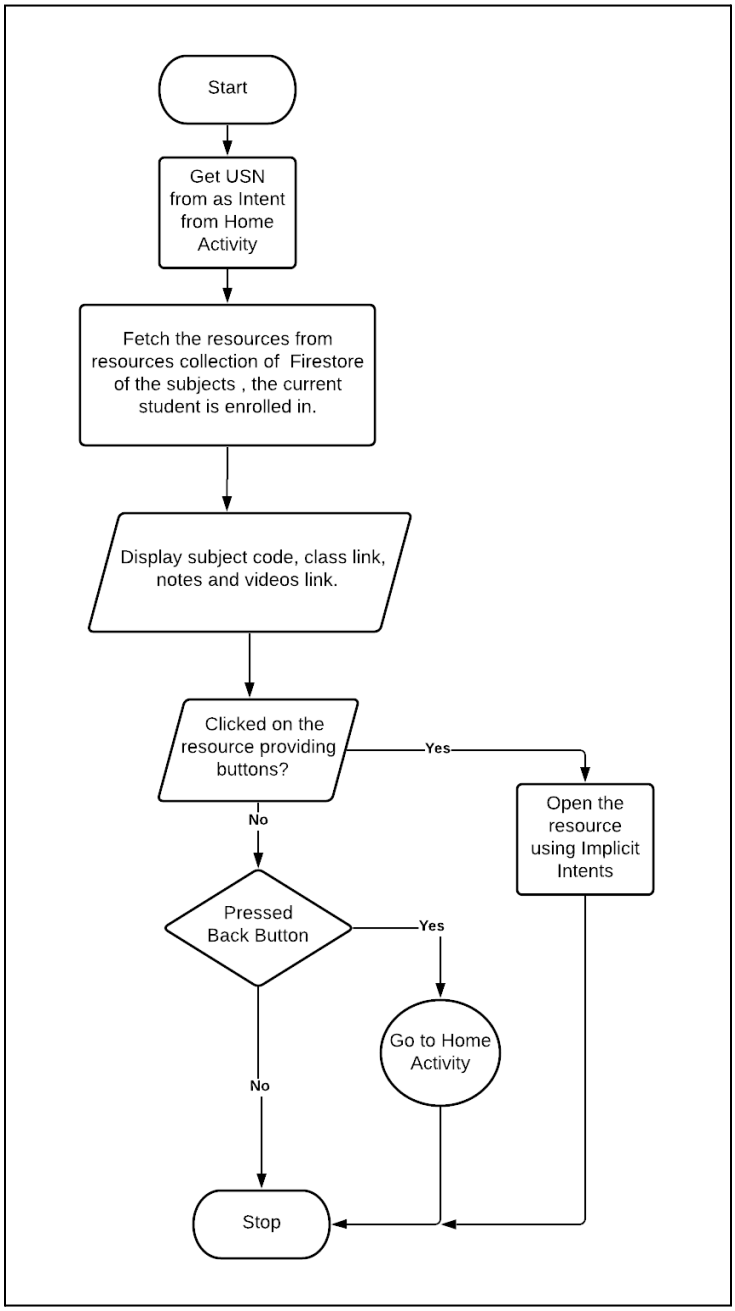

Fig. 2.13 Resources Activity Flowchart

7. Attendance Activity

Droid Assist helps students keep track of their attendance and fetches the attendance details from Firestore and displays it using a RecyclerView. The following flowchart shows the implementation of the Attendance Activity.

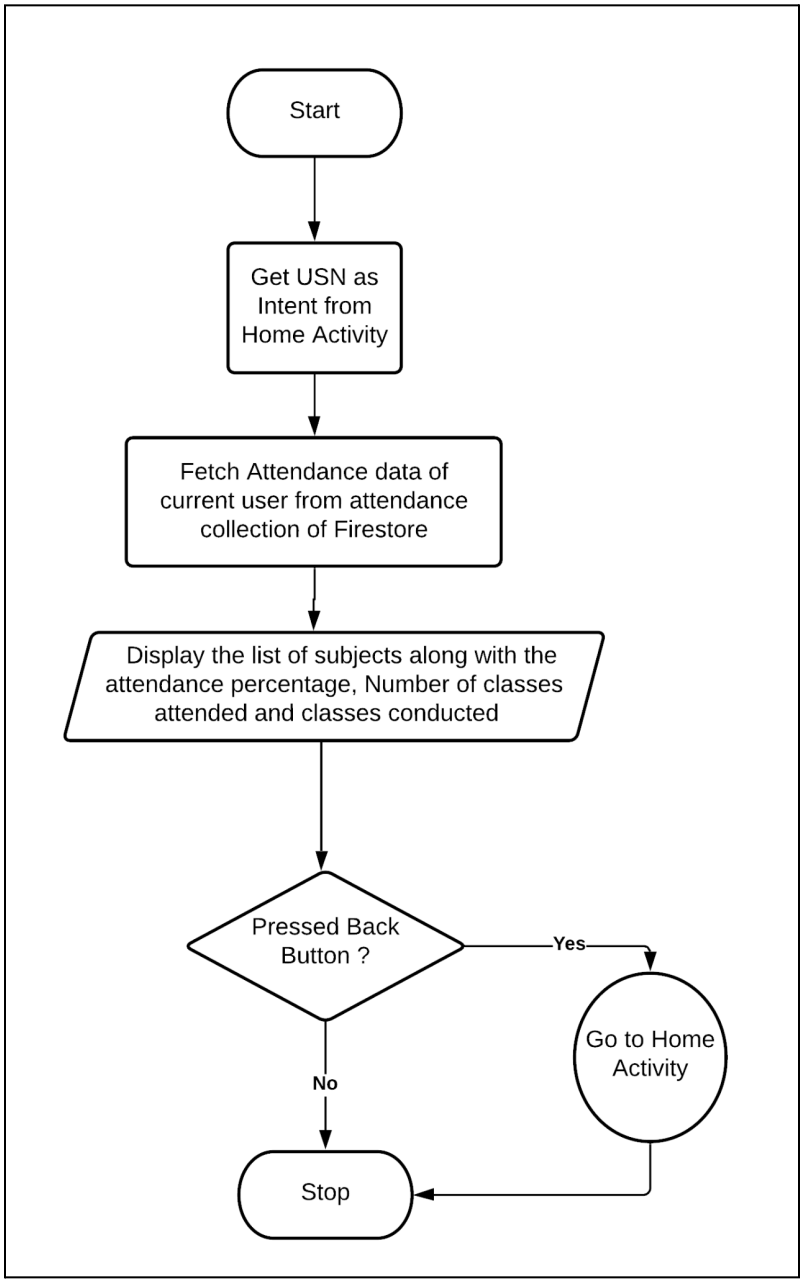

Fig. 2.14 Attendance Activity Flowchart

8. Internal Assessment Activity

The Internal Assessments scores are stored in a firebase collection for each student and the scores of each subject and each assessment are shown using a RecyclerView. The following flowchart explains the implementation of Internal Assessment Activity.

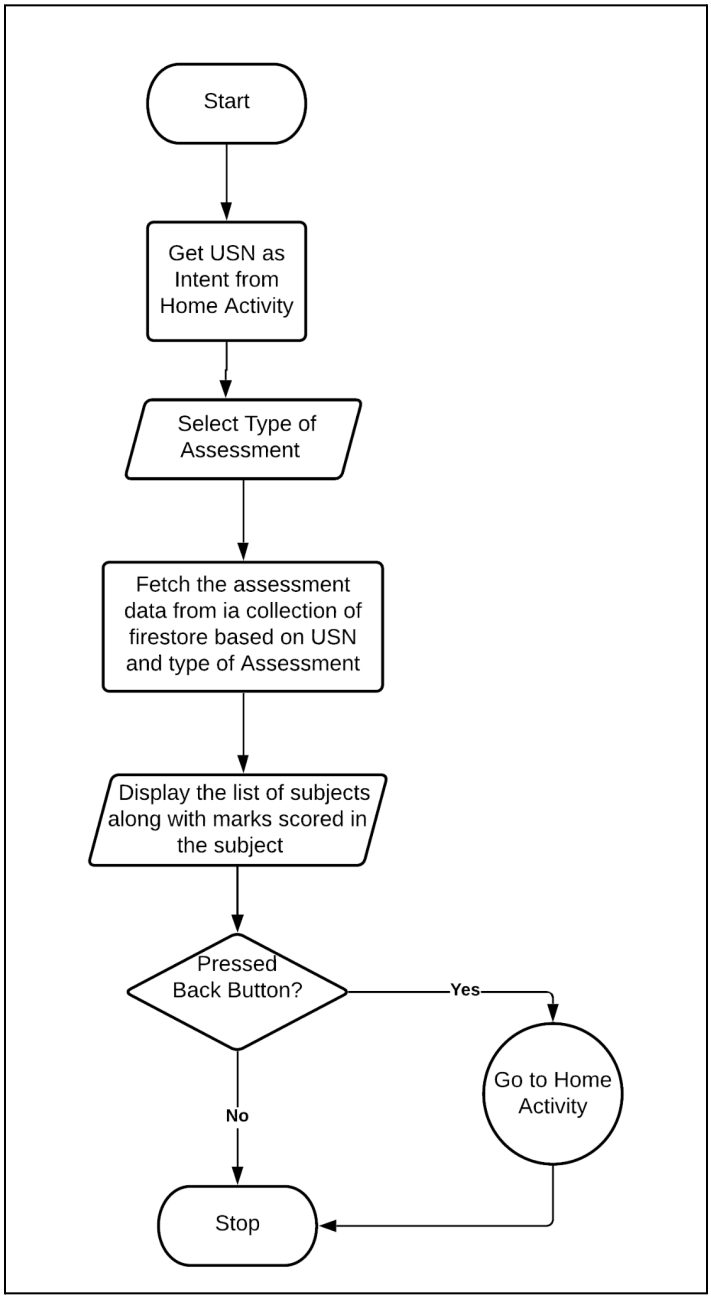

Fig. 2.15 Internal Assessment Activity Flowchart

2.5 APIs Used

Droid Assist uses the following APIs provided by Android and Firebase SDK to implement the functionalities.

APIs Used	**Explanation**
findViewById<View>(id)	Finds a view that was identified by the id attribute from the XML that was processed in OnCreate (Bundle).
startActivity(intent)	The method starts an instance of the Activity passed that's specified by the intent argument.
FirebaseAuth.getInstance( )	The entry point of the Firebase Authentication SDK. First, obtain an instance of this class by getInstance().
FirebaseFirestore.getInstance( )	Firebase Firestore provides the getInstance() method, which opens a socket (only one, at any time of execution of the app) and instantiates the Firebase Client.
auth.createUserWithEmailAndPassword (email, password)	Makes an asynchronous GET HTTP request to create the user in Firebase Authentication using email and password.
Toast.makeText(context, text, time).show( )	Display the message to the user as a Toast.
FirestoreDocument.get( )	Makes an asynchronous GET HTTP request to the Firestore. The result can be

	obtained by calling a lambda function addOnCompleteListener which will get the fetched results.
AdapterView.OnItemSelectedListener	Interface that has onItemSelected and onNothingSelected methods that are to be overridden. These functions are called whenever the spinner selects a new Input from the dropdown list.
Firestore.collection(particularCollection)	Creates a reference to a particular collection in the Firestore.
result.toObject(modelDataClass)	Query fetched from Firestore is converted to a model specified in the parameter. This is called during addOnSuccesListener or addOnCompleteListener
FirestoreCollectionRef.whereEqual(field, input).get()	Fetches the document with the matching fields passed a parameter.
RecyclerView	RecyclerView makes it easy to efficiently display large sets of data.RecyclerView library dynamically creates the elements when they're needed.
recyclerView.adapter	Recycler.Adapter is a class which provides the functionality of the recycler view. We have to override three main methods: onCreateViewHolder, onBindViewHolder, getItemCount. These three methods should have a class which

	extends RecyclerView.ViewHolder
RecyclerView.ViewHolder(itemView)	A class should extend to RecyclerView.Viewholder to get the necessary functionality of the RecyclerView. It helps in binding the data to the ViewHolder specified.
onCreateViewHolder(ViewGroup, viewType)	A function that is to be overridden when extending a RecycleView.Adapter. It creates an instance of the class as it extends the same.
onBindViewHolder(holder, position)	A function to bind the data between ViewGroup and the model data received from the Adapter parameters.
getItemCount ( )	Returns the size of the list. The list is received as a parameter from the Adapter.
intent.getStringExtra(key)	Get the specific val of the key from the intent passed from the other Activity.

Table 2.1 APIs Used

Chapter 3

RESULTS

3.1 Droid Assist Snapshots

Droid Assist mainly has 8 Activities, below are the snapshot of each Activity with the required explanation for the same.

3.1.1 Splash Activity

This is the first screen the user sees when the app is opened. He has basically two choices: Login and Signup. If the user is already registered then he can move forward with Login, else he should Sign Up. The below image shows the image of the splash Activity.

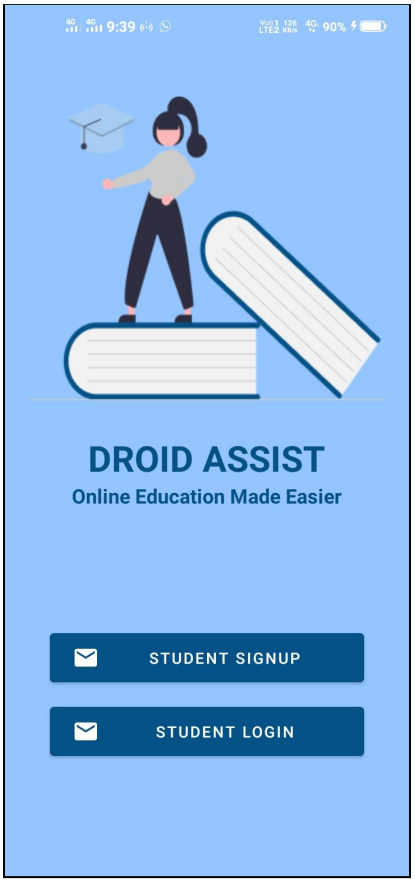

Fig. 3.1 Splash Activity Screen

3.1.2 Signup Activity

The User has to Register in Droid Assist to avail the functionalities of the App. The first image below shows the empty Signup Activity, and the second one shows the filled Signup Activity. All the validations are put into place wherever user input is required.

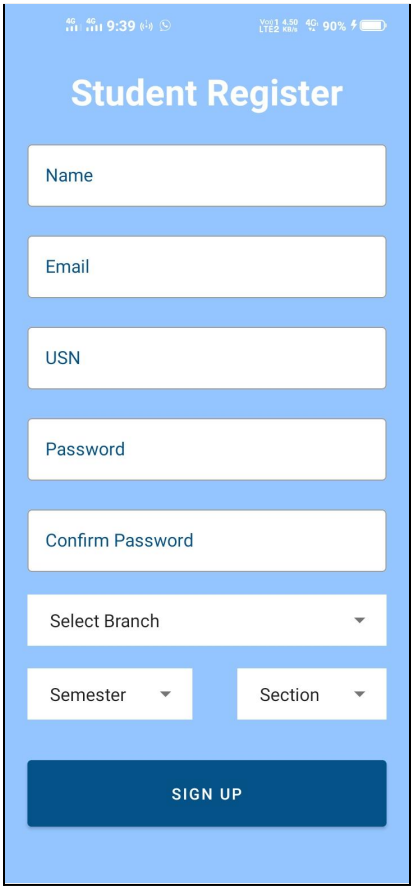

Fig. 3.2 Empty Signup Screen

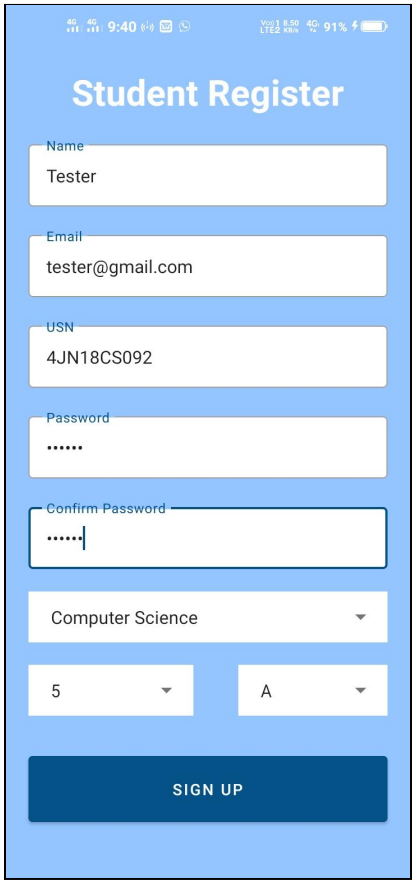

Fig. 3.3 Filled Signup Screen

3.1.3 Login Activity

Once the user has registered in Droid Assist, he can login to avail the functionalities of the App using the same credentials he used for signup. The below images show the result of Login Activity. First Image on the left shows the empty Login Form. Second Image shows the Filled Login Form.

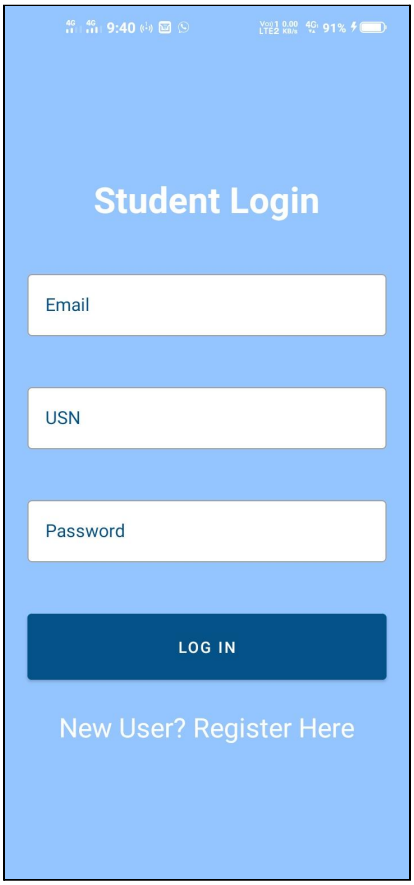

Fig. 3.4 Empty Login Screen

Fig. 3.5 Filled Login Screen

3.1.4 Home Activity

Once the user Logs in with valid credentials, he will be taken to the home screen, which has all the navigation buttons. The below image shows the Home Screen.

Fig. 3.6 Home Screen

3.1.5 Timetable Activity

User can access his timetable using the button present in Home Screen. Timetable Screen Images are shown below. The first image shows the timetable of Monday. Second image shows the timetable of Wednesday.

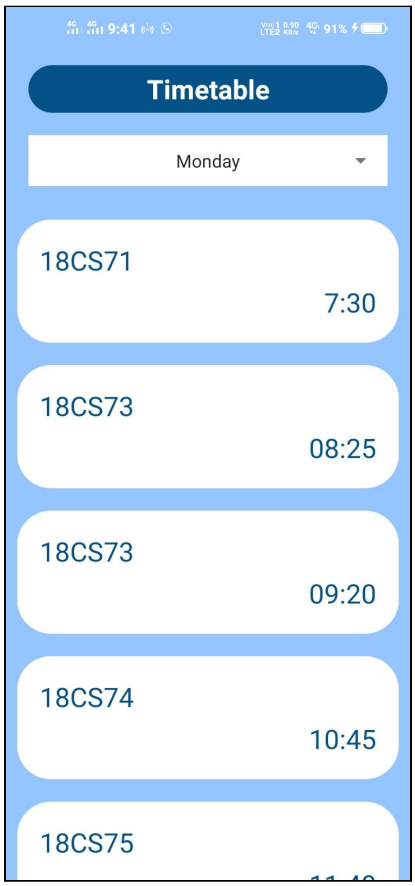

Fig. 3.7 Timetable Screen Monday

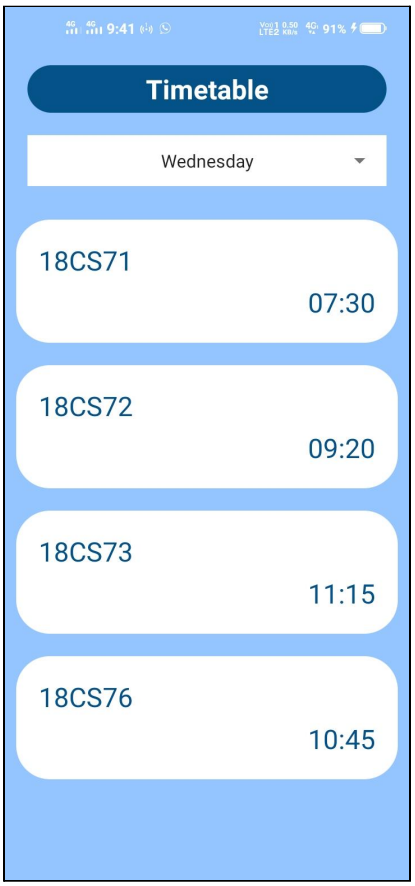

Fig. 3.8 Timetable Screen Wednesday

3.1.6 Attendance Activity

The user can access his attendance using the navigation button in the Home Screen. The below image shows the Attendance activity of the particular student. It shows the attendance percentage, classes conducted and the classes attended by the student.

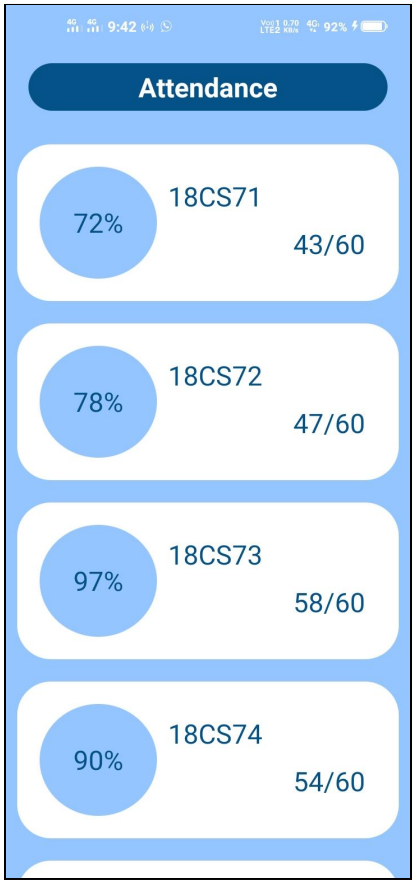

Fig. 3.9 Attendance Screen

3.1.7 Resources Activity

The user can access the Resources screen from the navigation buttons in the home screen. The below image shows the resources of a particular class. The first image shows the list of resources which includes class link, class notes link and class videos link. The second image shows when the user clicks on any of these resources, an implicit intent is called.

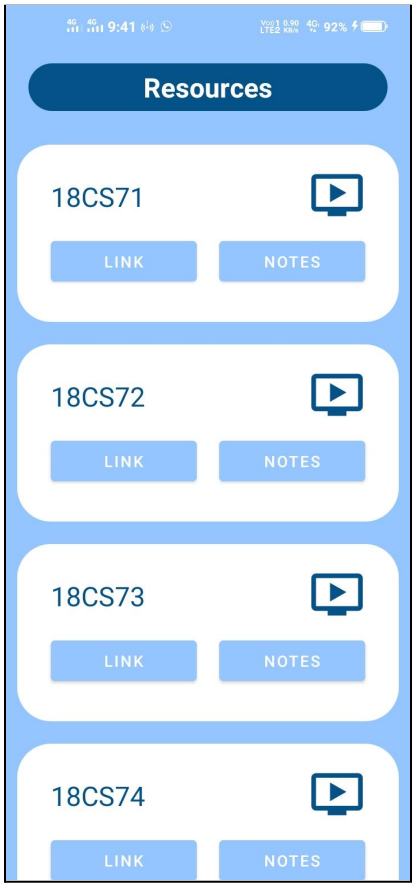

Fig. 3.10 Resources Screen

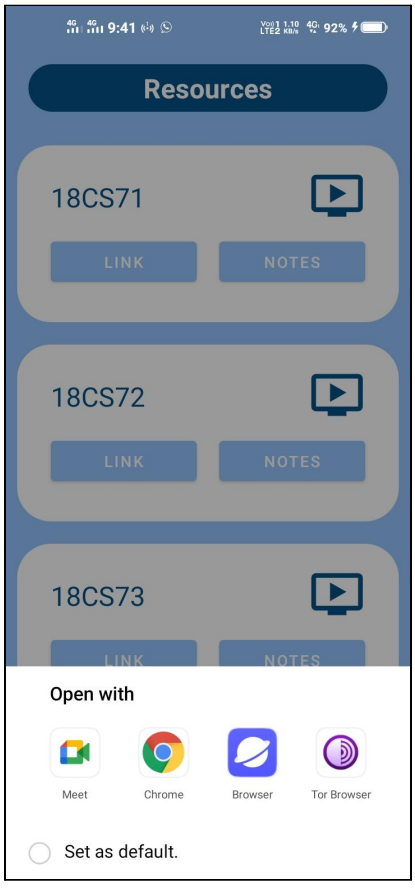

Fig. 3.11 Resource Implicit Intent

3.1.8 Internal Assessment Activity

The user can access the Internal Assessment screen from the Home Navigation buttons. The below image shows the internal assessment score of a particular student. The user can choose the type of assessment to view the scores of each assessment. The first Image shows the assessment scores of the First IA. Second Image shows the assessment scores of the third IA.

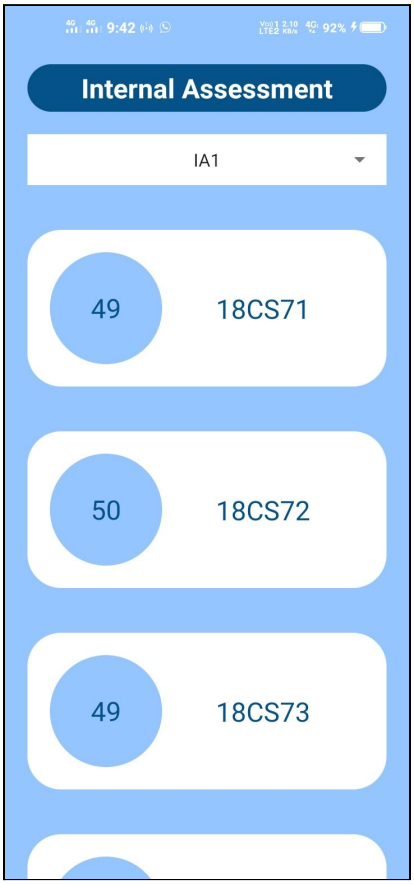

Fig. 3.12 IA Screen 1

Fig. 3.13 IA Screen 2

3.2 Firestore Snapshots

The below images show the collection and the documents present in the firstore of project Droid Assist. Each image has an explanation below it of all the fields present in the collection. All the firestore fields are explained in the Firestore Structure section of the report.

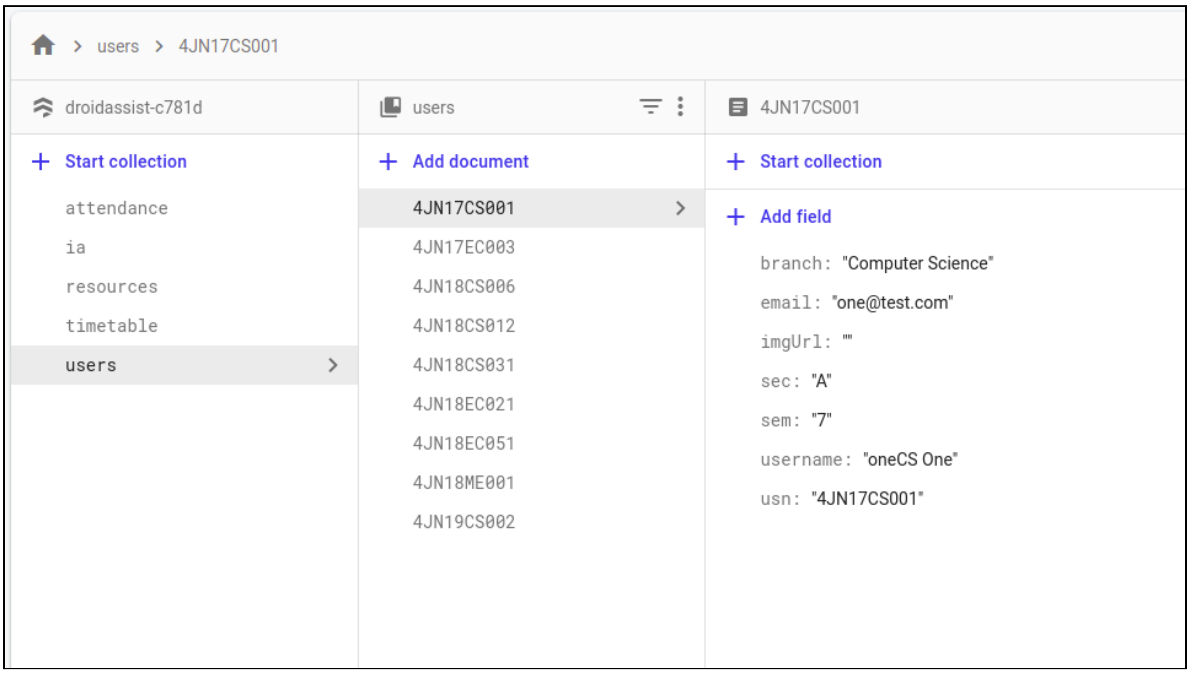

Fig. 3.14 Users Data Collection

The Above image shows the collection of “users”. User has the following fields: branch, email, imgUrl, sec, sem, username, usn. All of which are displayed in the above image.

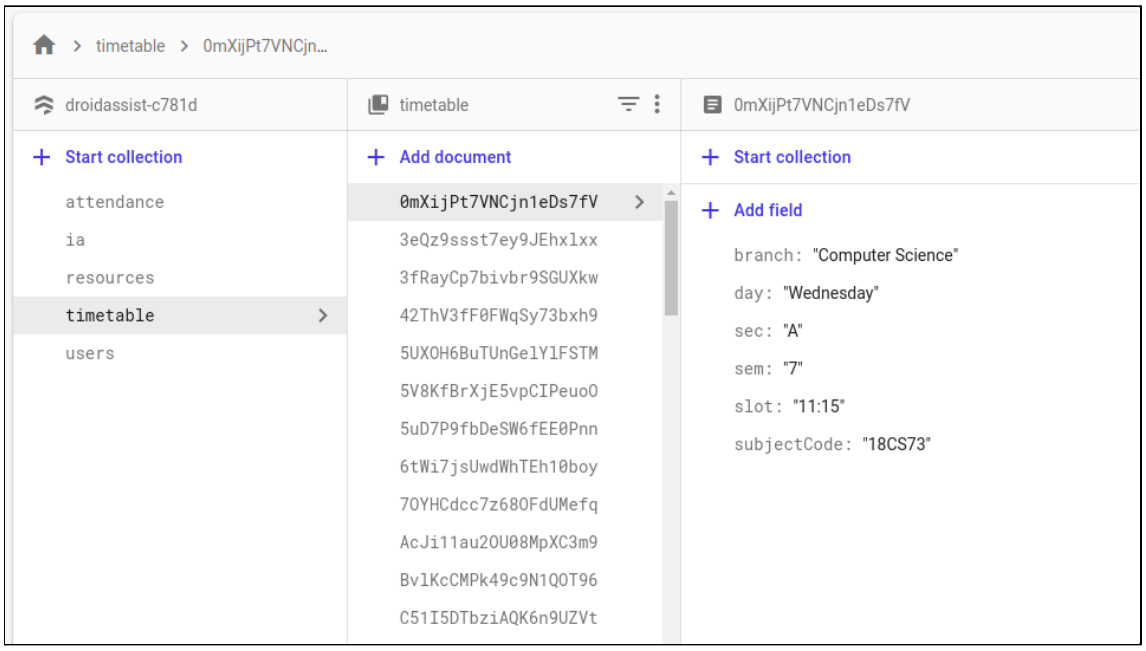

Fig. 3.15 Timetable Data Collection

The above image shows the collection of "timetable". Each document is a slot of each class. The document has the following fields: branch, day, sec, sem, slot and subjectCode.

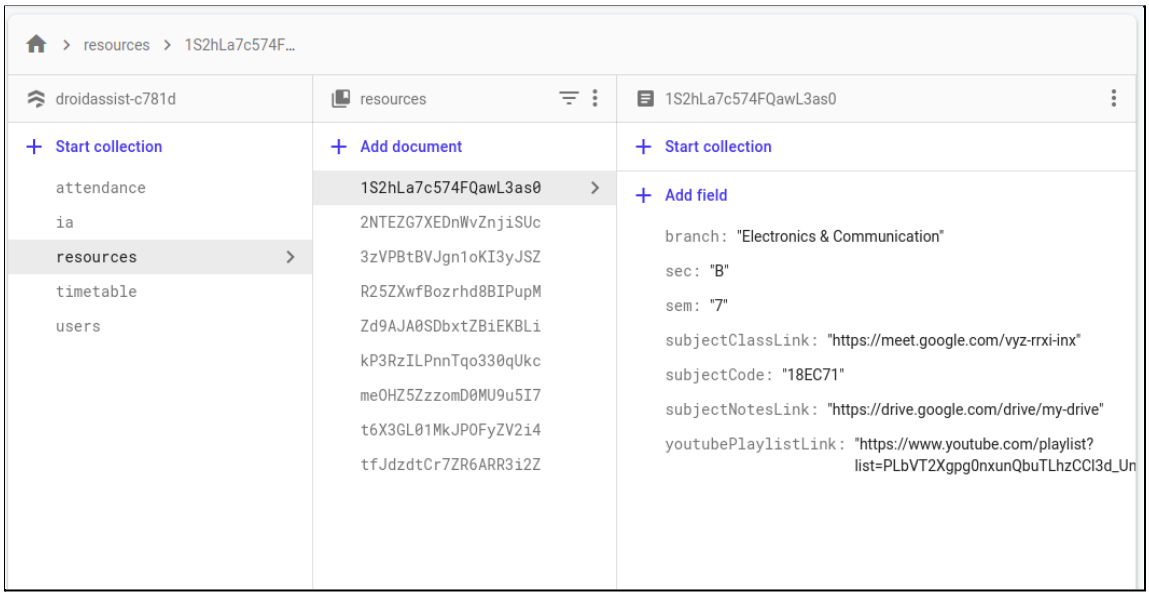

Fig. 3.16 Resources Data Collection

The above image shows the collection of "resources". Each document in resources represents the resources of a particular subject. It has the following fields: branch, sec, sem, subjectClassLink, subjectCode, youtubePlayListLink.

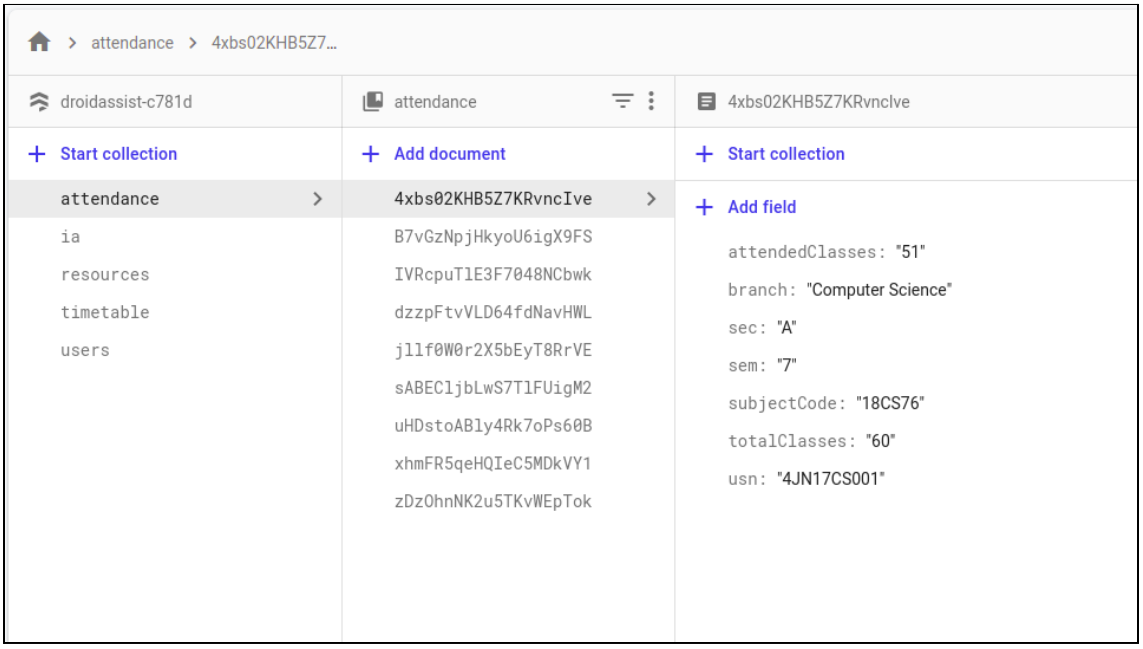

Fig. 3.17 Attendance Data Collection

The above image shows the collection of "attendance". Each document in a collection represents the attendance of a particular student in a particular subject. It has the following fields: attendedClasses, branch, sec, sem, subjectCode, totalClasses, usn.

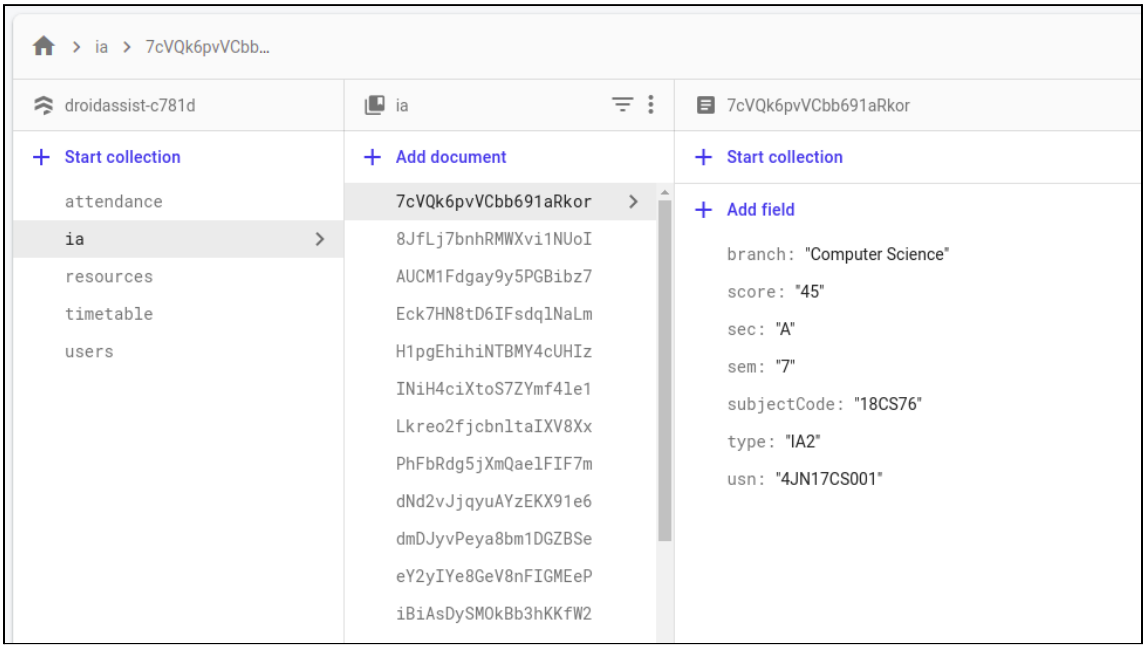

Fig. 3.18 Internal Assessment Data Collection

The above image shows the collection of “ia”. Each document is a score of a particular student in a particular subject. It has the following fields: branch, score, sec, sem, subjectCode, type and usn.

Chapter 4

CONCLUSION AND FUTURE SCOPE

Droid Assist was a project that was built to overcome the problems faced in online education. We believe we have addressed all the necessary problems, so that the student can focus on their work rather than worrying about the unnecessary details. Droid Assist works like a tool to enhance their productivity. We learned a lot while building this project, learned about Android development, Firebase and Firestore. All in all it was a good experience. There were some challenges along the way as well, in which our mentor guided us to the right path. And finally it was fun building this project with the team. This will solve the major problems faced during online education. We will be continuing to improve the project more. The projects which add value to the user at the end, are always fun to build and learn from. This is one such project.

4.1 Future Scope

There are some features that are missing currently. For example the user cannot upload his image as of now. This can be added as a new feature later with the help of firebase storage. Another app can be built for faculties where they can update attendance and IA marks of each student using the app itself, making it a little bit easier for their work. And help them avoid the mundane tasks and increase their productivity. Many more features can be added to help address the problems of students and faculties. All these are suggested features to improve the scope of this project. So that it can serve more people. This will also help faculties to focus on their work rather than the mundane task of maintaining attendance records in a manual manner.

REFERENCES

1. Textbook - A.V. Oppenheim and R.W. Schafer, Digital Signal Processing, Englewood, N.J., Prentice Hall, 3 Edition, 1975.
2. Paper - Devid, Insulation design to combat pollution problem, Proc of IEEE, PAS, Vol 71, Aug 1981, pp 1901-1907
3. Android Docs - https://developer.android.com/docs
4. Firebase Docs - https://firebase.google.com/docs/build
5. Firestore Docs - https://firebase.google.com/docs/firestore
6. App Icon Creator - https://romannurik.github.io/AndroidAssetStudio/icons-launcher