Student Fees Tracker Cross Platform App

Project Report

<Version 1.0>

Industrial Training (ECS599)

Degree

BACHELOR OF TECHNOLOGY (CSE IBM)
(Application Development Using Cloud and Analytics Platform)

PROJECT GUIDE:
Mr. Amit Singh

SUBMITTED BY:
Manraj Singh (TCA2057022)
Uday Raj (TCA2057032)

December, 2022

COLLEGE OF COMPUTING SCIENCES AND INFORMATION TECHNOLOGY

TEERTHANKER MAHAVEER UNIVERSITY, MORADABAD

ACKNOWLEDGEMENT

I would like to express my gratitude and appreciation to all those who gave me the possibility to complete this report. Special thanks are due to my supervisor **Mr. Amit Singh** whose help, stimulating suggestions and encouragement helped me in all time of fabrication process and in writing this report. I also sincerely thanks for the time spent proof reading and correcting my many mistakes.

I also take this opportunity to express my deep sense of gratitude to our honorable Principal **"Dr. Rakesh Kumar Dwivedi", TMU,** for providing excellent academic climate in the college that made this endeavor possible.

I would also like to acknowledge and Many thanks go to the whole lecturer and supervisors who have given their full effort in guiding the team in achieving the goal as well as their encouragement to maintain our progress in track. My profound thanks go to especially to my friends for spending their time in helping and giving support whenever I need it in fabricating my project.

Finally, I express my gratitude to all the Teaching and Non-Teaching staff of "**Computer Science Department**," TMU for their timely support and suggestions.

Manraj Singh

Uday Raj

Place:

Date:

DECLARATION

We hereby declare that this Project Report titled **Student Fees Tracking Cross Platform Application** submitted by us and approved by our project guide, the College of Computing Sciences, and Information Technology (CCSIT), Teerthanker Mahaveer University, Moradabad, is a bona fide work undertaken by us and it is not submitted to any other University or Institution for the award of any degree diploma / certificate or published any time before.

Student Name:	Manraj Singh	Signature
Project Guide: (External)	Name **Mr. Vinay Bansal**	Signature
Project Guide: (Internal)	**Mr. Amit Singh**	Signature

Brief About the Company

PAVAGE TECHNOLOGIES PVT LTD

Date: August 15, 2022

TO WHOM IT MAY CONCERN

This isto certify that *Mr* Manraj Singh, a student of B.TECH CSE, **Teerthanker Mahaveer University Moradabad** worked under my supervision during his industrial training program in Flutter Application Development at the CBRE, Noida, From July01, 2022 to August 10, 2022.

During the internship he demonstrated good skills with a self-motivated attitude to learn new things.

We wish him all the best for his future endeavors

Vinay Bansal
TECH LEAD
Em ID:030898N

For PAVAGE TECHNOLOGIES PVT LTD

Corp Off- H-160, H Block, Sector 63, Noida, Uttar Pradesh-201301
Tel.: 0120-41022112,114,117 Email: info@pavagetech.com

Pavage Industries was founded in April 2010.
The CTO of this company is Mr. Vinay Bansal. The company is well known for its services like mobile app development, web development, resource outsourcing, etc. Algomender provides two types of services to its user

1. Contract based services
2. Resource Outsourcing Contract Based Services includes web development, Mobile App development, digital marketing, etc.,

And Resource Outsourcing means it gives its resources to other company for their work for some period of time on the basis of pay as you service. Technologies in which Pavage Industries deals with are:

1. Flutter for mobile App development
2. Node.js for backend
3. Html and CSS for front-end in web
4. WordPress for web Etc.,

Table of Contents

Appendix

A: Data Flow Diagram (DFD)

B: Entity Relationship Diagram (ERD)

C: Use Case Diagram (UCD)

D: Data Dictionary (DD)

E: Screen Shots

1 Project Title

Student Fees Tracking Cross Platform Application.

2 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.

The old manual system was suffering from a series of drawbacks. Since whole of the system was to be maintained with hands the process of keeping, maintaining and retrieving the information was very tedious and lengthy. The records were never used to be in a systematic order. there used to be lots of difficulties in associating any particular transaction with a particular context. If any information was to be found it was required to go through the different registers, documents there would never exist anything like report generation. There would always be unnecessary consumption of time while entering records and retrieving records. One more problem was that it was very difficult to find errors while entering the records. Once the records were entered it was very difficult to update these records.

The reason behind it is that there is lot of information to be maintained and have to be kept in mind while running the business. For this reason, we have provided features Present system is partially automated (computerized), actually existing system is quite laborious as one has to enter same information at three different places.

Following points should be well considered:

- Documents and reports that must be provided by the new system: there can also be few reports, which can help management in decision-making and cost controlling, but since these reports do not get required attention, such kind of reports and information were also identified and given required attention.

- Details of the information needed for each document and report.

- The required frequency and distribution for each document.

- Probable sources of information for each document and report.

- With the implementation of computerized system, the task of keeping records in an organized manner will be solved. The greatest of all is the retrieval of information, which will be at the click of the mouse. So, the proposed system helps in saving the time in different operations and making information flow easy giving valuable reports.

3 Project Description

The purpose of Student Fees Tracking Cross Platform Application is to automate the existing manual system by the help of computerized equipment and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with.

Student Fees Payment System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus, it will help organization in better utilization of resources. The organization can maintain computerized records without redundant entries. That means that one need not be distracted by information that is not relevant, while being able to reach the information. The aim is to automate its existing manual system by the help of computerized equipment and full-fledged computer software, fulfilling their requirements, so that equipment and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. Basically, the project describes how to manage for good performance and better services for the clients.

3.1 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.

3.2 Project Modules

1. FLUTTER

Flutter is a cross-platform framework that targets developing high-performance mobile applications. Flutter was publicly released in 2016 by Google. Besides running on Android and iOS flutter applications also run on Fuschia. Flutter is chosen as Google's application-level framework for its next-generation operating system. Flutter is exceptional because it is dependent on the device's OEM widgets rather than consuming web views. Flutter uses a high-performance rendering engine to render each view component using its own. This provides a chance to build applications that are as high-performance as native applications can be. In view of architecture, the engine's C or C++ code involves compilation with Android's NDK and LLVM for iOS respectively, and during the compilation process, the Dart code is compiled into native code. Hot reload feature in Flutter is called as Stateful hot reload and it is a major factor for boosting the development cycle. Flutter supports it during development. Stateful hot reload is implemented by sending the updated source code into the running Dart Virtual Machine (Dart VM) without changing the inner structure of the application, therefore the transitions and actions of the application will be well-preserved after hot reloading.

2. DART

In Flutter, every application is written with the help of Dart. Google has developed and maintained a programming language called Dart. It is extensively used inside Google and it has been verified to have the proficiency to develop enormous web applications, such as AdWords. Originally Dart was developed to

replace and succeed JavaScript. Thus, it implements most of the important characteristics of JavaScript's next standard (ES7), such as the keywords "async" and "await". Nonetheless, to attract developers that are not acquainted with JavaScript, Dart has a Java-like syntax. Flutter application renews the view tree on every new frame even when few other systems use reactive views. This behavior leads to a drawback that many objects, which might survive for a singular frame, will be created. As Dart is a modern programming language, it is optimized to handle this scenario in memory level with the help of "Generational Garbage Collection"

3. Firebase

NoSQL Database

Store and sync data with our NoSQL cloud database. Data is synced across all clients in Realtime, and remains available when your app goes offline. The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client. When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.

Key Capabilities Realtime

Instead of typical HTTP requests, the Firebase Realtime Database uses data synchronization every time data changes, any connected device receives that update within milliseconds. Provide collaborative and immersive experiences without thinking about networking code.

Offline

- Firebase apps remain responsive even when offline because the Firebase Realtime Database SDK persists your data to disk. Once connectivity is reestablished, the client device receives any changes it missed, synchronizing it with the current server state.

Accessible from Client Devices

- The Firebase Realtime Database can be accessed directly

from a mobile device or web browser; there's no need for an application server. Security and data
validation is available through the Firebase Realtime Database Security Rules, expression-based rules that are executed when data is read or written.

Scale across multiple databases

With Firebase Realtime Database on the Blaze pricing plan, you can support your app's data needs at scale by splitting your data across multiple database instances in the same Firebase project. Streamline authentication with Firebase Authentication on your project and authenticate users across your database instances. Control access to the data in each database with custom Firebase Realtime Database Rules for each database instance.

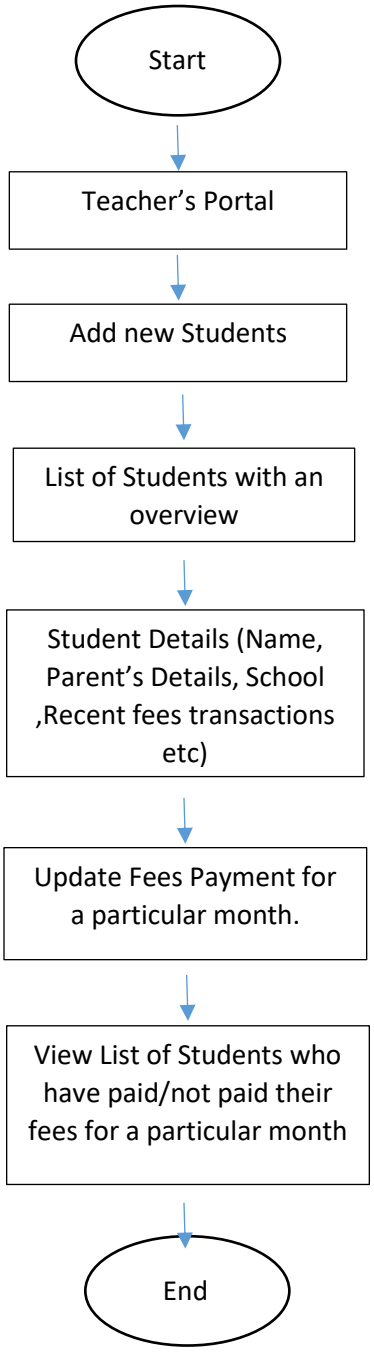
How does it work?

The Firebase Realtime Database lets you build rich, collaborative applications by allowing secure access to the database directly from client-side code. Data is persisted locally, and even while offline, realtime events continue to fire, giving the end user a responsive experience. When the device regains connection, the Realtime Database synchronizes the local data changes with the remote updates that occurred while the client was offline, merging any conflicts automatically.

The Realtime Database provides a flexible, expression-based rules language, called Firebase Realtime Database Security Rules, to define how your data should be structured and when data can be read from or written to. When integrated with Firebase Authentication, developers can define who has access to what data, and how they can access it. The Realtime Database is a NoSQL database and as such has different optimizations and functionality compared to a relational database.

The Realtime Database API is designed to only allow operations that can be executed quickly. This enables you to build a great realtime experience that can serve millions of users without compromising on responsiveness. Because of this, it is important to think about how users need to access your data and then structure it accordingly.

3.3 Context Diagram (High Level)



4 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.

   All these functions have been implemented to make this a robust and user friendly application:

   - 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.

   Testing of the Project

   Testing is vital for the success of any software. no system design is ever perfect. Testing is also carried in two phases. first phase is during the software engineering that is during the module creation.
   Second phase is after the completion of software. this is system testing which verifies that the whole set of programs hanged together.

   White Box Testing:

   In this technique, the close examination of the logical parts through the software are tested by cases that exercise species sets of conditions or loops. all logical parts of the software checked once.
   Errors that can be corrected using this technique are typographical errors, logical expressions which should be executed once may be getting executed more than once and error resulting by using wrong controls and loops.
   When the box testing tests all the independent part within a module a logical decisions on their true and the false side are exercised , all loops and bounds within their operational bounds were exercised and internal data structure to ensure their validity were exercised once.

Alpha Testing:

Acceptance testing is also sometimes called alpha testing. Be spoke systems are developed for a single customer. The alpha testing proceeds until the system developer and the customer agree that the provided system is an acceptable implementation of the system requirements.

Beta Testing:

On the other hand, when a system is to be marked as a software product, another process called beta testing is often conducted. During beta. Testing, a system is delivered among a number of potential users who agree with it, then report problems to the

Entity Relationship Diagram

E-R Model is a popular high level conceptual data model. This model and its variations are frequently used for the conceptual design of database application and many database design tools employ its concept.

A database that confirms to an E-R diagram can be represented by a collection of tables in the relational system. The mapping of E-R diagram to the entities are:

o Many-to-many
o Many-to-one
o One-to-many
o One-to-one

Weak entities
Sub-type and super-type

The entities and their relationships between them are shown using the following conventions.

An entity is shown in rectangle.

A diamond represents the relationship among number of entities.

The attributes shown as ovals are connected to the entities or relationship by lines.

Diamond, oval and relationships are labeled.

5 Technologies to be used

5.1 Software Platform

a) **Front-end**
Flutter – version 3.0.5 (Frontend)

b) **Back-end**
Google Firebase with Firebase Realtime Database (Backend)

5.2 Hardware Platform

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

5.3 Tools

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

6 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.

7 Assumptions, if any

NONE

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 Project Repository Location

S#	Project Artifacts (softcopy)	Location (Mention Lab-ID, Server ID, Folder Name etc.)	Verified by Project Guide	Verified by Lab In-Charge
1.	Project Synopsis Report (Final Version)	https://github.com/manraj2712/student_fees_tracking_app	Name and Signature	Name and Signature
2.	Project Progress updates		Name and Signature	Name and Signature
3.	Project Requirement specifications		Name and Signature	Name and Signature
4.	Project Report (Final Version)		Name and Signature	Name and Signature
5.	Test Repository		Name and Signature	Name and Signature
6.	Any other document, give details	https://github.com/manraj2712/student_fees_tracking_app	Name and Signature	Name and Signature

10 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

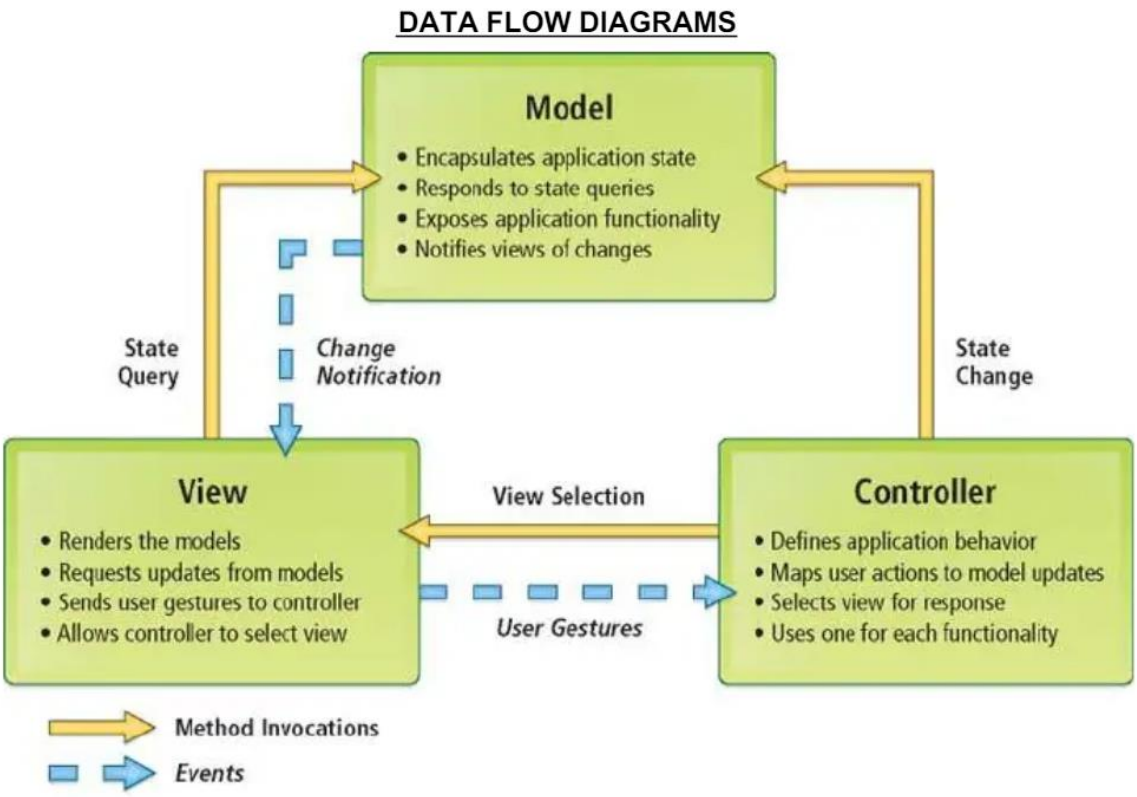
11 References

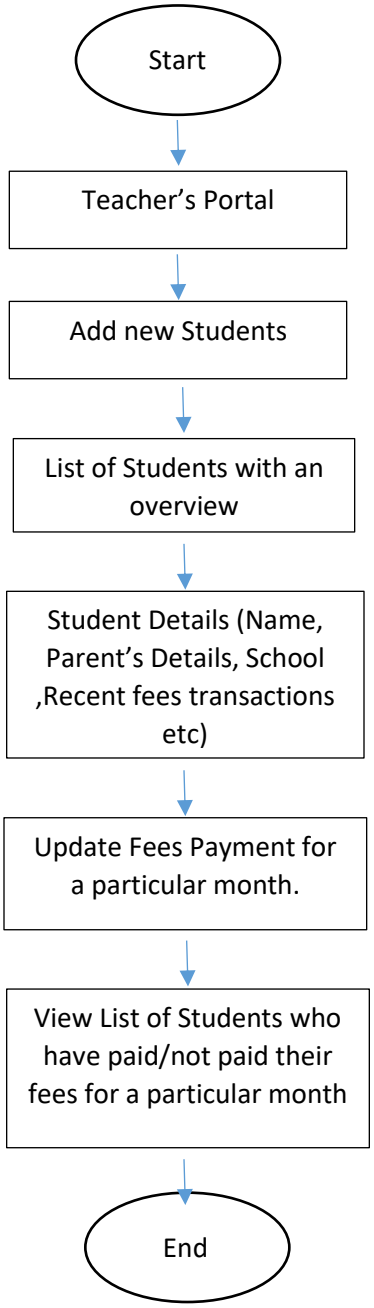
1. https://www.pub.dev
2. https://www.google.com
3. https://www.stackoverflow.com
4. https://www.udemy.com
5. https://www.wikipedia.com
6. https://www.stackexchange.com
7. https://www.javatpoint.com
8. https://www.geeksforgeeks.org

S#	Reference Details	Owner	Version
1.	Project Synopsis	Manraj Singh	1.0
2.	Project Requirements	Manraj Singh	1.0
3.			

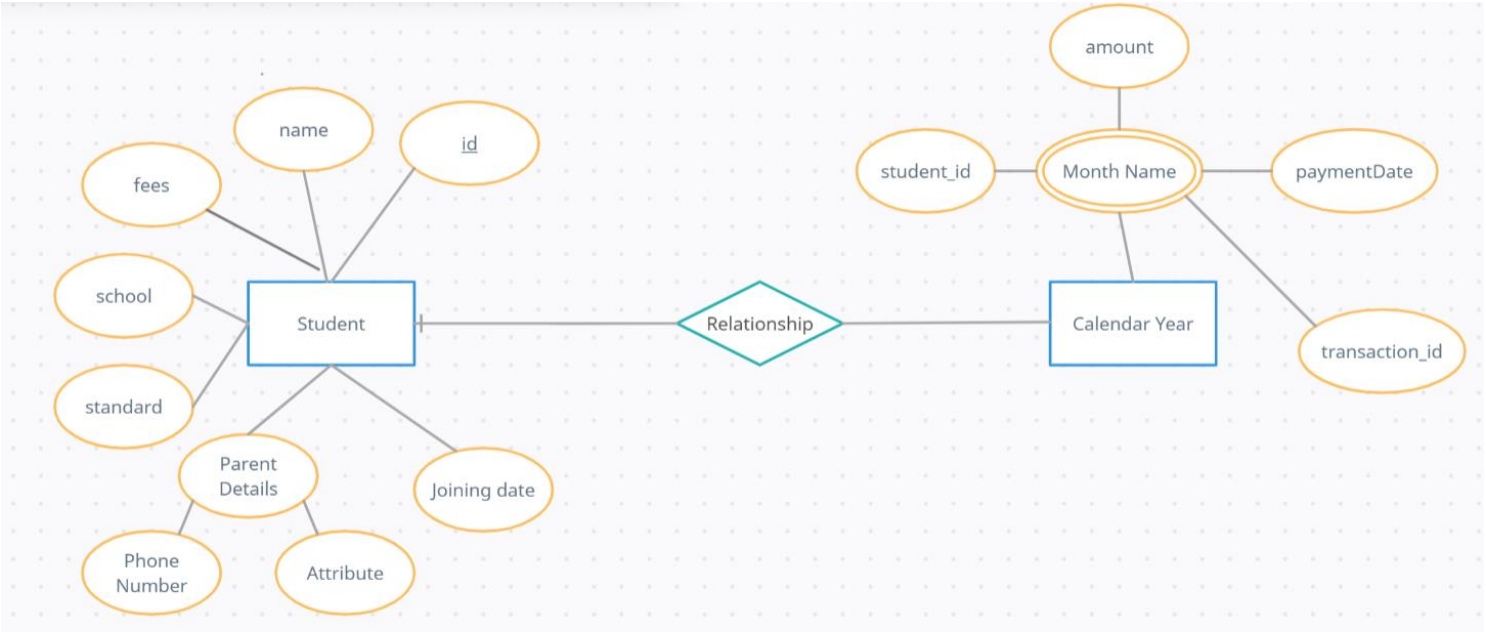
Annexure A
Data Flow Diagram (DFD)
(Mandatory)

Data Flow Diagram (level 0)

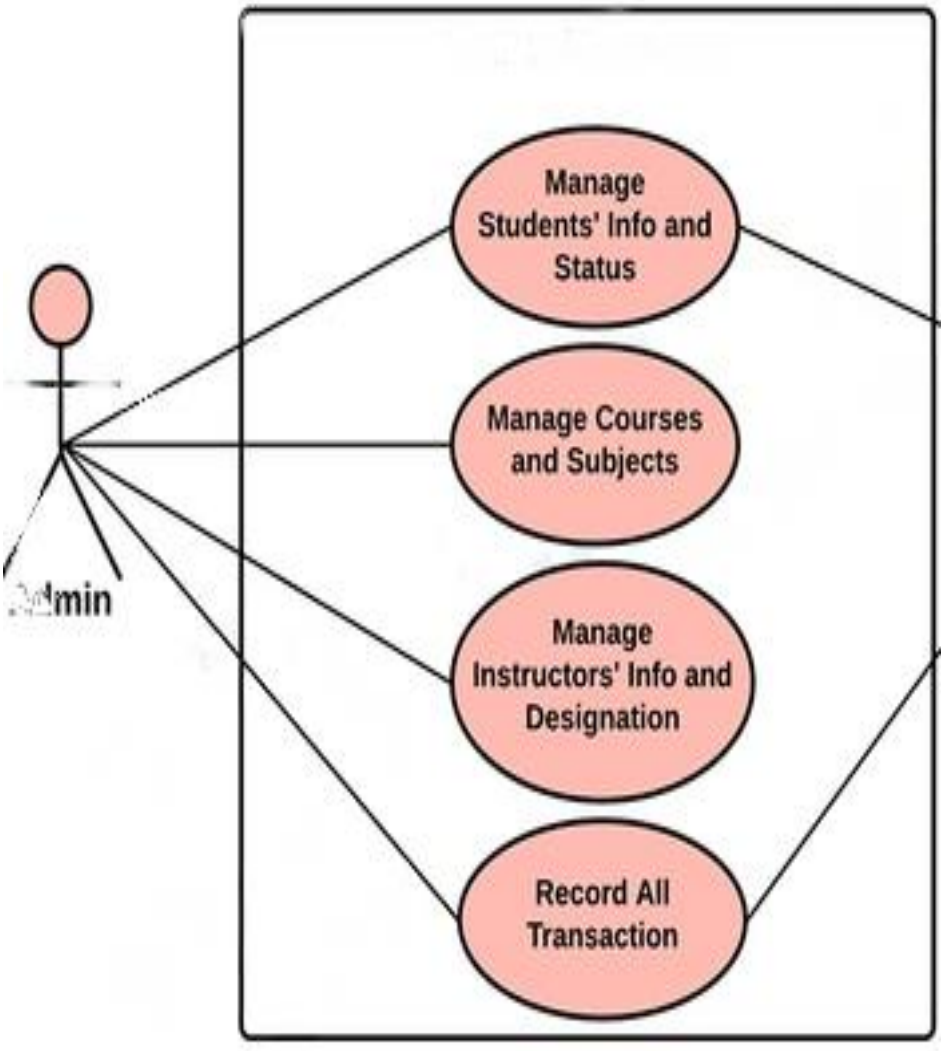



Start
Teacher's Portal
Add new Students
List of Students with an overview
Student Details (Name, Parent's Details, School ,Recent fees transactions etc)
Update Fees Payment for a particular month.
View List of Students who have paid/not paid their fees for a particular month
End

Annexure B
Entity-Relationship Diagram (ERD)
(Mandatory)



Annexure C
Use-Case Diagram (UCD)



Annexure D
Data Dictionary (DD)
(Mandatory)

Students Table (student)

Fields	Data type	Description
id	String	Unique student identity
name	Text	User Name
fees	Number	Amount student pays
joiningDate	ISO Date String	Date student started studying
parentsName	Text	Student Parent's name
parentsMobNumber	Number	Student Parent Mobile Number
studentMobNumber	Number	Student's mob number (if any)
school	Text	School Name
standard	Number	Class in which student currently is

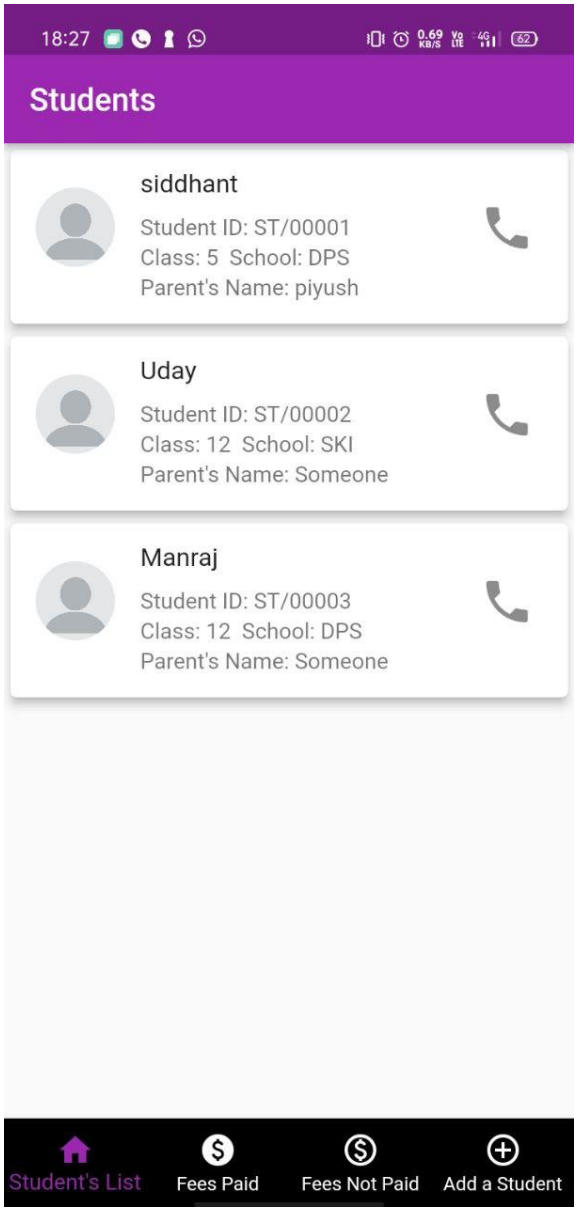
Year Table (year ex:2022)

Fields	Data type	Description
id	BSON String	Unique Transaction Id
studentId	String	Acts as a foreign key to students table
amount	Number	Amount Paid in this transaction
paymentDate	ISO Date String	Date at which transaction took place

Annexure E
Screen Shots

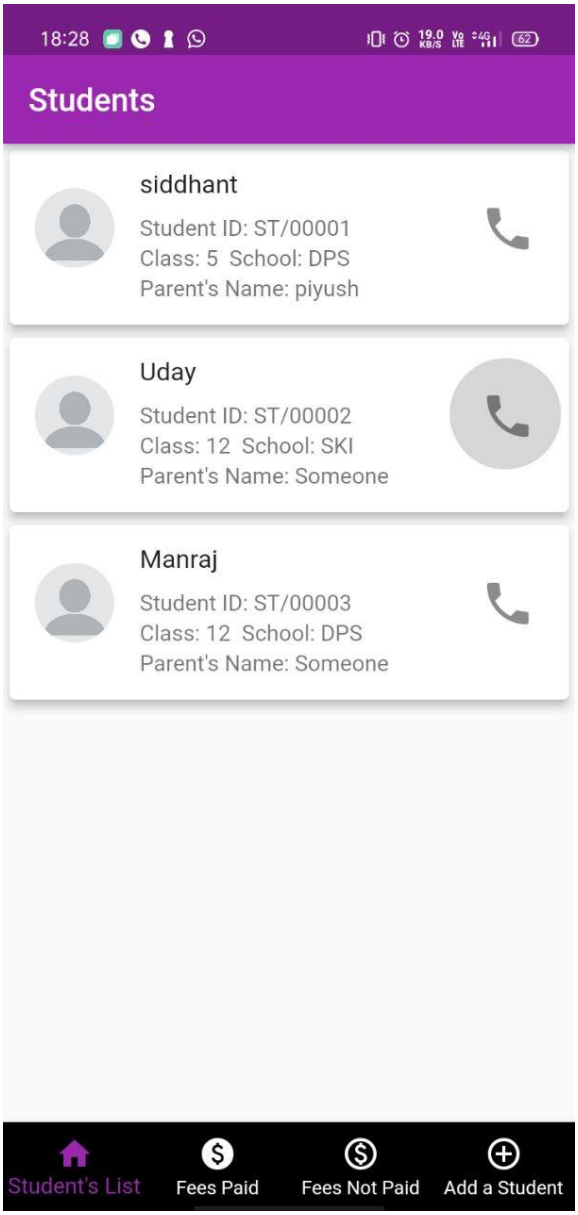
Screen Shot: 1

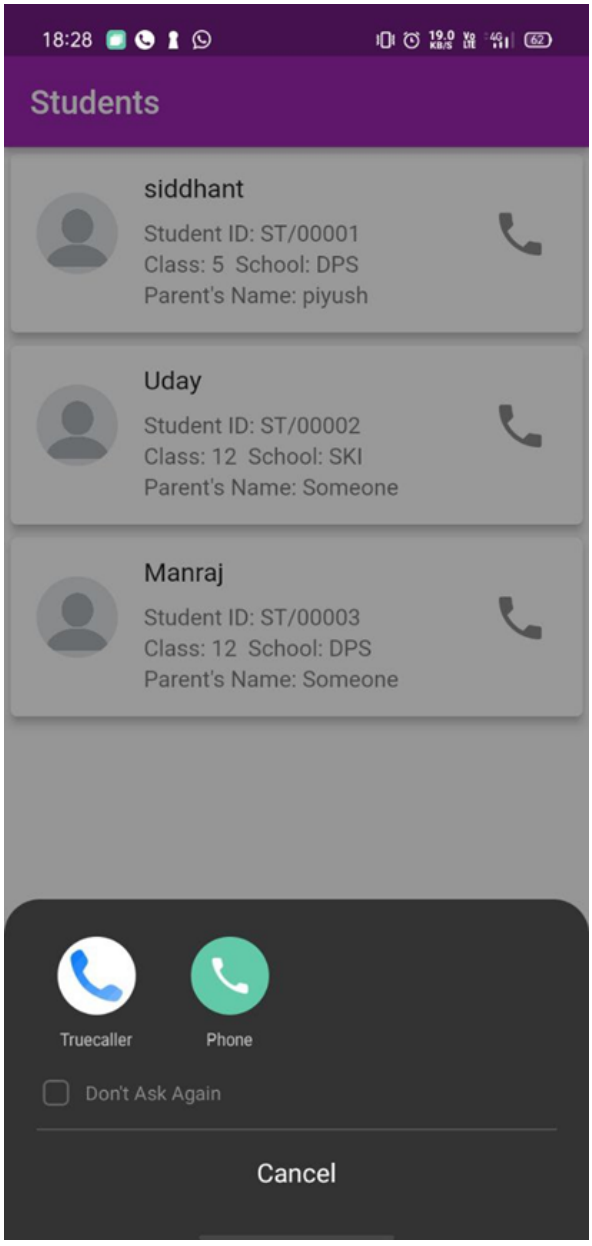
Home Screen Displays List of all enrolled Students

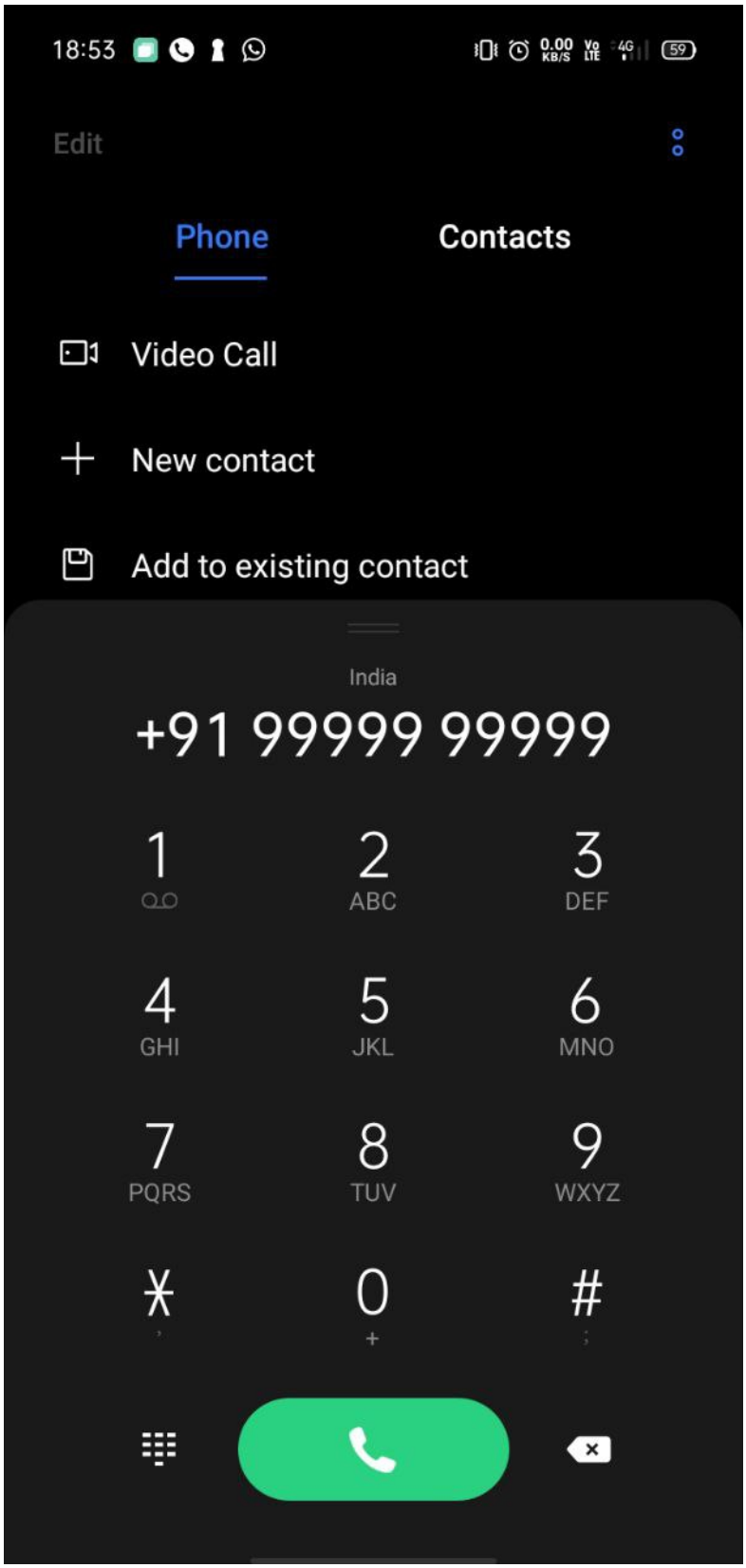


Screen Shot: 2

Clicking on the call button opens the phone dialer

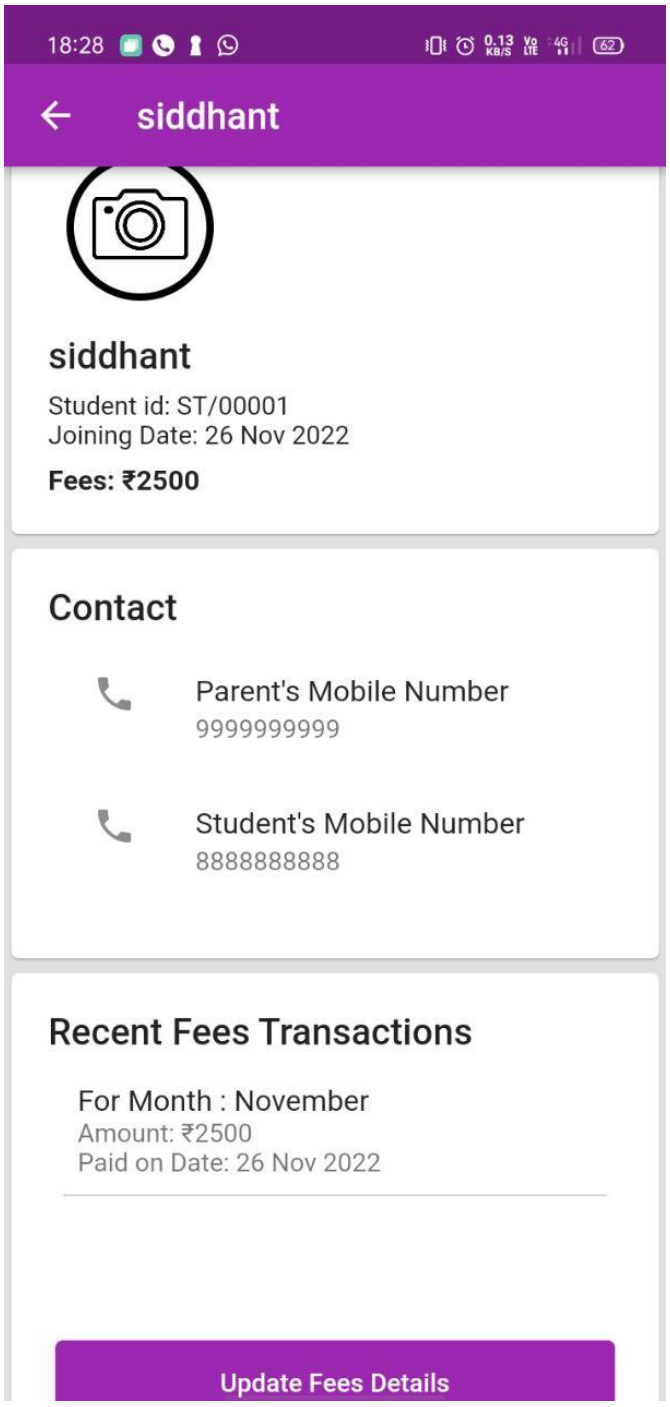





18:53
Edit
Phone
Contacts
Video Call
New contact
Add to existing contact
India
+91 99999 99999
1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PQRS
8 TUV
9 WXYZ
*
0 +
#

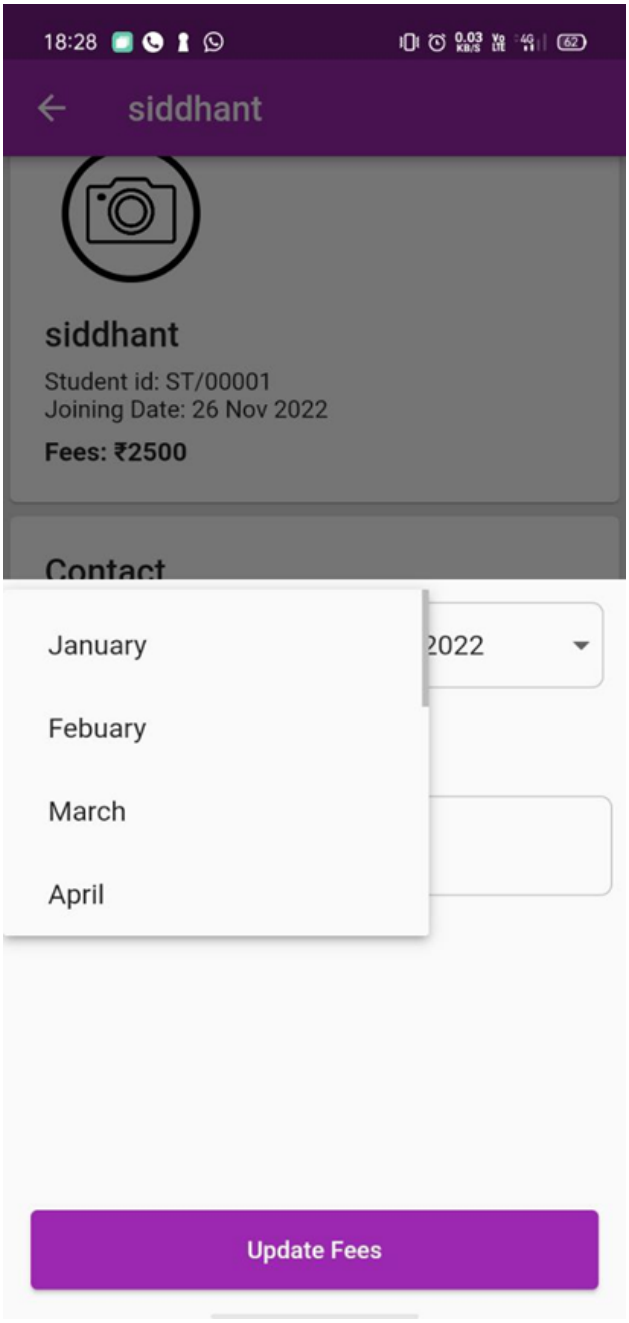
Screen Shot: 3

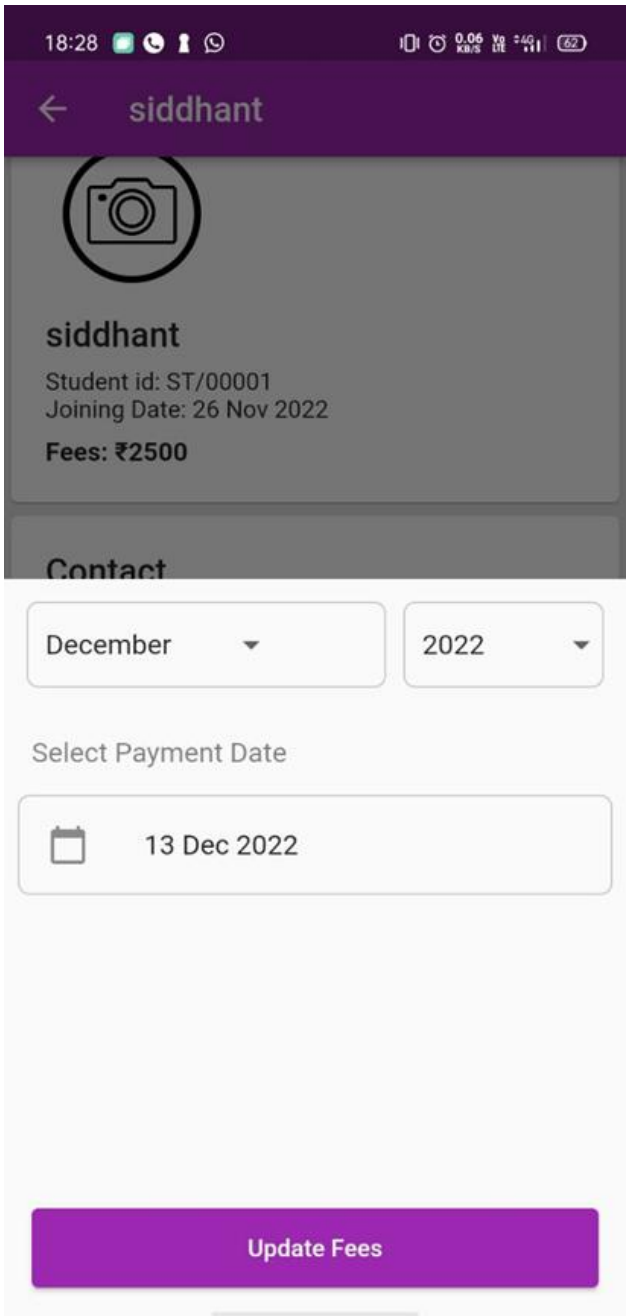
Student's description Screen, Opens on clicking a user tab

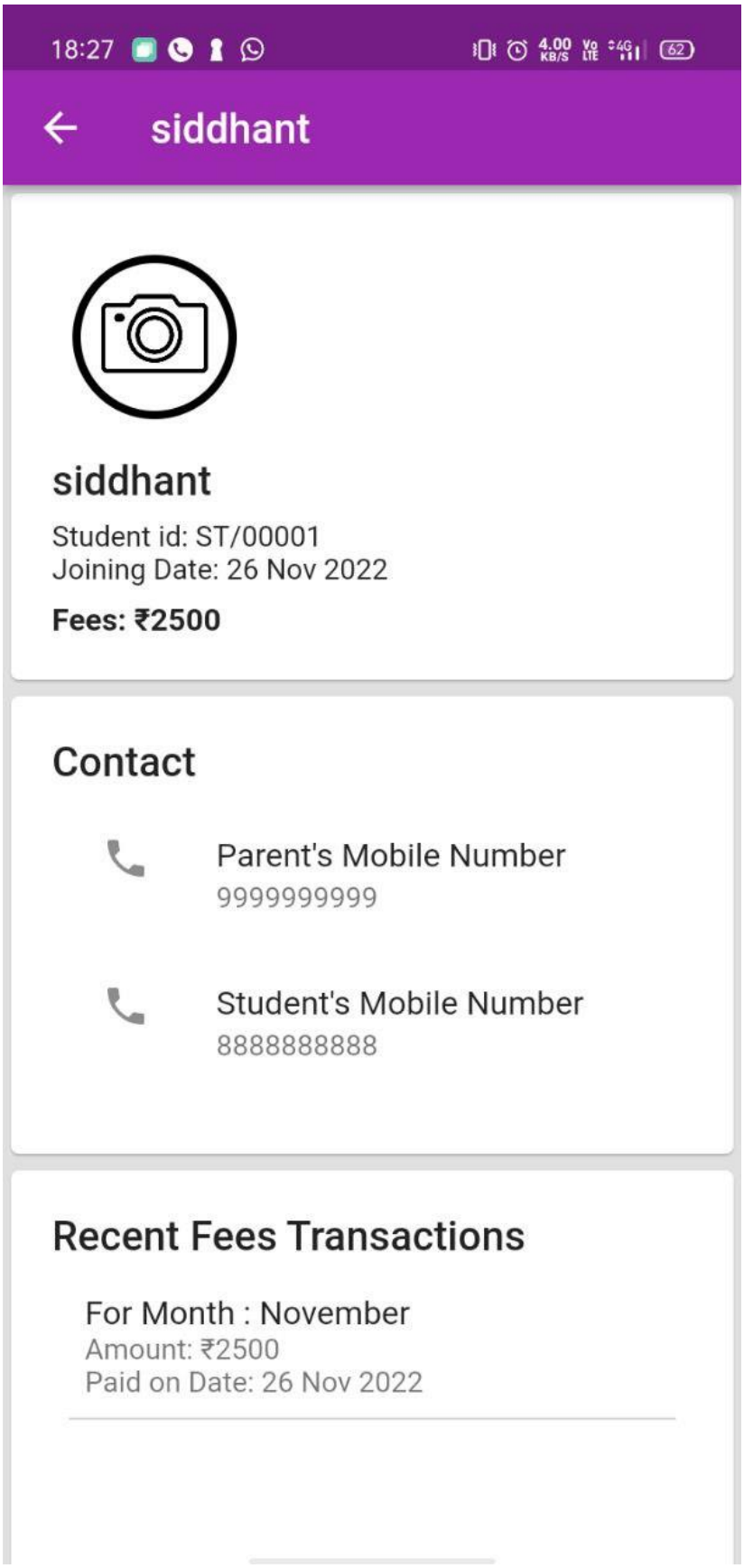


Screen Shot: 4

Update Student fees Transaction Record from student's description screen

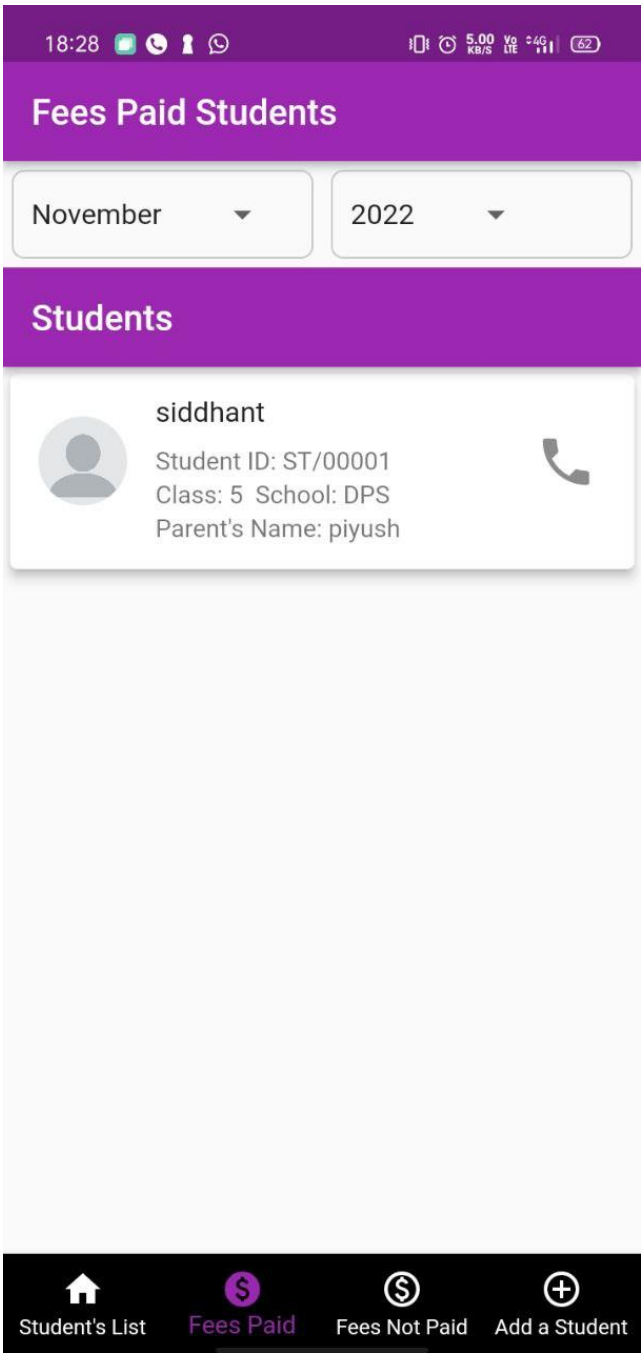





18:27
siddhant
siddhant
Student id: ST/00001
Joining Date: 26 Nov 2022
Fees: ₹2500
Contact
Parent's Mobile Number
9999999999
Student's Mobile Number
8888888888
Recent Fees Transactions
For Month : November
Amount: ₹2500
Paid on Date: 26 Nov 2022

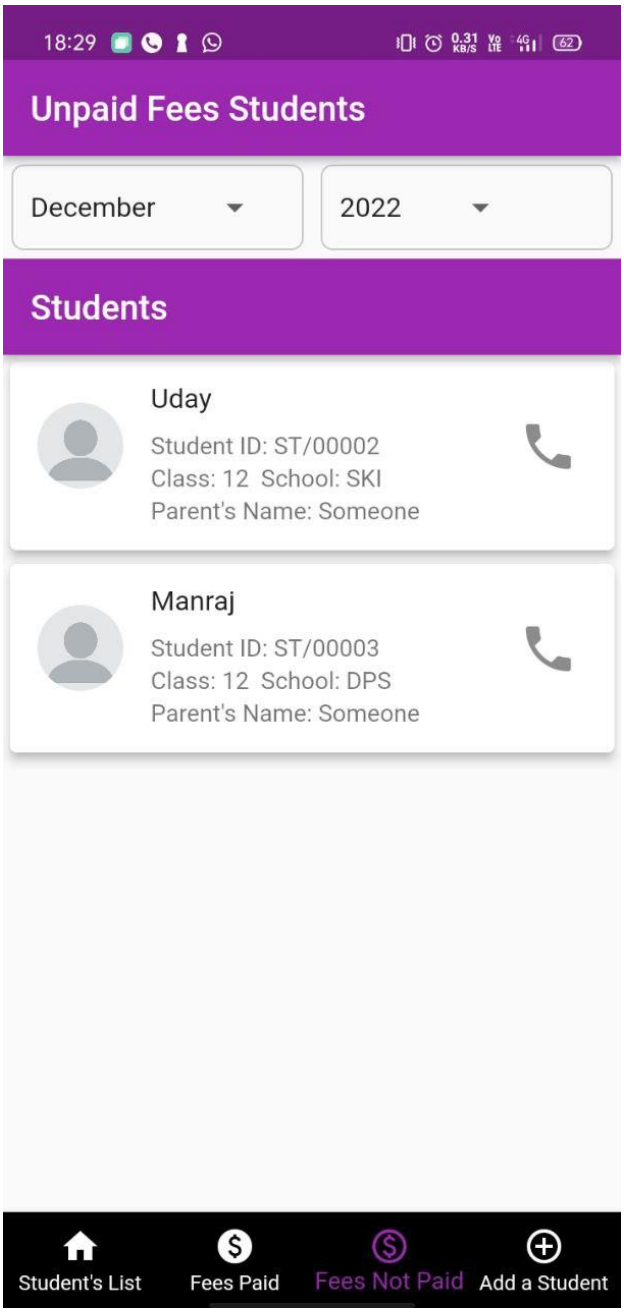
Screen Shot 5:

Screen to see list of all students those who have paid the fees for a specific month



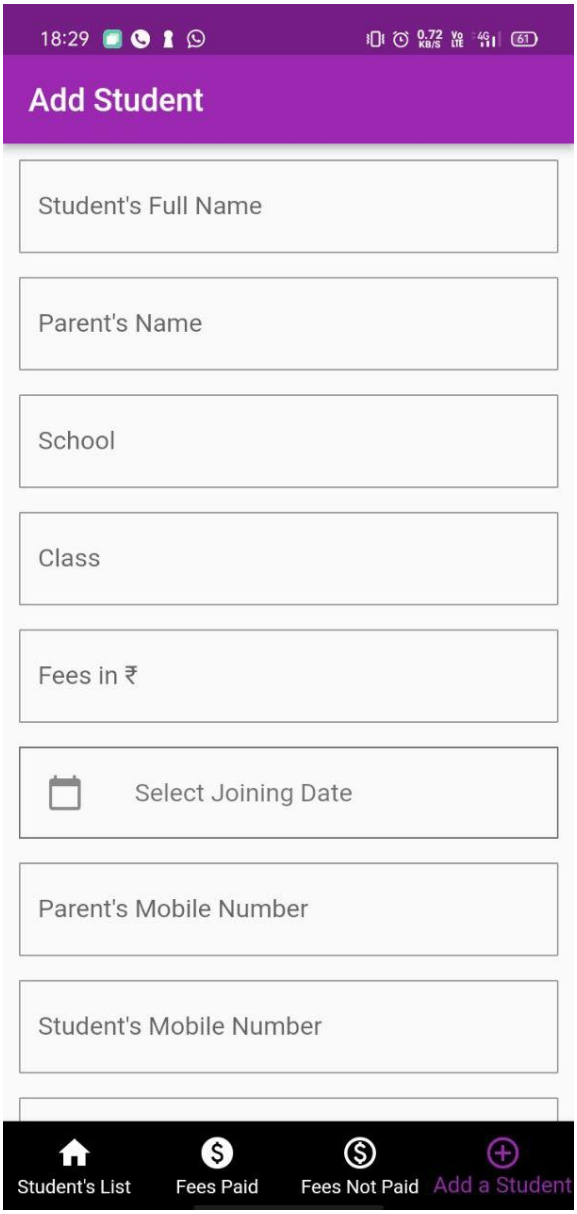
Screen Shot: 6

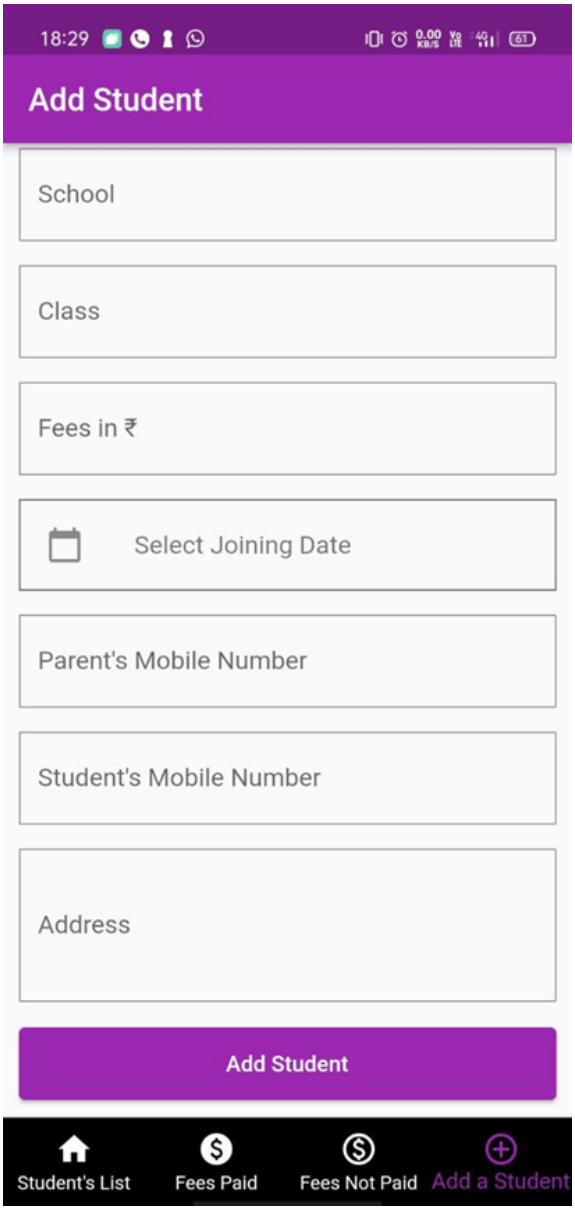
List of all students who have not paid their fees for a specific month



Screen Shot: 7

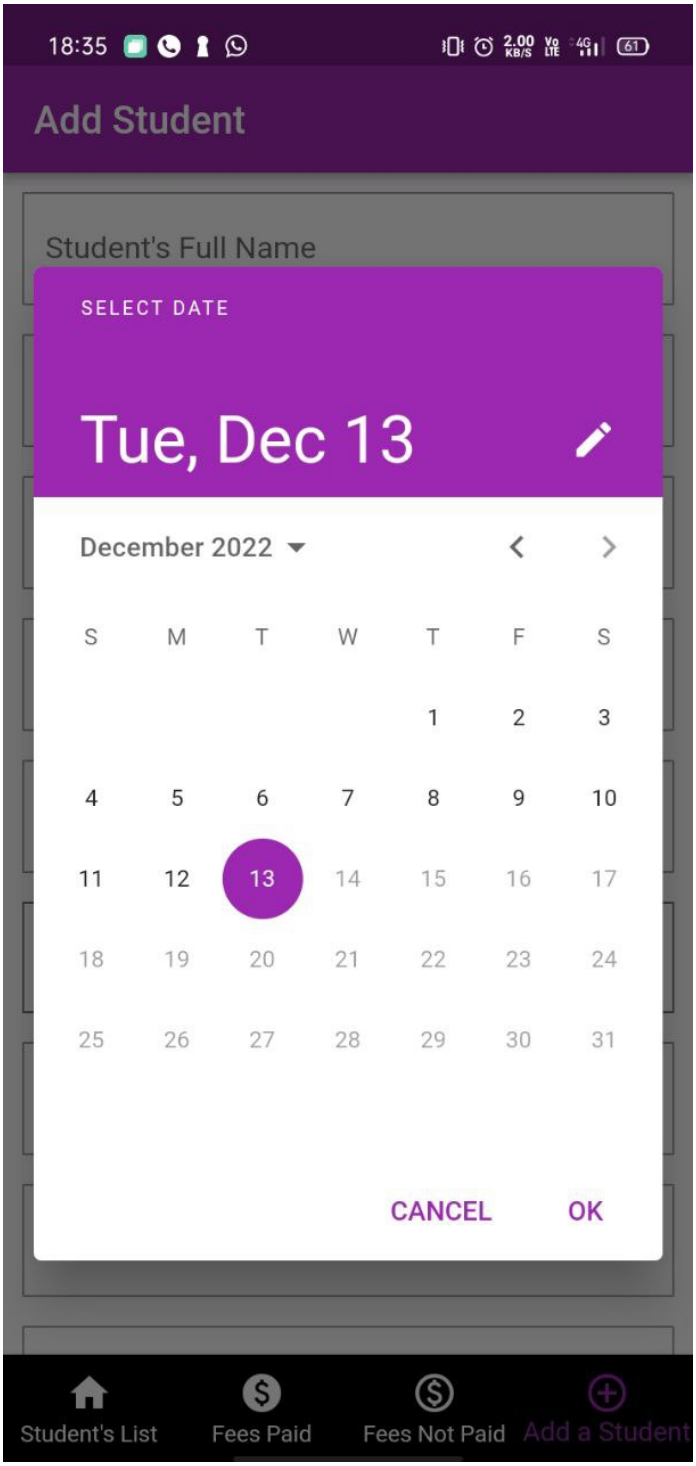
Registration Form to add a new Student





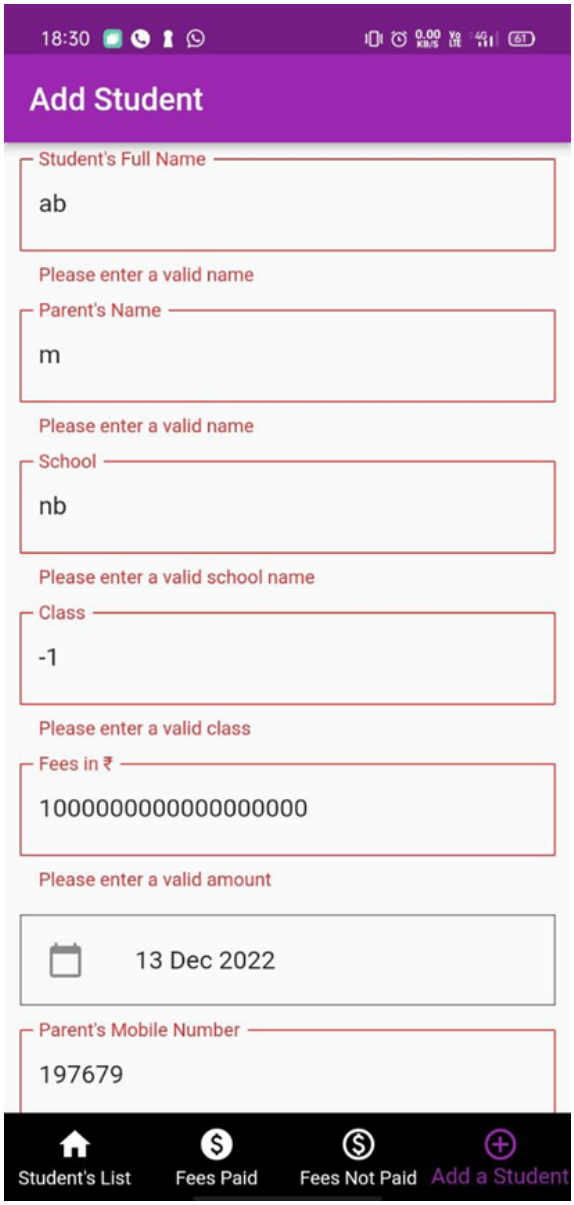
Screen Shot: 8

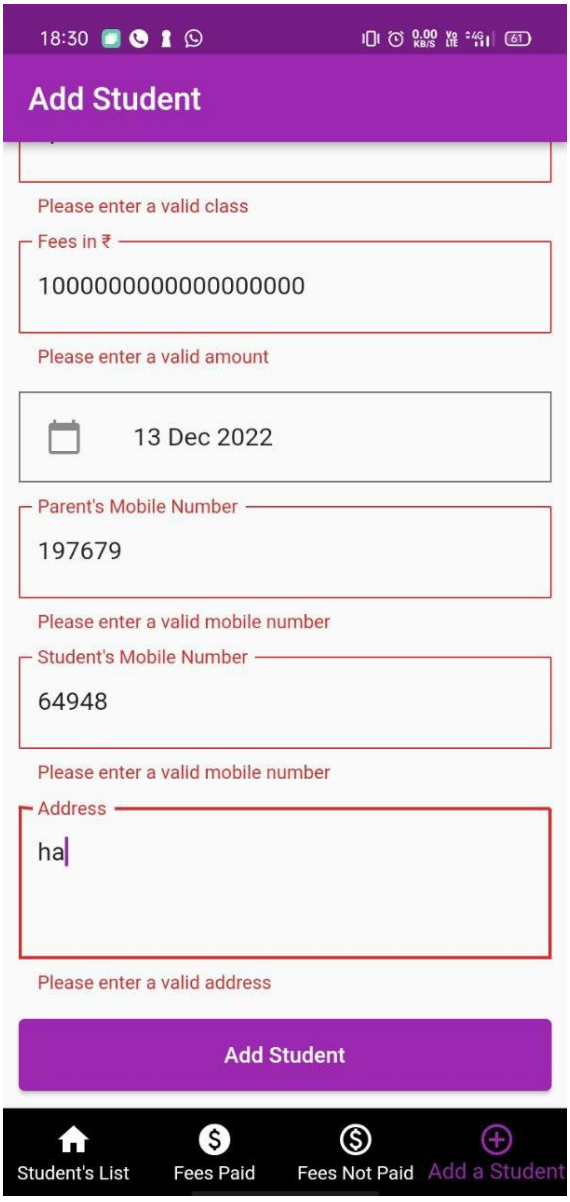
Use of Date Picker Widget in student registration form



Screen Shot: 9

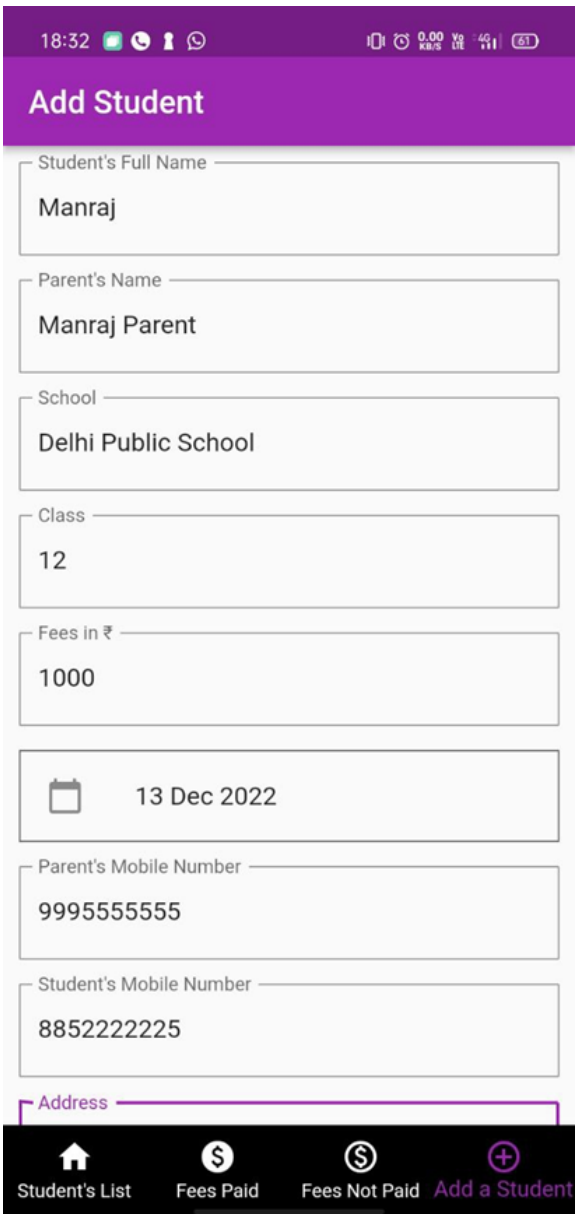
Validations in various form fields

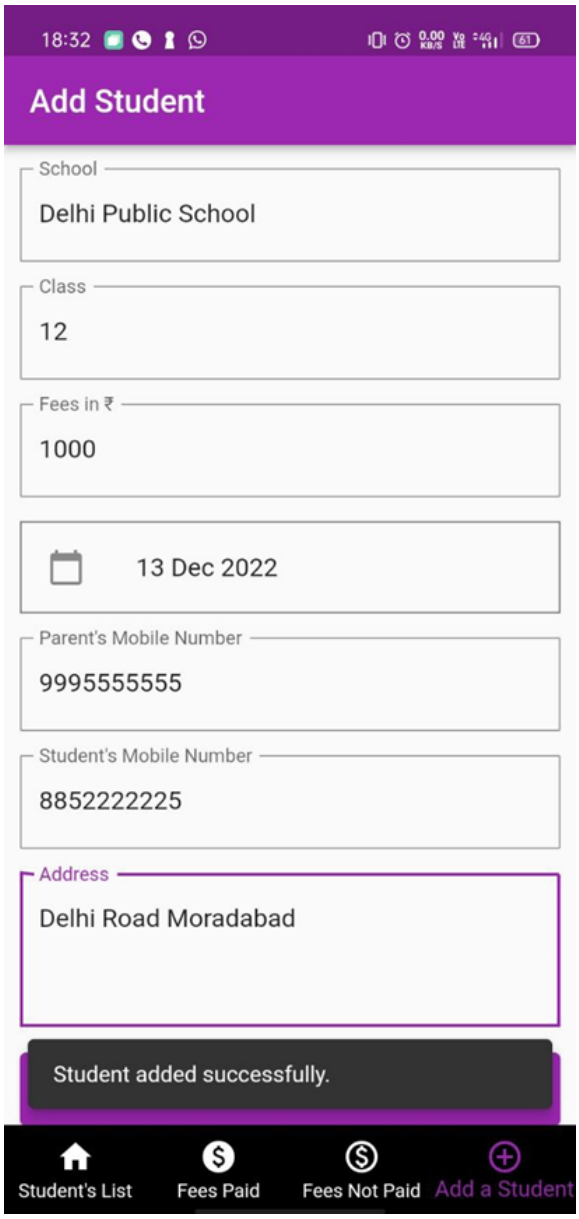




Screen Shot: 10

A Valid Student Form Registration





Screen Shot :11

User automatically added by input stream from firebase realtime database using a stream builder at the home page