

Student Fees Tracker Cross Platform App

Project Synopsis

Industrial Training (ECS591)

Degree

BACHELOR OF TECHNOLOGY (CSE)

**Specialization in Application Development using
Cloud and Analytics Platforms**

PROJECT GUIDE:

Mr. Amit Singh

SUBMITTED BY:

Manraj Singh (TCA2057022)

June – July 2022



FACULTY OF ENGINEERING & COMPUTING SCIENCES

TEERTHANKER MAHAVEER UNIVERSITY, MORADABAD

Table of Contents

1	Project Title	3
2	Domain	3
3	Problem Statement	3
4	Project Description	4
4.1	Scope of the Work	4
4.2	Project Modules	4
5	Implementation Methodology	5
	6
6	Technologies to be used.	7
6.1	Software platform	7
6.2	Hardware platform	7
6.3	Tools Used	7
7	Advantages of this Project	7
8	Future Scope and further enhancement of the Project	8
9	Conclusion	8
10	References	8

1 Project Title

Student Fees Tracking Cross Platform Application.

2 Domain

Cross Platform Application Development using Flutter and Firebase (Google Cloud)

Cross-platform mobile development is the creation of software applications that are compatible with multiple mobile operating systems. Originally, the complexity of developing mobile apps was compounded by the difficulty of building out a backend that worked across multiple platforms. Although it was time-consuming and expensive, it was often easier to build native applications for each mobile operating system. The problem was that the code built for one operating system could not be repurposed for another OS.

Flutter is an open-source UI software development kit created by Google. It is used to develop cross-platform applications for Android, iOS, Linux, macOS, Windows, Google Fuchsia, and the web from a single codebase.

Firebase is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps, grow their user base, and earn profit. It is built on Google's infrastructure.

Firebase is categorized as a NoSQL database program, which stores data in JSON-like documents.

3 Problem Statement

Teachers and individual coaching centers struggled to manage fee records of their students and maintained a register for the same, manually entering and verifying each transaction. This led to miscommunication and potential loss in case of a default.

4 Project Description

In this project we have build a hybrid application that is capable of running on android, IOS, Windows, MAC, Linux and browser as well.

This project enables tuition teachers and coaching centers to manage students' data and their fee details on a simple app.

4.1 Scope of the Work

- The application will allow users to add student details, store them on the backend server and manage them using a simple interface.
- Displaying a list of students sorted in alphabetical order with an overview about them which will be fetched from the backend server.
- A students details screen which will display information about a student and their parents along with recent fee transactions they have made.
- Allowing users to mark fees of a particular month as paid and update that in the backend.
- Displaying a list of students who have paid their fees for a particular month fetched from the backend.
- A list of students who have not paid the fees for a specified month in order to remind them to do so at the earliest.

4.2 Project Modules

- A form to add student details.
- Calling backend API's via get and post methods.
- Storing student details on backend server and assigning a unique id to them.
- Fetching a list of students with fees paid for a particular month from the backend.
- Displaying recent fees transactions of a particular student.
- Updating student fees details on payment date.
- Opening native call dialer by clicking the call button.

5 Implementation Methodology

1. Adding Student Details

Student registration is the first part of using the application. A form with different validations is created for the same in order to maintain data consistency and integrity that is saved onto the backend servers. For example not allowing mobile number less than or greater than 10 digits, shifting focus on press of done button on the keyboard, changing keyboard type according to the field being entered. Also, a snackbar is displayed to tell the user what corrections needs to take place before successful registration of a student.

Finally, the data is sent and stored onto the server with a unique Id assigned to it in order to identify it further down the road.

2. Fetching data from the server using firebase queries

Firebase realtime database allows to access streaming data and as and when there is any change in the data present on the server. A stream builder is used to continuously keep in touch update the UI when there is any change in the data on the server. Firebase has its own set of query rules and types which are used to store and fetch data from its database.

Using these queries fee transactions are recorded stored onto the server and a list of students who have paid or not paid their fee for a particular month is fetched.

Firebase SDK ensures these transactions are molecular and uses encryption to safeguard the information travelling from the client to the server.

3. Theme Settings

Theme setting is another critical factor which improves customer interaction and brings in a feel-good factor. Flutter allows to set a default theme color for the complete UI all at once and manage it at a single place. The Primary Color swatch gives the app a pallet of all shades for a particular color which is applied to the key elements of the app automatically. Like appbar, buttons, loading indicator etc.

4. Building User Interface using widgets to display data fetched from the server

Widgets are developed using dart and material UI package provided by flutter which uses flutter's default screen painting architecture to display UI components to the user.

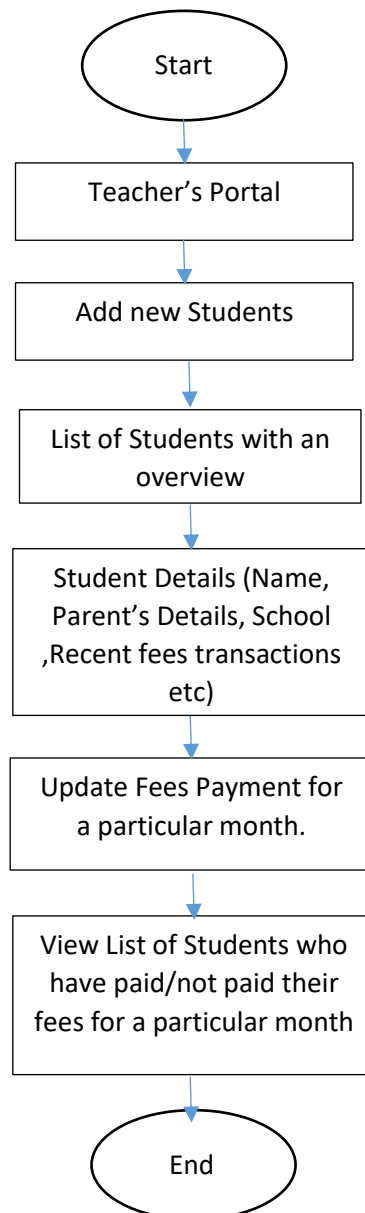
Widgets include future builder which is used to send request to the server based on user's action and update the User Interface based on the response from the server. List View is also one of the widgets which displays a scrollable list dynamically build according to the data.

Further a drop-down menu is used to trigger a server call to fetch the list of students with there fees in pending state for the current month.

5. Screen Routing and Passing data between different screens.

Multiple screens are created in order to display different information and they all are connected with the help of routes and Navigator which is built into flutter for the very same purpose.

Passing data from one screen to another is achieved using the provider package which abstracts a lot of backend complexity and stores data which is accessible throughout the application without managing each state independently.



WORKFLOW DIAGRAM

6 Technologies to be used.

6.1 Software platform

- a. Flutter – version 3.0.5 (Frontend)
- b. Google Firebase with Firebase Realtime Database (Backend)

6.2 Hardware platform

- a. RAM : Greater than 8GB
- b. Hard Disk:
- c. Processor : I3 and above

6.3 Tools Used

- a. Android Studio
- b. Android Emulator with android version – 12.3
- c. Visual Studio Code
- d. Flutter Firebase Plugin
- e. Dart
- f. URL Launcher Plugin

7 Advantages of this Project

1. **Reduces Potential Monitory loses:** with the help of this app a well structured and organized system is provided to search for students who have not paid their fees for any specific month, which was quite tedious with handwritten records maintained earlier. This will help teachers to identify those students and notify them about the same which they could have missed with their traditional methods just because of some human error.
2. **All records are safe and secure:** The data is stored on cloud and is constantly being backed up by the cloud service provider so there is no risk of data loss or breach in any unforeseen circumstance.
3. **A Simplified Paperless system:** This enables to develop a paperless system as all the fee receipts and payment records will be maintained in a digital format. This saves from the hassle of dealing with reams of papers, receipts as well as is environmentally friendlier.

8 Future Scope and further enhancement of the Project

1. To integrate another student facing application that will enable students and their parents to keep a track of the fees.
2. Students can directly pay through their portal.
3. Payment links will be automatically sent to parent's mobile number via text SMS and WhatsApp to directly pay their fees on due date.
4. Further teachers can notify students through push notifications about change in time, homework, marks in exams etc. and also displayed on their dashboard.

9 Conclusion

Implementation of this system is one of the most effective ways for schools, colleges and other educational institutes to streamline their fee management processes in a way that helps save on way power, makes fee collection proficient and decreases the staff workload so that they can use their time and resources to concentrate on the students

10 References

1. <https://www.pub.dev>
2. <https://www.google.com>
3. <https://www.stackoverflow.com>
4. <https://www.udemy.com>
5. <https://www.wikipedia.com>
6. <https://www.stackexchange.com>
7. <https://www.javatpoint.com>
8. <https://www.geeksforgeeks.org>