PROJECT REPORT

ON

Just Split

SUBMITTED BY

Faiz Ziauddin Ansari

SEAT NO: 701

PROJECT GUIDE

Ms. MANALI PATIL

BSC. (COMPUTER SCIENCE) SEM -V
2021 – 2022

CONDUCTED

AT

CHIKITSAK SAMUHA’S

S. S. & L.S. PATKAR COLLEGE OF ARTS & SCIENCE

AND

V. P. VARDE COLLEGE OF COMMERCE & ECONOMICS

GOREGAON (W). MUMBAI -400062

CHIKITSAK SAMUHA'S

S.S. & L.S. PATKAR COLLEGE OF ARTS & SCIENCE AND V. P. VARDE COLLEGE OF COMMERCE & ECONOMICS

Accredited A Grade by NAAC

S.V.Road, Goregaon (West), Mumbai - 400 062. Tel.: 91-022-28723731/28781188 Fax : 91-022-2874 4755
Website : www.patkarvardecollege.edu.in
E-mail : principal@patkarvardecollege.edu.in info@patkarvardecollege.edu.in

Self-Financed Courses

Project Certificate

This is to certify that Mr. /Ms.__________**Faiz Ziauddin Ansari**_________ of T.Y.B.Sc. Computer Science with University Seat no _**701**_ has completed his/her project titled _______**Just Split**________ under the guidance of Project Guide Mr./Ms.____**Manali Patil**_______ as laid by University of Mumbai in the college during the year 2021-22.

B.Sc. Computer science

Project Guide **Co-ordinator**

Ms Manali Patil **Ms Namrata Shinde**

External Examiner

Date: 18-10-2021

ACKNOWLEDGEMENT

I take this opportunity of submitting this report to express my profound gratitude to the management of “S.S. & L.S. Patkar College of Science and Commerce” for giving me the opportunity to accomplish this project work.

I am also thankful to all the teachers for their kind support and help.

I express my deepest gratitude towards my project guide **Ms Manali Patil** for her valuable and timely advice during the various phases in my project.

I extend my sincere thanks to our respected Head of Department **Ms. Namrata Shinde** for allowing us to use the facilities available.

I would also like to thank all those unnamed but important people and Computer Lab Assistants and supportive staff who directly or indirectly helped me in the completion of this project and to my family and friends without whose support, motivation and encouragement this would not have been possible.

Thanking you,

Faiz Ansari

JUST SPLIT

INDEX

Sr. No.	Contents	Page No.
1	**Introduction**	7
	1.1 Abstract	
	1.2 Objective and Scope of the project	
2	**Requirement Specification**	9
	2.1 Functional Requirement	
	2.2 Non-Functional Requirement	
	2.3 Hardware Requirement	
	2.4 Software Requirement	
3	**System planning**	12
	3.1 Gantt Chart	
4	**System Design**	13
	4.1 Methodology Adopted(object oriented)	
	4.2 Architecture (Spiral Model)	
	4.3 UML Diagrams (For Object oriented methodology)	
	4.3.1 Use-Case Diagram	
	4.3.2 Class Diagram	
	4.3.3 Activity Diagram	
	4.3.4 Sequence Diagram	
	4.3.5 Database Design: ER Diagram	
5	**System Implementation:**	23

	5.1 Code implementation	
6	**Results**	28
	6.1 Test Cases	
	6.2 Screen Shot	
7	**Conclusion and Future Scope**	38
8	**References**	39

1. Introduction

1.1 Abstract

JustSplit is an application share expenses with friends and family. JustSplit is the easiest way to share expenses with friends and family and stop stressing about "who owes who." organize group bills for households, trips, and more. Create groups or private friendships for any splitting situation Add expenses, IOUs, or informal debts in INR currency, with support for offline entry.

Expenses are backed up online so any everyone can log in, view their balances, and add expenses, Keep track of who should pay next, or settle up by recording cash payments or using integrations.

JustSplit is great for:

- Roommates splitting rent and apartment bills

- Group trips around the world

- Splitting a vacation house for skiing or at the beach

- Weddings and bachelor/bachelorette parties

- Couples sharing relationship costs

- Friends and co-workers who go out to lunch or dinner together frequently

- Loans and IOUs between friends

1.2 Objective of the project

Did you pay for your friends and feel anxious about asking money back! Just Split all expenses with friends, family or anybody and bring peace to your mind. mission is to reduce the stress and awkwardness that money places on our most important relationships.

1.2 The scope of the project

This Report describes all the requirements for the project. The purpose of this research is to provide a all information for the combination of both structured and unstructured information of our project "JustSplit" Application.

Split expenses equally or unequally by percentages, shares, or exact amounts

- Create bills that recur monthly
- Add multiple payers on a single expense
- See total balances with a person across multiple groups and private expenses
- Custom user Profile
- View your edit history for changes to an expense

2. Requirement Specification

2.1 Functional Requirement

The functional requirements specify what the product must do. They relate to the actions that the product must carry out in order to satisfy the fundamental reasons for its existence. The functional requirements must fully describe the actions that the intended product can perform.

android application

- User Registration or Login
- Email verification
- Profile section
- User can add their friends from contact list
- User can create group of our friends
- Add expenses with our friends or group of friends by defining the amout and description of their expenses Select the appropriate Option in list
  - Paid by you Split equally,
  - You owe the full amout,
  - They owe the full amount,
  - Paid by other person and split Equally,

and select the appropriate date.

- Manage and delete their expenses.
- Manage your personal notes.

2.2 Non-Functional Requirement:

The non-functional requirement add functionally to the product's Existence, but are needed to make the product perform in the desired manner. In Just Split, these requirements are met with utmost care. The design is made so that even the new users can understand the navigation and walk through of the application. This program requires above average specification of hardware but most users can run it on their system with included options.

- Platform constraints: This Application run on android Phone. It require 1GB ram to run smoothly.
- Usability: Easy to Use
- Speed and Responsiveness: Execution of operations should be fast.

2.3 HARDWARE REQUIREMENT:

- Processor type: Minimum: Intel i3 or more.
- Minimum Ram : 8gb .
- Recommended: 16 GB.
- Processor speed: 1.5 GHz or higher.

Android Phone

- OS: Android 5(Lollipop) or higher

Ram

- Minimum: 2GB
- Recommended: 4 GB.

- Processor : Qualcomm or Snapdragon(1.0 GHz or higher).

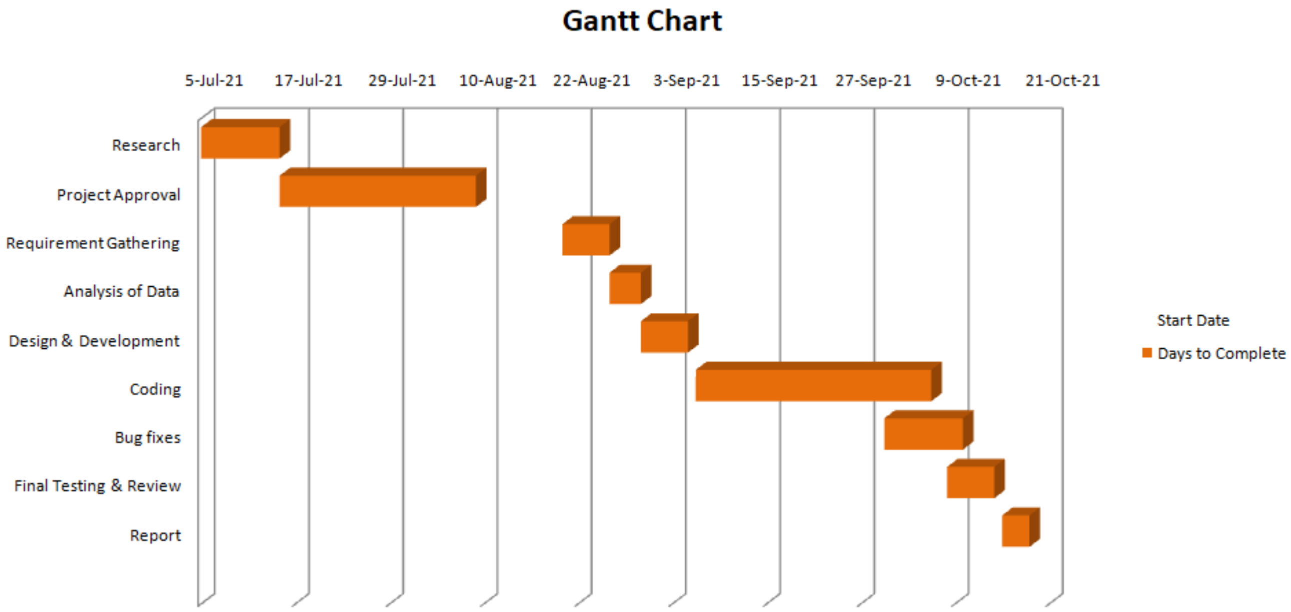
2.4 Software Requirements:

- Operating system :
  Windows 7 or above
- Language :
  Flutter
- Software Application :
  Android Studio/Visual Studio Code
- SDK : 2.0 or higher

3.0 System planning

3.1 Gantt Chart

Gantt chart, commonly used in project management, is one of the most popular and useful ways of showing activities (tasks or events) displayed against time. On the left of the chart is a list of the activities and along the top is a suitable time scale. Each activity is represented by a bar, the position and length of reflects the start date, duration and end date of the activity.



4. System Design

4.1 Methodology Adopted

OBJECT ORIENTED PROGRAMMING:

Object-oriented programming (OOP) refers to a type of computer programming (software design) in which programmers define not only the datatype of a data structure, but also the types of operations (functions) that can be applied to the data structure. In this way, the data structure becomes an object that includes both data and functions. In addition, programmers can create relationships between one object and another .For example, objects can inherit characteristics from other object

THE BASICS CONCEPTS: If you are new to object-oriented programming languages, you will need to know a few basics before you can get started with code.:

ABSTRACTION : The process of picking out (abstracting) common features of objects and procedures. CLASS: A category of objects. The class defines all the common properties of the different objects that belong to it.

ENCAPSULATION : The process of combining elements to create a new entity. A procedure is a type of encapsulation because it combines a series of computer instructions. INFORMATION HIDING: The process of hiding details of an object or function. Information hiding is a powerful programming technique because it reduces complexity.

INHERITANCE : a feature that represents the "is a" relationship between different classes.

INTERFACE : the languages and codes that the applications use to communicate with each other and with the hardware.

MESSAGE : Message passing is a form of communication used in parallel programming and object-oriented programming.

OBJECT : a self-contained entity that consists of both data and procedures to manipulate the data. POLYMORPHISM: A programming language's ability to process objects differently depending on their data type or class.

PROCEDURE : a section of a program that performs a specific task

Google Firebase : is a Google-backed application development software that enables developers to develop iOS, Android and Web apps.

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

Flutter is a cross-platform UI toolkit that is designed to allow code reuse across operating systems such as **iOS** and **Android**, while also allowing applications to **interface** directly with underlying platform services. The goal is to enable developers to deliver high-performance apps that feel natural on different platforms, embracing differences where they exist while sharing as much code as possible.

During development, Flutter **apps** run in a VM that offers stateful hot reload of changes without needing a full recompile. For release, Flutter apps are compiled directly to machine code, whether **Intel** x64 or ARM instructions, or to **JavaScript** if targeting the web. The framework is open source, with a permissive **BSD** license, and has a thriving ecosystem of third-party packages that supplement the core library functionality.

DART : Dart is a programming language designed for client development, such as for the web and mobile apps. It is developed by Google and can also be used to build server and desktop applications.

Dart is an object-oriented, class-based, garbage-collected language with C-style syntax.] Dart can compile to either native code or JavaScript. It supports interfaces, mixins, abstract classes, reified generics, and type inference.

- **Dart platform**
- **Flutter engine**
- **Foundation library**
- **Design-specific widgets**
- **Flutter Development Tools (DevTools)**

- Dart platform

Flutter apps are written in the Dart language and make use of many of the language's more advanced features.

On Windows, macOS, and Linux Flutter runs in the Dart virtual machine, which features a just-in-time execution engine. While writing and debugging an app, Flutter uses Just In Time compilation, allowing for "hot reload", with which modifications to source files can be injected into a running application. Flutter extends this with support for stateful hot reload, where in most cases changes to source code are reflected immediately in the running app without requiring a restart or any loss of state.

- Flutter engine

Flutter's engine, written primarily in C++, provides low-level rendering support using Google's Skia graphics library. Additionally, it interfaces with platform-specific SDKs such as those provided by Android and iOS.[10] The Flutter Engine is a portable runtime for hosting Flutter applications. It implements Flutter's core libraries, including animation and graphics, file and network I/O, accessibility support, plugin architecture, and a Dart runtime and compile toolchain. Most developers interact with Flutter via the Flutter Framework, which provides a reactive framework and a set of platform, layout, and foundation widgets.

- Foundation library

The Foundation library, written in Dart, provides basic classes and functions that are used to construct applications using Flutter, such as APIs to communicate with the engine.

- Design-specific widgets

The Flutter framework contains two sets of widgets that conform to specific design languages: Material Design widgets implement Google's design language of the same name, and Cupertino widgets implement Apple's iOS .

Widgets are generally defined in three basic types: Stateful widgets, Stateless widgets, and Inherited widgets. Being the central class hierarchy in the Flutter framework the three basic types of widgets are used in the construction of every Flutter application.

External Packages :

Intl : It defines the Intl class, with the default locale and methods for accessing most of the internationalization mechanisms. This library also defines the DateFormat, NumberFormat, and BidiFormatter classes.

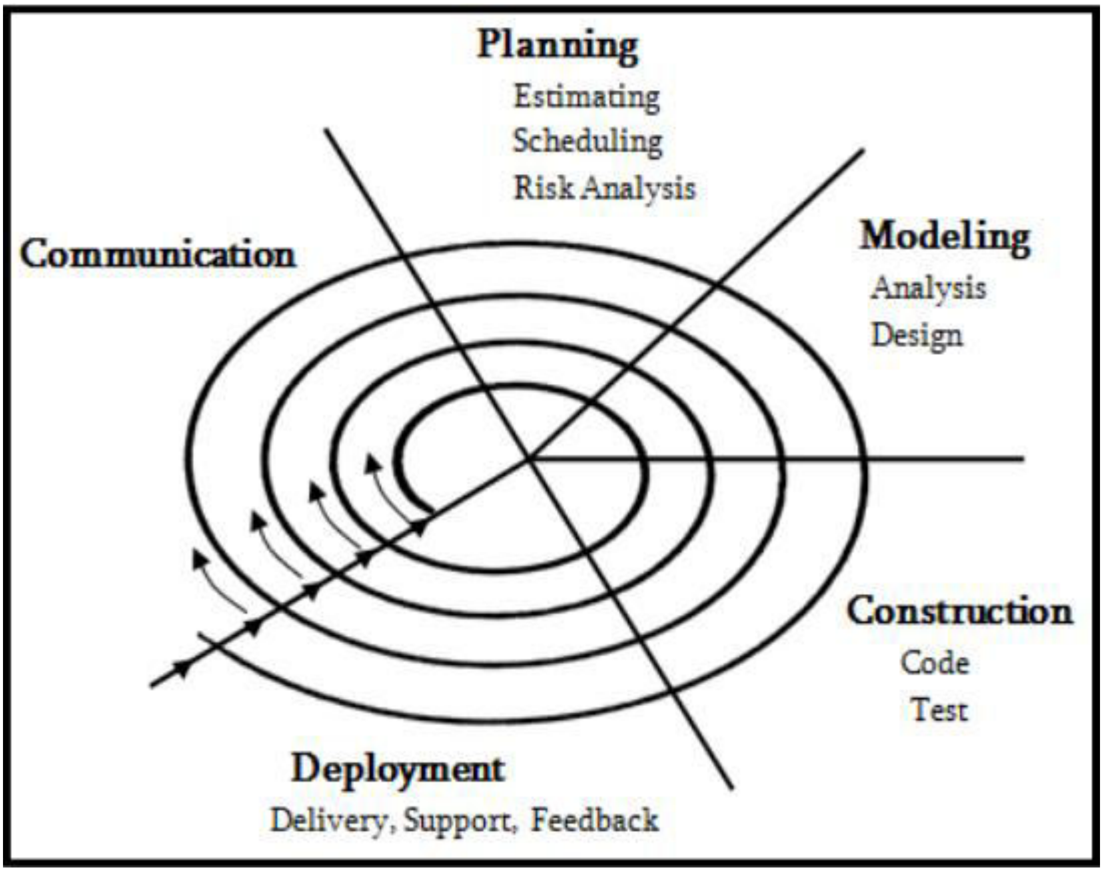
Theme : Themes are an integral part of UI for any application. Themes are used to design the fonts and colors of an application to make it more presentable. In Flutter, the Theme widget is used to add themes to an application.

Country codes : Country codes is an helper package that provides country details given a particular localization, such as dial codes

4.2 Architecture

Spiral model:

Is a combination of sequential and prototype model. This model is best used for large projects which involves continuous enhancements. There are specific activities which are done in one iteration (spiral) where the output is a small prototype of the large software. The same activities are then repeated for all the spirals till the entire software is build.



Functions of Spiral Model:

Requirements : Requirements are gathered from the customers and the objectives are identified, elaborated and analyzed at the start of every phase. Then alternative solutions possible for the phase are proposed in this quadrant.

Planning : All the possible solutions are evaluated to select the best possible solution. Then the risks associated with that solution is identified and the risks are resolved using the best possible strategy. At the end of this quadrant, Prototype is built for the best possible solution.

Modeling : In this Section we will Analysis and requirements, according to needs. We will create a design of our projects and analysis of project. Internal testing and Deployments are done in Modeling.

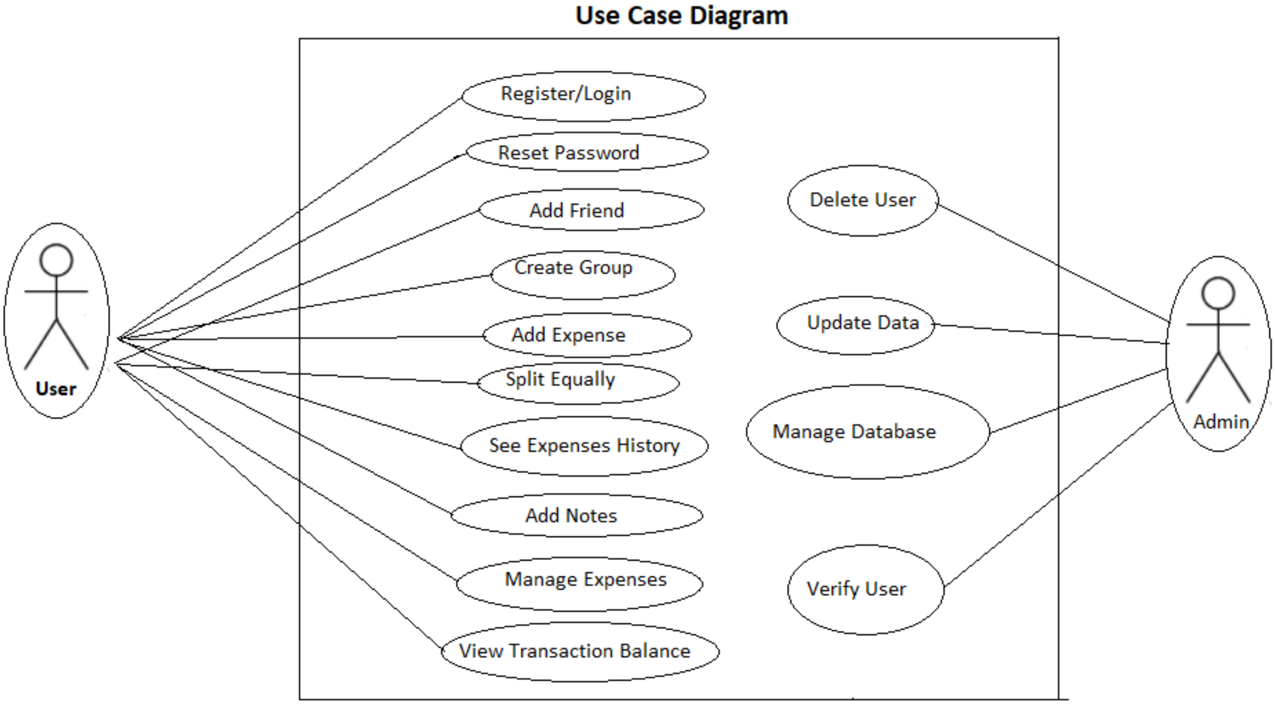
Construction : The team delivered high-quality working software in priority order, which was created in accordance with the changing needs of our potential users. What‟s more important, the team could deploy this solution into a pre-production testing/QA sandbox for system integration testing.

Deployment : Once the software testing phase is over and no bugs or errors left in the system then the final deployment process starts. Based on the feedback given by the project manager, the final software is released and checked for deployment issues if any.

4.3 UML Diagrams

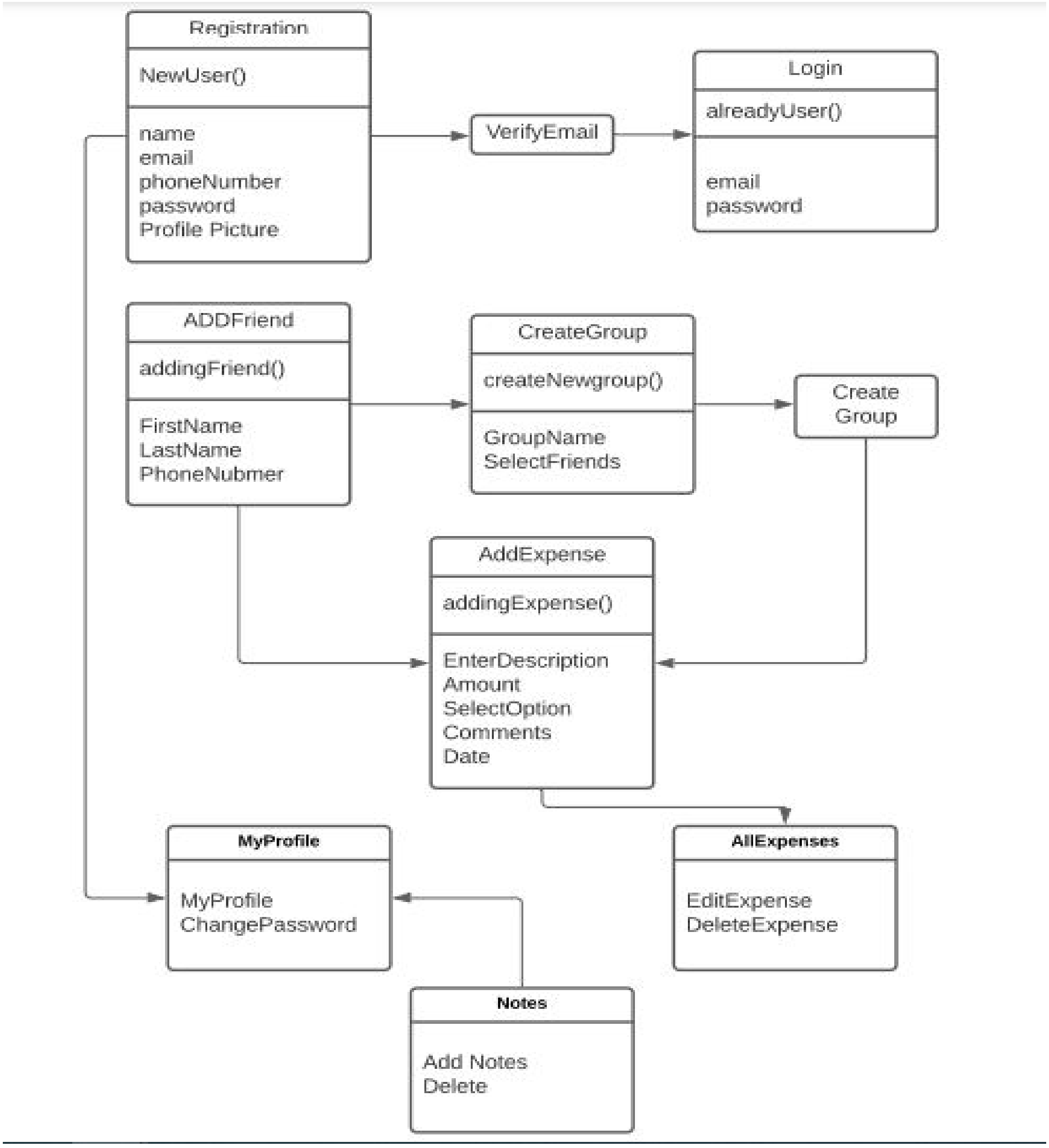
A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well.

4.3.1 Use Case Diagram



4.3.2 Class Diagram

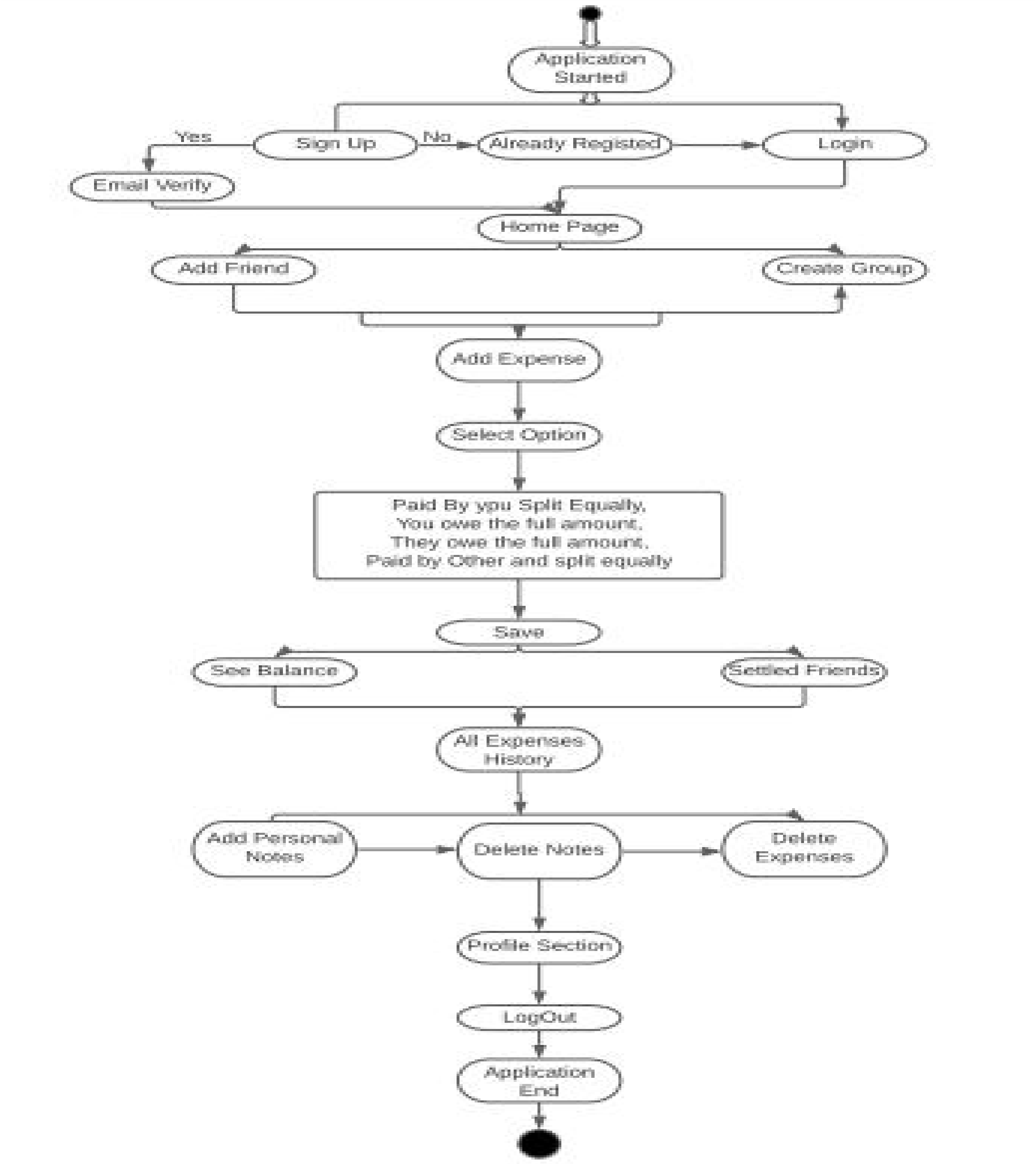
Class diagram is a static diagram. ... The class diagrams are widely used in the modeling of objectoriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints.



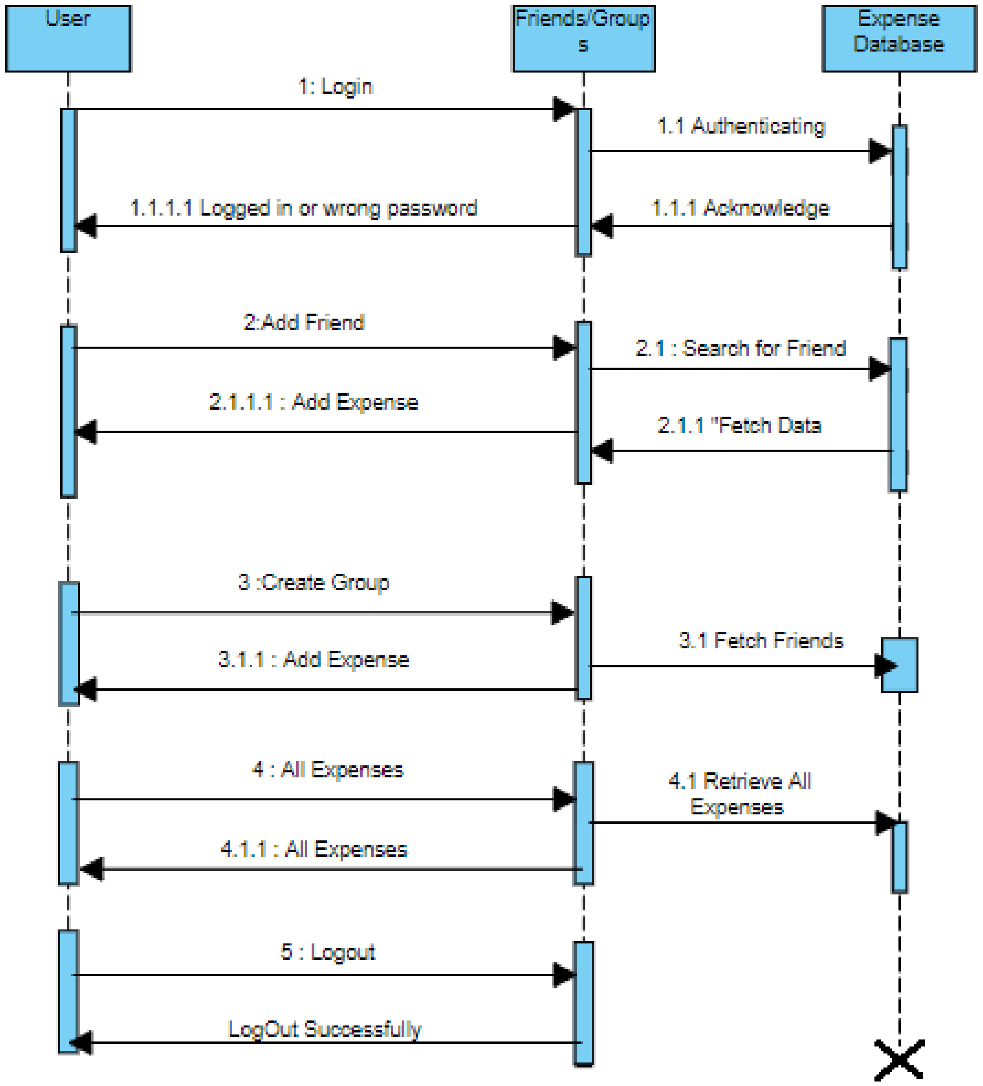
4.3.3 Activity Diagram:

An Activity Diagram visually presents a series of actions of control is system similar to a flowchart or a dataflow .

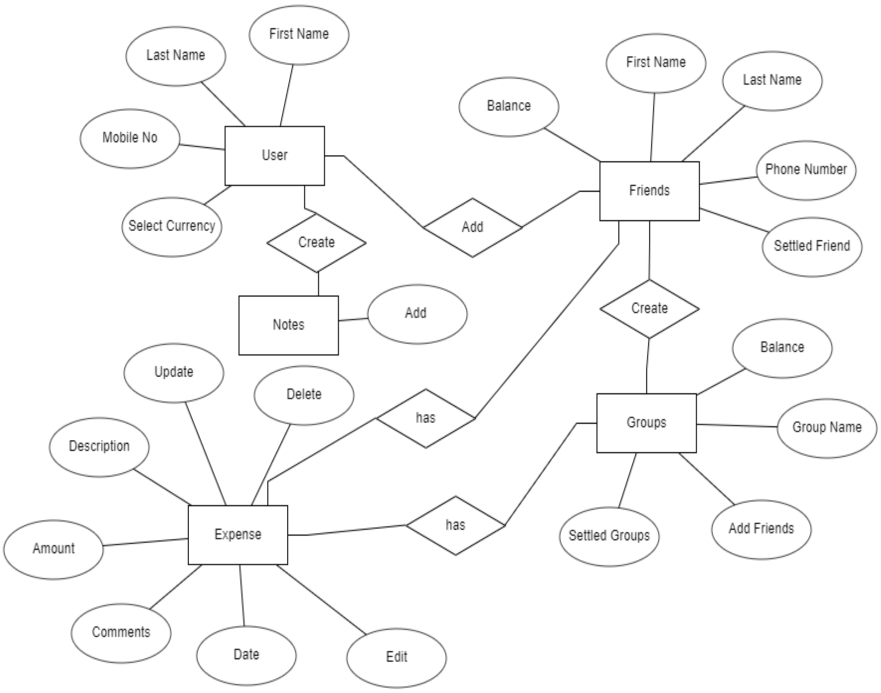
An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.



4.3.4 Sequence Diagram



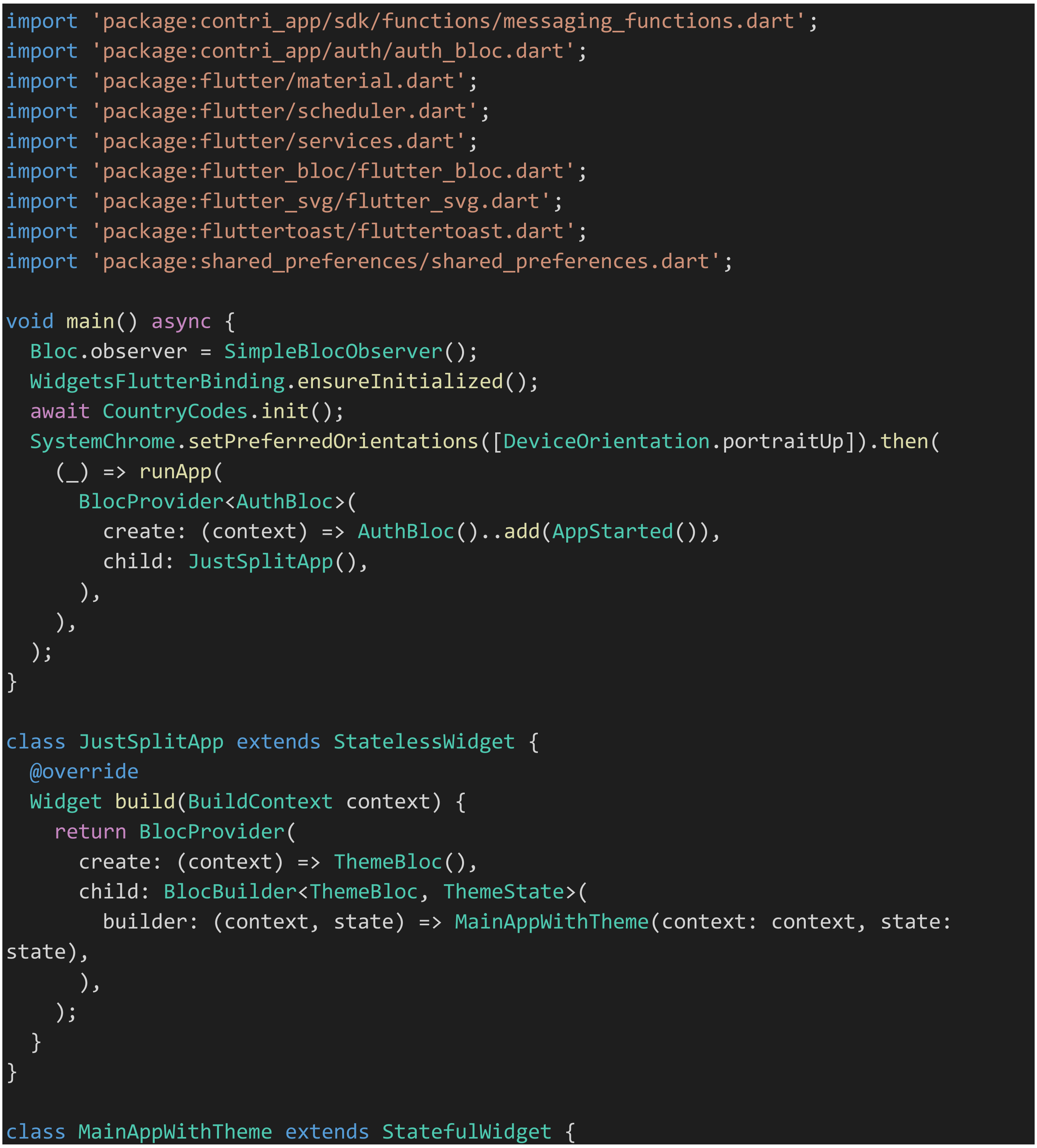
4.3.5 Database Design: ER Diagram


First Name
Last Name
Mobile No
User
Select Currency
Create
Notes
Add
Add
Balance
First Name
Last Name
Phone Number
Friends
Settled Friend
Create
Balance
Groups
Group Name
Settled Groups
Add Friends
Update
Delete
has
Description
Expense
has
Amount
Comments
Date
Edit

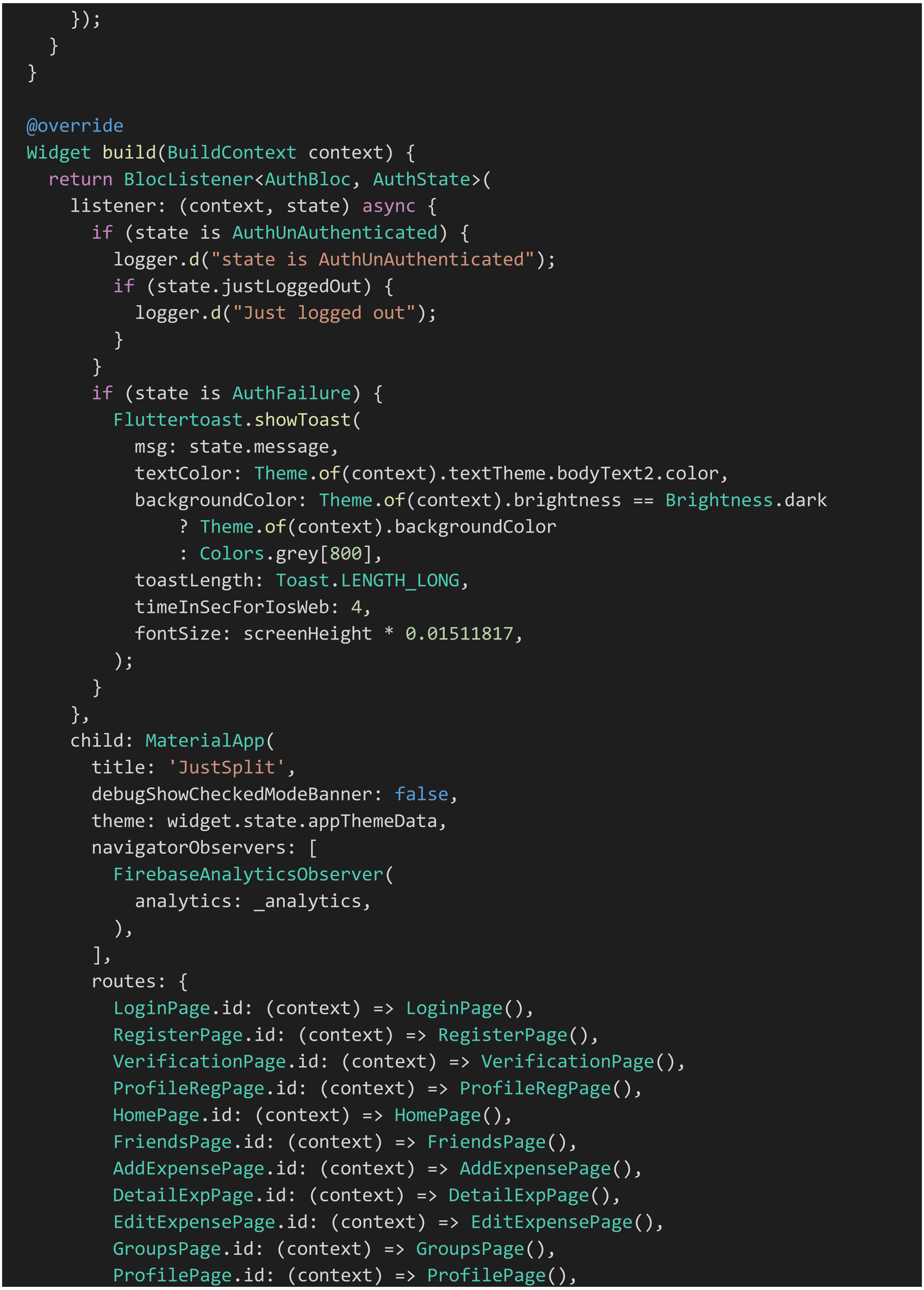
5. System Implementation:

5.1 Code implementation

Sample code :



```dart
import 'package:contri_app/sdk/functions/messaging_functions.dart';
import 'package:contri_app/auth/auth_bloc.dart';
import 'package:flutter/material.dart';
import 'package:flutter/scheduler.dart';
import 'package:flutter/services.dart';
import 'package:flutter_bloc/flutter_bloc.dart';
import 'package:flutter_svg/flutter_svg.dart';
import 'package:fluttertoast/fluttertoast.dart';
import 'package:shared_preferences/shared_preferences.dart';

void main() async {
  Bloc.observer = SimpleBlocObserver();
  WidgetsFlutterBinding.ensureInitialized();
  await CountryCodes.init();
  SystemChrome.setPreferredOrientations([DeviceOrientation.portraitUp]).then(
    (_) => runApp(
      BlocProvider<AuthBloc>(
        create: (context) => AuthBloc()..add(AppStarted()),
        child: JustSplitApp(),
      ),
    ),
  );
}

class JustSplitApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return BlocProvider(
      create: (context) => ThemeBloc(),
      child: BlocBuilder<ThemeBloc, ThemeState>(
        builder: (context, state) => MainAppWithTheme(context: context, state:
state),
      ),
    );
  }
}

class MainAppWithTheme extends StatefulWidget {
  const MainAppWithTheme({
    Key key,
    @required this.context,
    @required this.state,
  }) : super(key: key);

  final BuildContext context;
  final ThemeState state;

  @override
  _MainAppWithThemeState createState() => _MainAppWithThemeState();
}

class _MainAppWithThemeState extends State<MainAppWithTheme> {
  final FirebaseAnalytics _analytics = FirebaseAnalytics();

  int getSystemTheme() {
    final _sysBrightness = SchedulerBinding.instance.window.platformBrightness;
    return _sysBrightness == Brightness.dark ? 0 : 1;
  }

  /// To get preferred theme
  Future<void> loadSavedThemeData() async {
    SharedPreferences _prefs = await SharedPreferences.getInstance();
    final _appThemeIndex = _prefs.getInt('theme_id') ?? getSystemTheme();
    final _appTheme = AppTheme.values[_appThemeIndex];
    BlocProvider.of<ThemeBloc>(context).add(ThemeChanged(appTheme: _appTheme));
  }

  Map<String, dynamic> _currencyData = {};
  int _systemTheme = 1;

  @override
  void initState() {
    loadSavedThemeData().then((_) => logger.i("Got Saved Theme"));
    _systemTheme = getSystemTheme();
    _analytics.logAppOpen().then((_) => logger.v("AnalyticsEvent: AppStarted"));
    super.initState();
  }

  @override
  void didChangeDependencies() async {
    super.didChangeDependencies();

    if (_currencyData.isEmpty) {
      final _d = await getCurrencyData(context);
      setState(() {
        _currencyData = _d;
```



```
    });
  }
}

@override
Widget build(BuildContext context) {
  return BlocListener<AuthBloc, AuthState>(
    listener: (context, state) async {
      if (state is AuthUnAuthenticated) {
        logger.d("state is AuthUnAuthenticated");
        if (state.justLoggedOut) {
          logger.d("Just logged out");
        }
      }
      if (state is AuthFailure) {
        Fluttertoast.showToast(
          msg: state.message,
          textColor: Theme.of(context).textTheme.bodyText2.color,
          backgroundColor: Theme.of(context).brightness == Brightness.dark
              ? Theme.of(context).backgroundColor
              : Colors.grey[800],
          toastLength: Toast.LENGTH_LONG,
          timeInSecForIosWeb: 4,
          fontSize: screenHeight * 0.01511817,
        );
      }
    },
    child: MaterialApp(
      title: 'JustSplit',
      debugShowCheckedModeBanner: false,
      theme: widget.state.appThemeData,
      navigatorObservers: [
        FirebaseAnalyticsObserver(
          analytics: _analytics,
        ),
      ],
      routes: {
        LoginPage.id: (context) => LoginPage(),
        RegisterPage.id: (context) => RegisterPage(),
        VerificationPage.id: (context) => VerificationPage(),
        ProfileRegPage.id: (context) => ProfileRegPage(),
        HomePage.id: (context) => HomePage(),
        FriendsPage.id: (context) => FriendsPage(),
        AddExpensePage.id: (context) => AddExpensePage(),
        DetailExpPage.id: (context) => DetailExpPage(),
        EditExpensePage.id: (context) => EditExpensePage(),
        GroupsPage.id: (context) => GroupsPage(),
        ProfilePage.id: (context) => ProfilePage(),
          SettingsPage.id: (context) => SettingsPage(),

        },
        home: BlocBuilder<AuthBloc, AuthState>(
          builder: (context, state) {
            logger.i("Configuring FCM");
            NotificationHandler().configureFcm(context);
            logger.i("FCM configured");
            initializeUtils(context);
            currencySymbol = getCurrencySymbol(currencyData: _currencyData);

            if (state is AuthUnInitialized)
              return LoginPage();
            else if (state is AuthUnAuthenticated)
              return LoginPage();
            else if (state is AuthAuthenticated) {
              _analytics.logEvent(name: "home_page_loaded");
              return HomePage();
            } else if (state is AuthNeedsVerification)
              return VerificationPage();
            else if (state is AuthNeedsProfileComplete)
              return ProfileRegPage();
            else
              return SafeArea(
                child: Theme(
                  data: _systemTheme == 0 ? darkTheme : lightTheme,
                  child: Builder(
                    builder: (context) => Scaffold(
                      body: Container(
                        width: screenWidth,
                        child: Column(
                          mainAxisAlignment: MainAxisAlignment.center,
                          crossAxisAlignment: CrossAxisAlignment.center,
                          children: [
                            const SizedBox(height: 70.0),
                            SvgPicture.asset(
                              'assets/icons/misc/JustSplit_logo_v3-08.svg',
                              width: screenWidth * 0.25,
                              placeholderBuilder: (context) => Container(),
                            ),
                            const SizedBox(height: 75.0),
                            CircularProgressIndicator(
                              backgroundColor: Theme.of(context).brightness ==
Brightness.light
                                  ? Theme.of(context).primaryColor
                                  : Theme.of(context).colorScheme.primary,
                            )
                          ],
                        ),
                      ),
                    ),
                  ),
                ),
              );
          },
        ),
      ),
    );
  }
}
```

6. Results

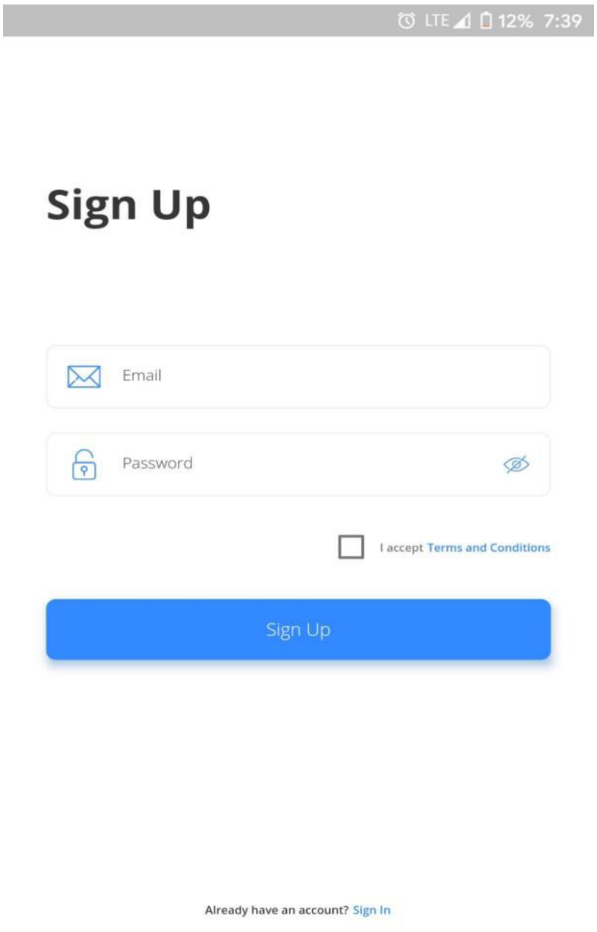
6.1 Test Cases

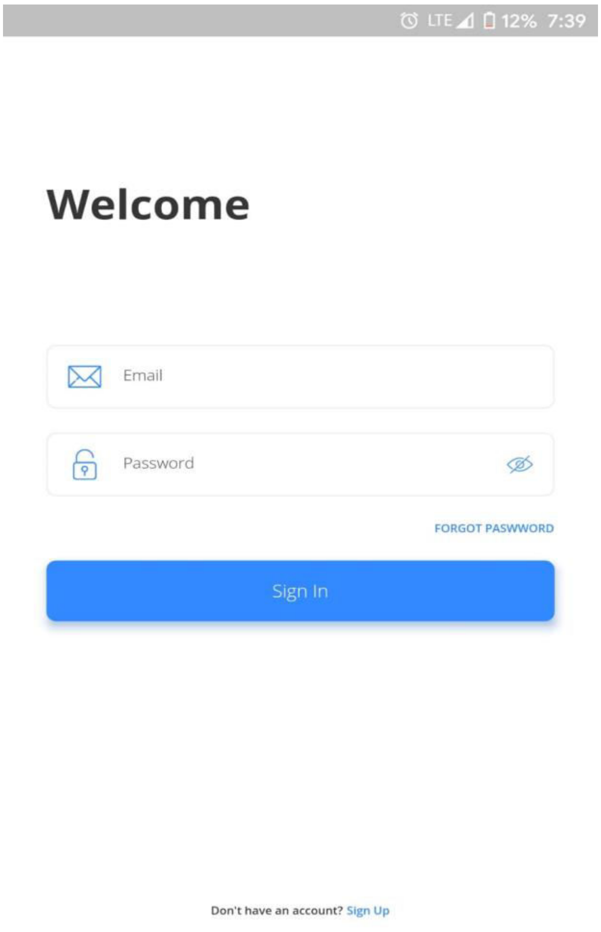
SR. No	**Test Cases**	**Expected Result**	**Status**	**Output**	**Test Result**
1	User SignUp	Register Successfully	Performed	Yes	Success
2	User Login	Login Successfully	Performed	Yes	Success
3	Home Page	Displayed Data	Performed	Yes	Success
4	Add Friend	Added	Performed	Yes	Success
5	Create Group	Create	Performed	Yes	Success
6	Add Expense In Friend/Group	Added Expense Friend/Group	Performed	Yes	Success
7	Paid by you split equally	Split equally	Performed	Yes	Success
8	You owe the full amount	Worked	Performed	Yes	Success
9	They owe the full amout	Worked	Performed	Yes	Success
10	Paid by other person split Equally	Worked	Performed	Yes	Success
11	Delete/Update Expense	Worked	Performed	Yes	Success
12	Add Notes	Added	Performed	Yes	Success

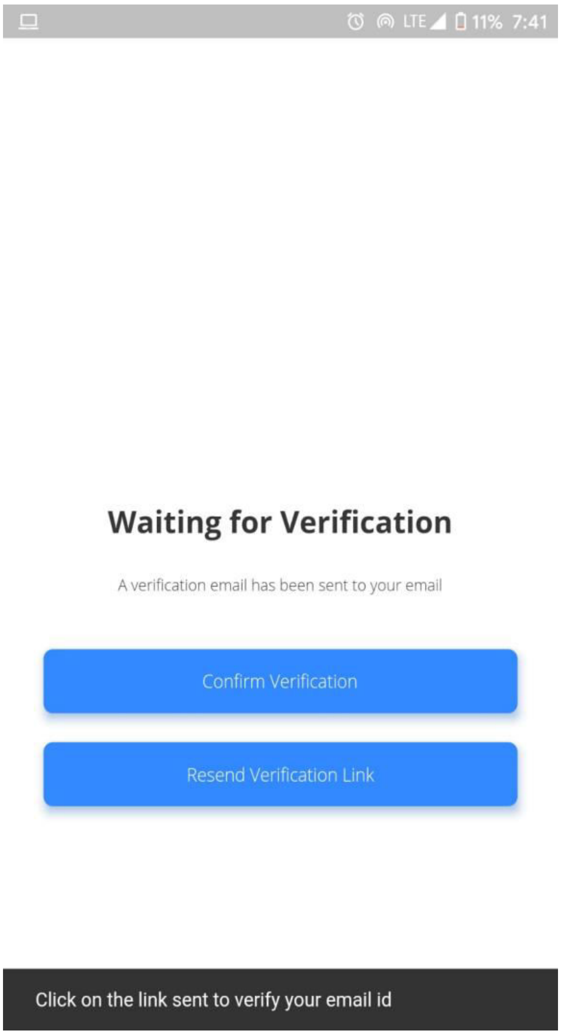
13	Profile Edit	Edited	Performed	Yes	Success
14	Change Password	Changed	Performed	Yes	Success
15	Contact Us Redirect	Redirected	Performed	Yes	Success
16	Logout	End Session	Performed	Yes	Success

6.2 Screen Shot

Application UI:


LTE 12% 7:39
Sign Up
Email
Password
I accept Terms and Conditions
Sign Up
Already have an account? Sign In

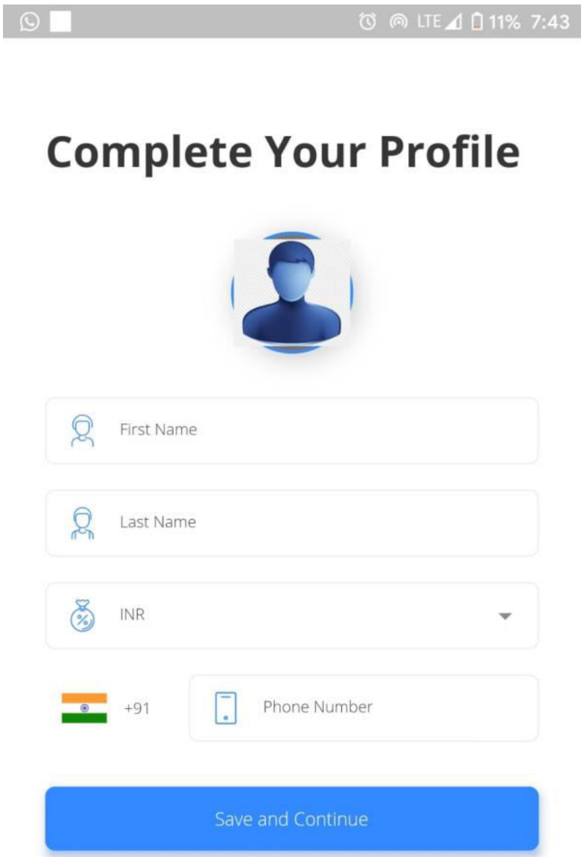




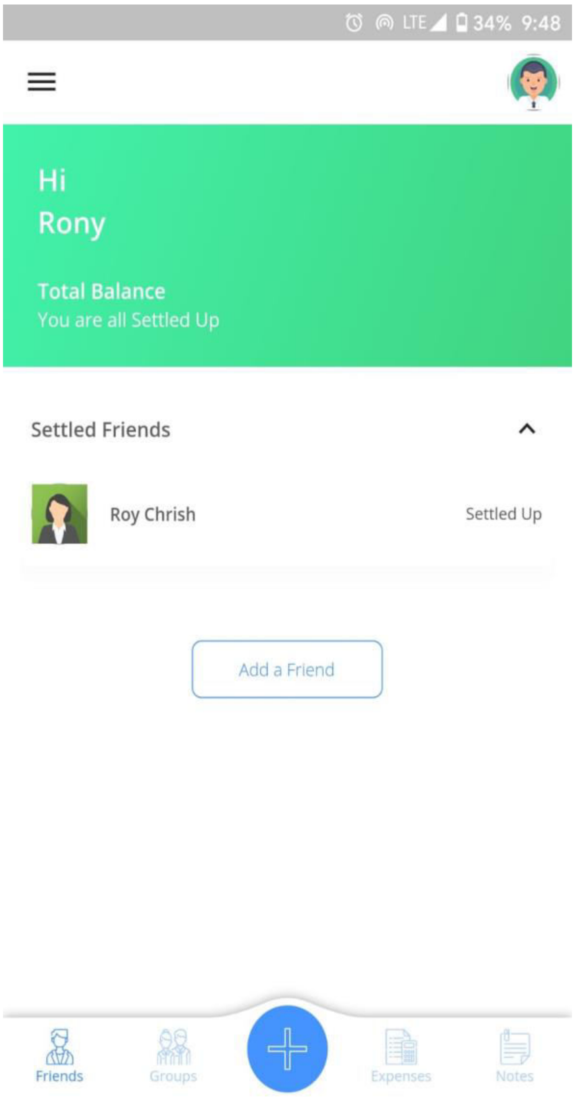
Waiting for Verification

A verification email has been sent to your email

Click on the link sent to verify your email id



Complete Your Profile

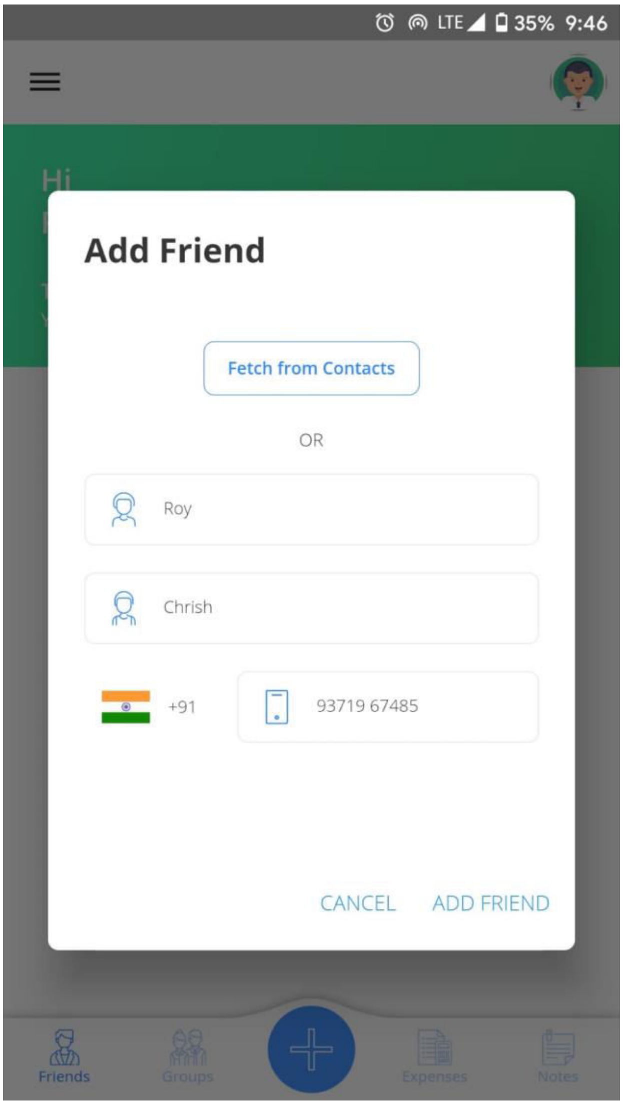


Hi
Rony

Total Balance
You are all Settled Up

Settled Friends

Roy Chrish Settled Up

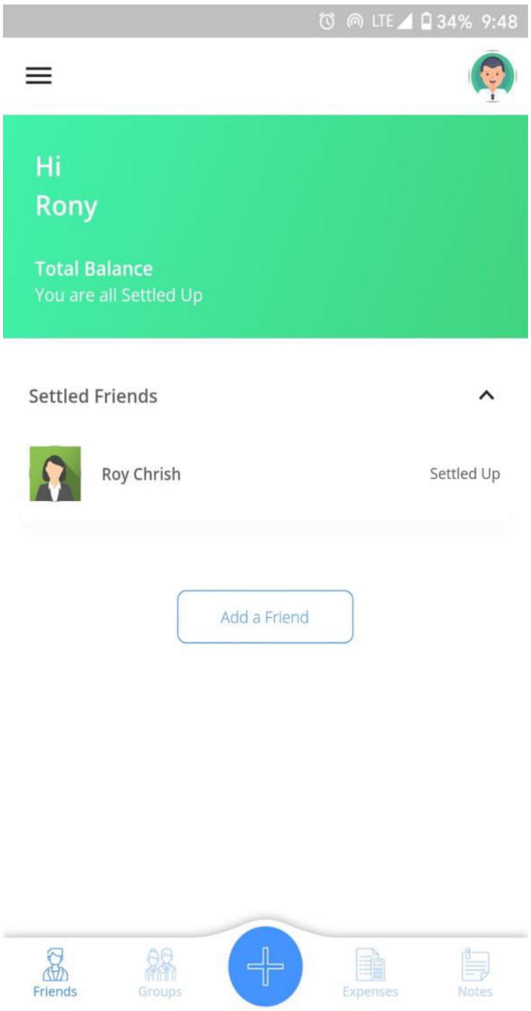


Add Friend

Roy

Chrish

+91 93719 67485

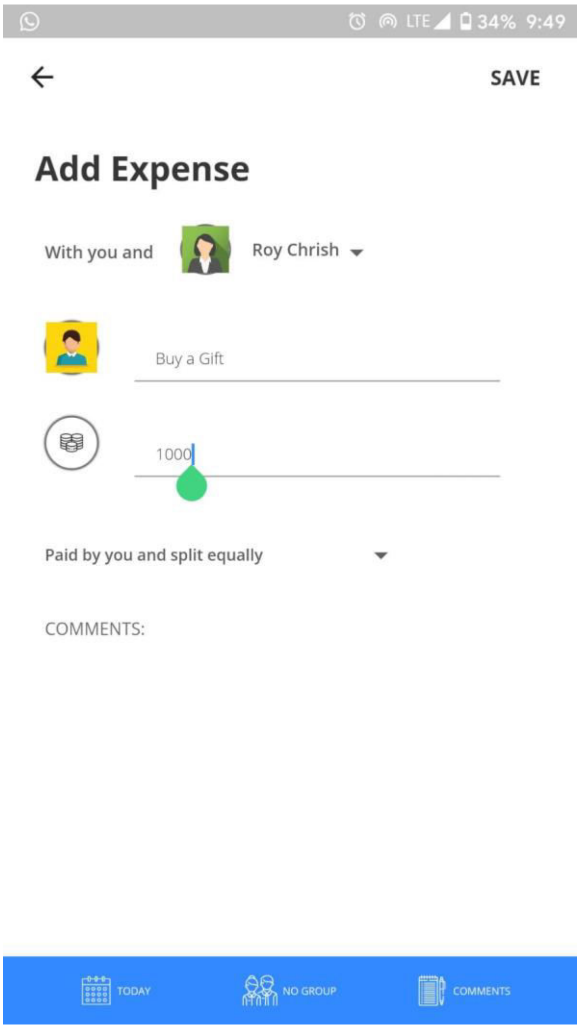


Hi
Rony

Total Balance
You are all Settled Up

Settled Friends

Roy Chrish	Settled Up



Add Expense

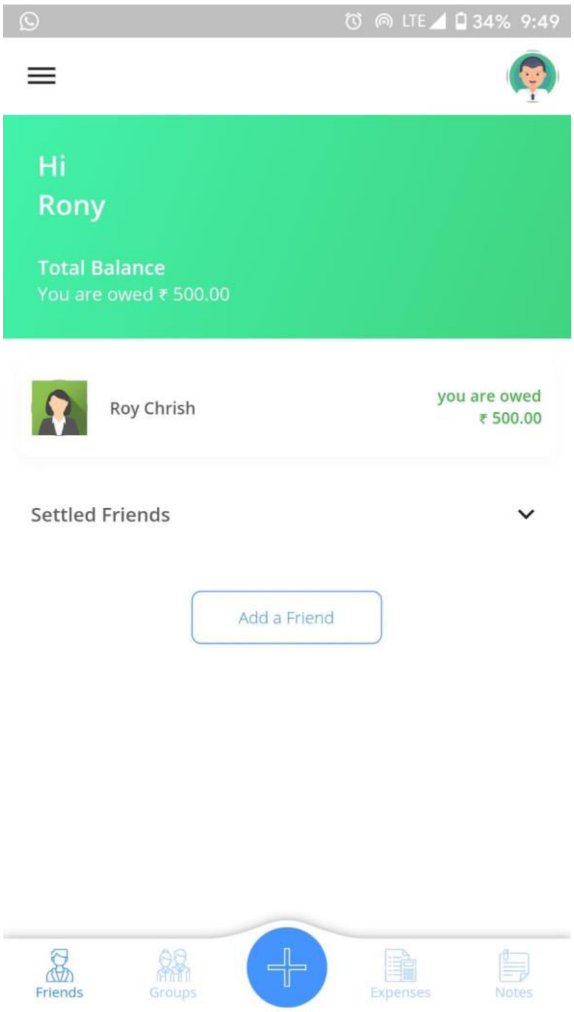
With you and Roy Chrish

Buy a Gift

1000

Paid by you and split equally

COMMENTS:

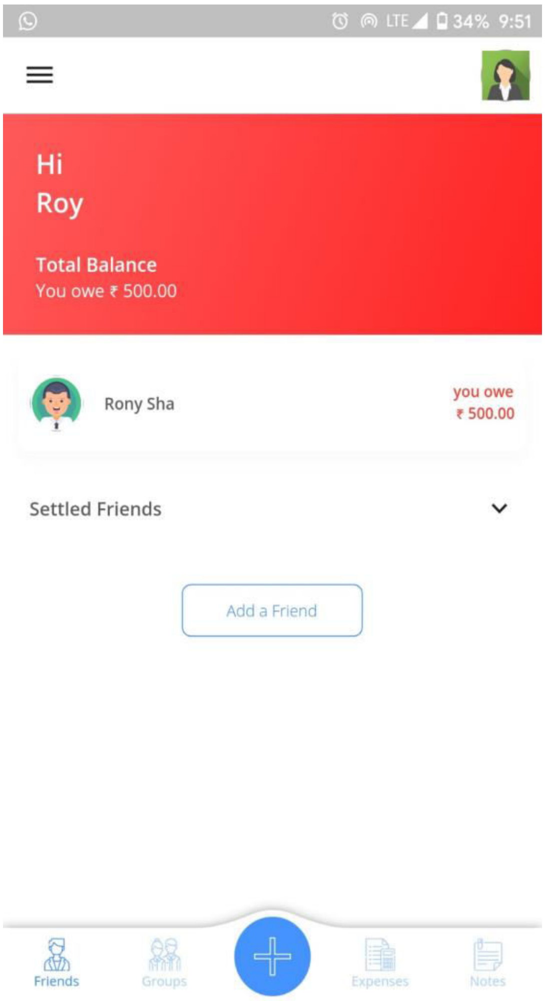


Hi
Rony

Total Balance
You are owed ₹ 500.00

Roy Chrish
you are owed
₹ 500.00

Settled Friends

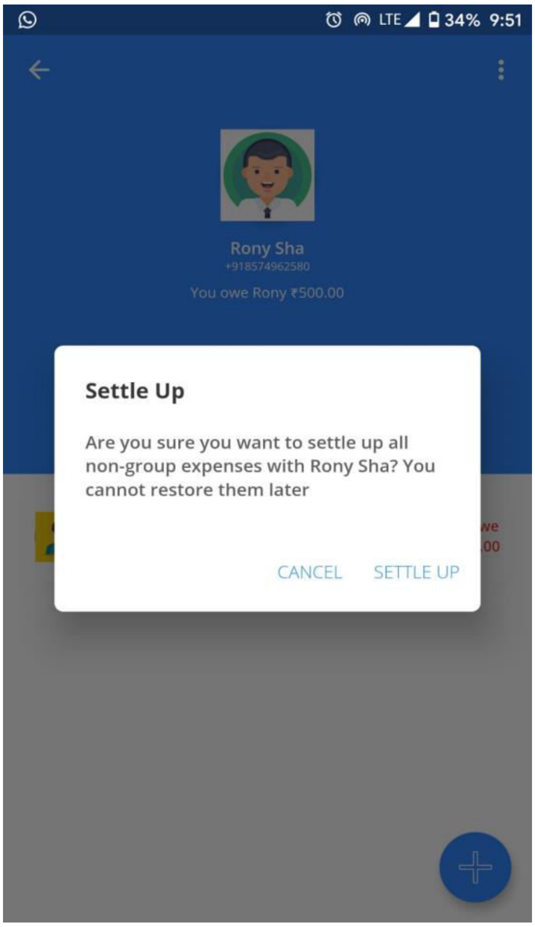


Hi
Roy

Total Balance
You owe ₹ 500.00

Rony Sha
you owe
₹ 500.00

Settled Friends



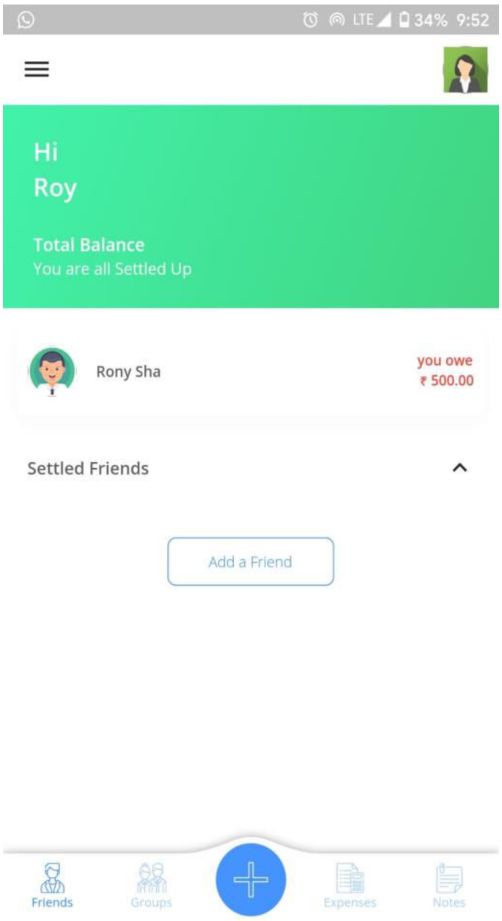
Rony Sha

+918574962580

You owe Rony ₹500.00

Settle Up

Are you sure you want to settle up all non-group expenses with Rony Sha? You cannot restore them later

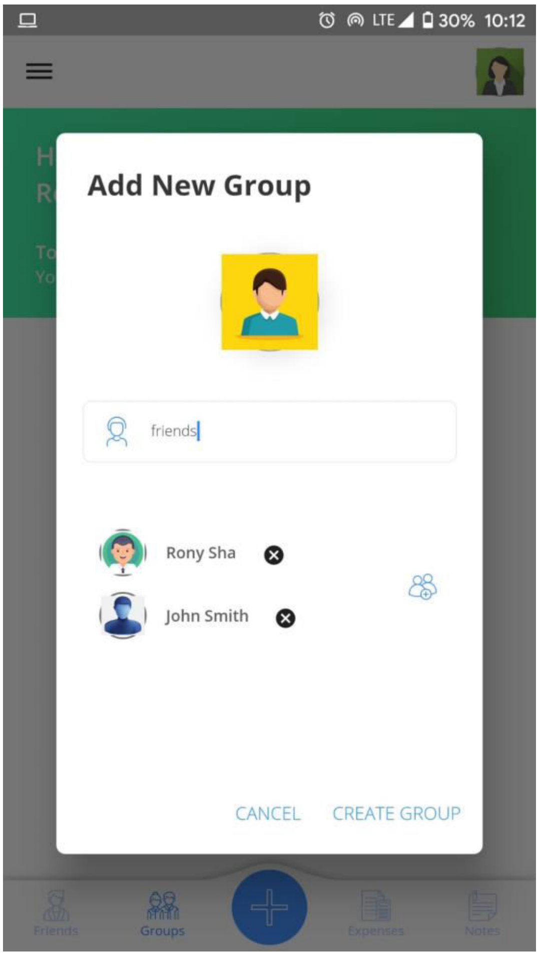


Hi
Roy

Total Balance
You are all Settled Up

Rony Sha	you owe ₹ 500.00

Settled Friends

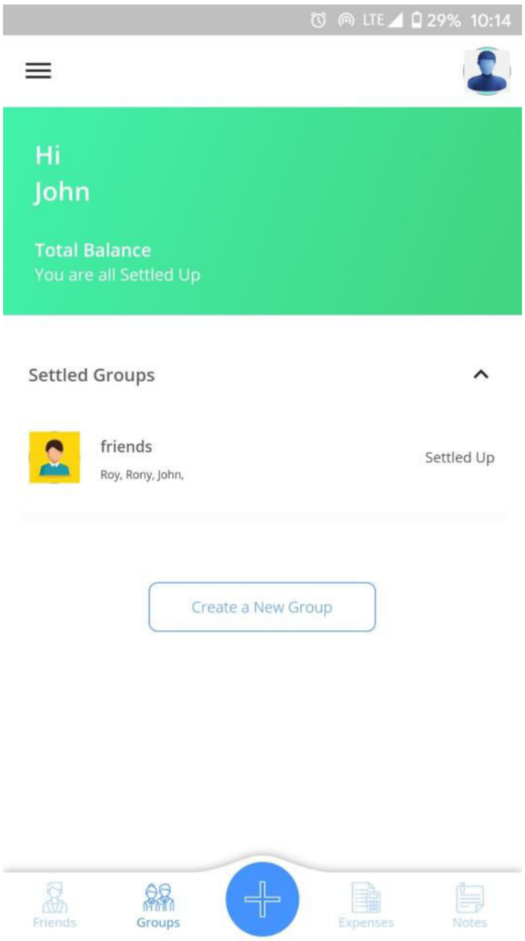

10:12 30% LTE

Add New Group

friends

Rony Sha

John Smith

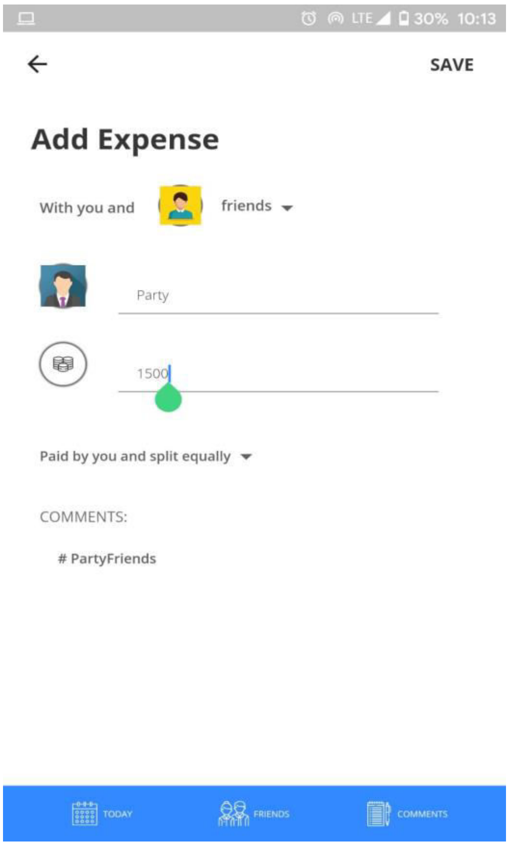

10:14 29% LTE

Hi
John

Total Balance
You are all Settled Up

Settled Groups

friends
Roy, Rony, John,
Settled Up


10:13 30% LTE

Add Expense

With you and friends

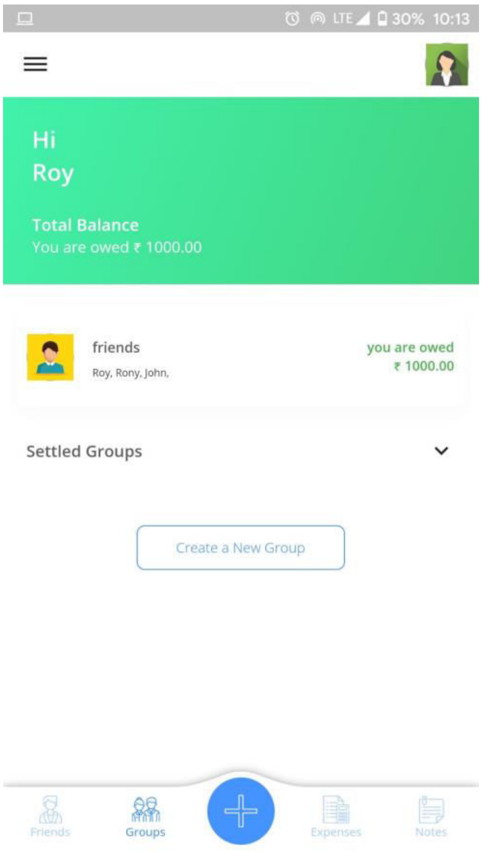
Party

1500

Paid by you and split equally

COMMENTS:

PartyFriends

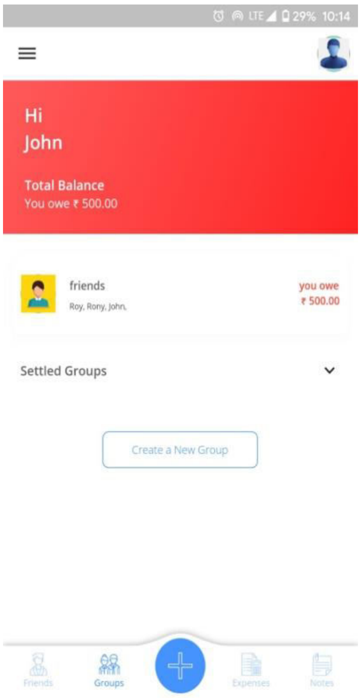

10:13 30% LTE

Hi
Roy

Total Balance
You are owed ₹ 1000.00

friends
Roy, Rony, John,
you are owed
₹ 1000.00

Settled Groups

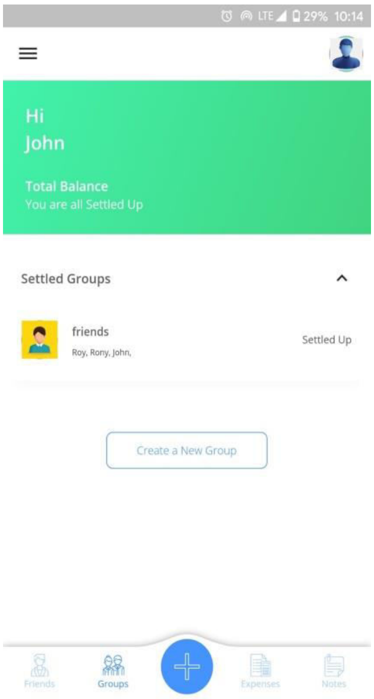


Hi
John

Total Balance
You owe ₹ 500.00

friends
Roy, Rony, John,
you owe
₹ 500.00

Settled Groups

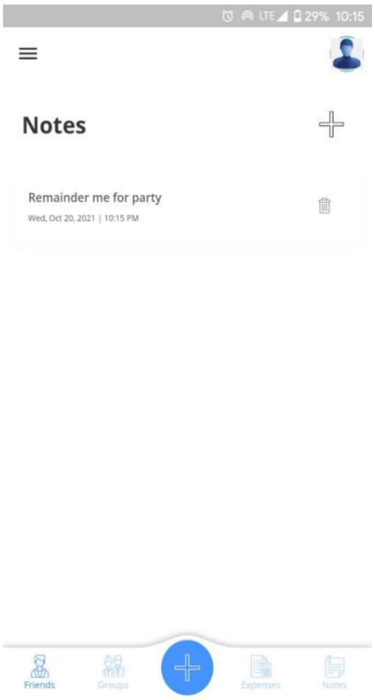


Hi
John

Total Balance
You are all Settled Up

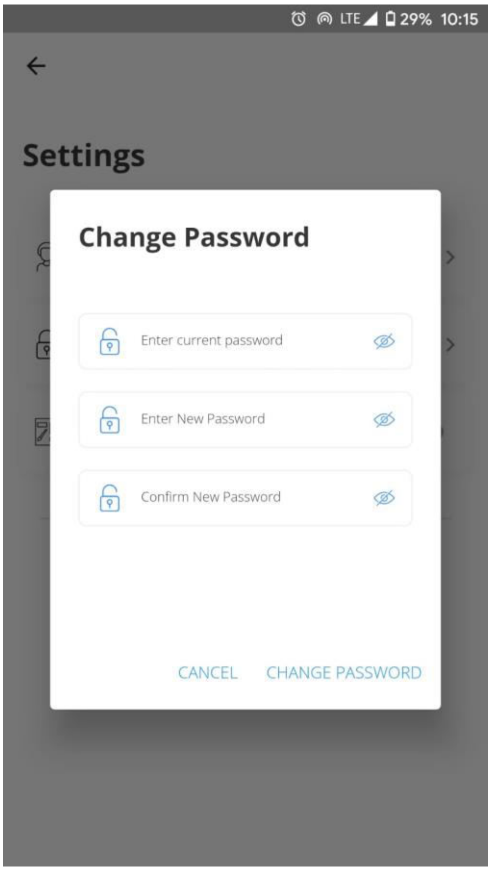
Settled Groups

friends
Roy, Rony, John,
Settled Up



Notes

Remainder me for party
Wed, Oct 20, 2021 | 10:15 PM



Settings

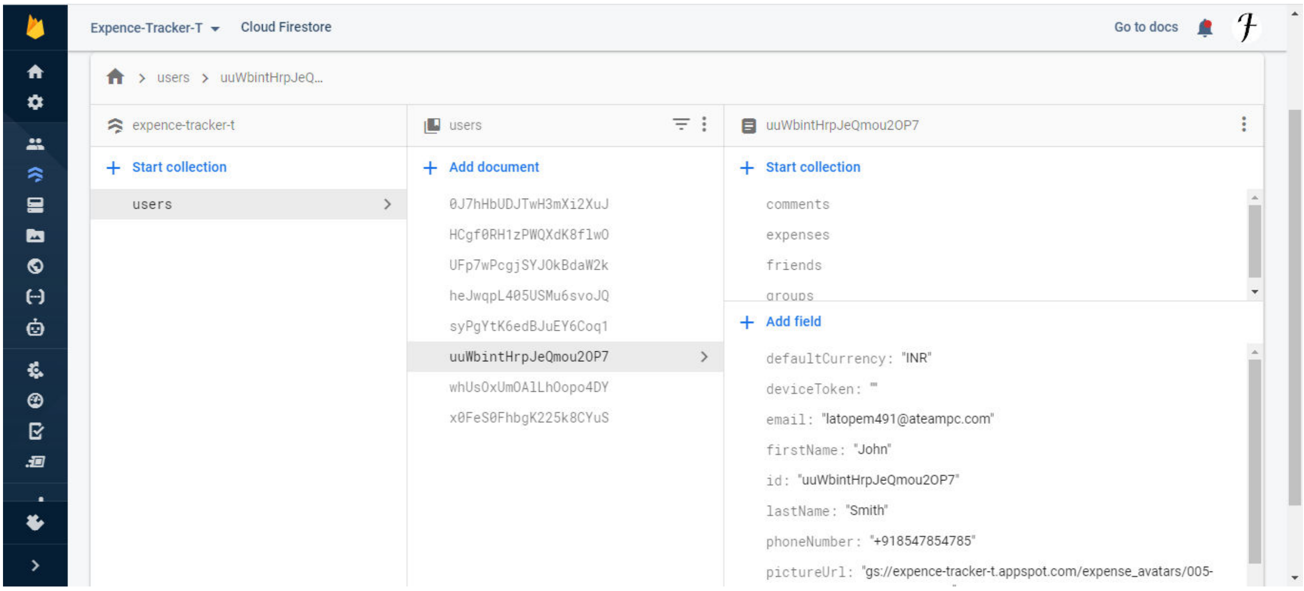
Change Password

Enter current password

Enter New Password

Confirm New Password

Firebase

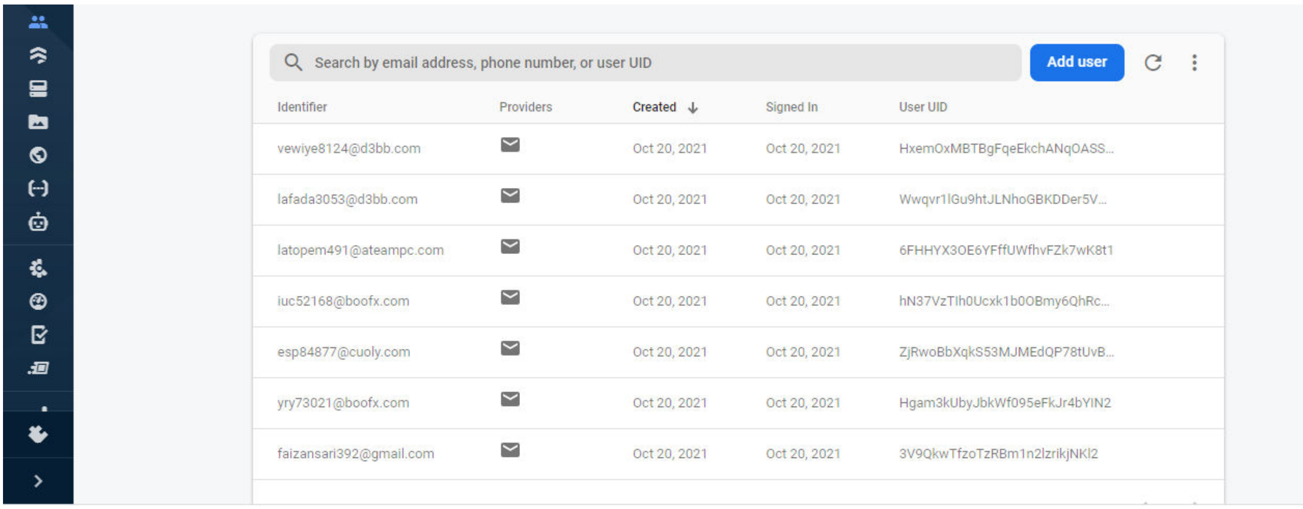


users

```
0J7hHbUDJTwH3mXi2XuJ
HCgf0RH1zPWQXdK8flwO
UFp7wPcgjSYJOkBdaW2k
heJwqpL405USMu6svoJQ
syPgYtK6edBJuEY6Coq1
uuWbintHrpJeQmou2OP7
whUsOxUmOAlLhOopo4DY
x0FeS0FhbgK225k8CYuS
```

```
comments
expenses
friends
groups
```

```
defaultCurrency: "INR"
deviceToken: ""
email: "latopem491@ateampc.com"
firstName: "John"
id: "uuWbintHrpJeQmou2OP7"
lastName: "Smith"
phoneNumber: "+918547854785"
pictureUrl: "gs://expence-tracker-t.appspot.com/expense_avatars/005-
```



Identifier	Providers	Created ↓	Signed In	User UID
vewiye8124@d3bb.com	✉	Oct 20, 2021	Oct 20, 2021	HxemOxMBTBgFqeEkchANqOASS...
lafada3053@d3bb.com	✉	Oct 20, 2021	Oct 20, 2021	Wwqvr1lGu9htJLNhoGBKDDer5V...
latopem491@ateampc.com	✉	Oct 20, 2021	Oct 20, 2021	6FHHYX3OE6YFffUWfhvFZk7wK8t1
iuc52168@boofx.com	✉	Oct 20, 2021	Oct 20, 2021	hN37VzTlh0Ucxk1b0OBmy6QhRc...
esp84877@cuoly.com	✉	Oct 20, 2021	Oct 20, 2021	ZjRwoBbXqkS53MJMEdQP78tUvB...
yry73021@boofx.com	✉	Oct 20, 2021	Oct 20, 2021	Hgam3kUbyJbkWf095eFkJr4bYIN2
faizansari392@gmail.com	✉	Oct 20, 2021	Oct 20, 2021	3V9QkwTfzoTzRBm1n2lzrikjNKl2

7. Conclusion and Future Scope

This app is used to when any splitting situation created in real life when we talk about money, don’t need to remainder who owes and who owed the amount of money,It split equal or unequal amount of money.Notes are there for anything you want to store for remainder.

will add more feature in future

- Expense Categorization
- Push notification
- Track expenses
- Visualize data in tabular for format for quick understand
- Calculator for Gross calculations
- Export data in pdf format

8. References

- https://www.udemy.com/course/learn-flutter-dart-to-build-ios-android-apps/
- https://flutter.dev/
- https://www.youtube.com/watch?v=1ukSR1GRtMU&list=PL4cUxeGkcC9jLYyp2Aoh6hcWuxFDX6PBJ
- https://stackoverflow.com/
- https://dart.dev/guides
- https://www.youtube.com/watch?v=qWL1lGchpRA&list=PLR2qQy0Zxs_UdqAcaipPR3CG1Ly57UlhV