



FACULTY-STUDENT INTERACTION PORTAL(ANDROID APP)

TECHNICAL DOCUMENTATION

Contents

About the project	5
Introduction	5
Definitions, Acronyms and Abbreviations	5
Overview	5
Intended Users	5
System and Software requirements	6
Hardware	6
Software	6
Description of the Project	6
CLASSESS AND OBJECTS used in this Application	7
1. Course Material	7
2. Course Project	7
3. Courses	7
4. Event	8
5. Messages	8
6. NewUser	8
7. Options	9
8. Polls	9
9. Replies	9
10. Thread	10
ACTIVITIES IN THE APPLICATION	11
AddCourseContent Activity	11
Android Widgets and features used in this Activity	11
Buttons used in this Activity	11

Functions used in this Activity _____	12
Description of the Activity _____	12
FlowChart of the Activity _____	12
ComposeMessage Activity _____	13
Android Widgets and Features used in this Activity _____	13
Buttons Used _____	14
Functions used in this Activity _____	14
Description of the Activity _____	14
FlowChart of the Activity _____	15
Course Add Activity _____	17
Android Widgets and Features used in this Activity _____	17
Buttons used in this Activity _____	17
Description of the Activity _____	17
button click function flowchart _____	18
CourseMainPageStudent Activity _____	18
Functions used in this Activity _____	18
Description of the Activity _____	19
CreatePollActivity _____	19
Android Widgets and features used in this Activity _____	19
Buttons Used in this Activity _____	19
Functions used in this Acitvity _____	19
DiscussionThreads Activity _____	20
Description of the Activity _____	20
Home Activity _____	20
Android Widgets and features used in this Activity _____	20
Options Available on Drawer _____	20
Functions Description _____	21
Description of the Activity _____	21

Login Activity _____	21
Android Widgets and features used in this Activity _____	21
Buttons used in this Activity _____	22
Functions Description _____	22
Description of the Activity _____	22
FlowChart of Functions: _____	23
MessageActivity _____	23
Buttons used in this Activity _____	24
Messagebox Activity _____	24
Android Widgets and Features used in this Activity _____	24
Buttons used in this Activity _____	24
Description of the Activity _____	24
PollLayout Activity _____	24
Android Widgets and Features used in this Activity _____	24
Buttons used in this Activity _____	25
Description of the Activity _____	25
Profile Activity _____	25
Android Widgets and Features used in this Activity _____	25
Description of the Activity _____	25
SignUp Activity _____	25
Android Widgets and features used in this Activity _____	26
Buttons Used _____	26
Functions Description _____	26
Description of the Activity _____	26
Splashscreen Activity _____	27
Description _____	27
Shal Custom Activity _____	28

Splash Screen	28
Student Enroll Activity	28
Android Widgets and Features used in this Activity	28
Functions used in this Activity	29
Time Table Fragment	29
Android widgets	29
Description of the Fragment	29
To Do List	30
Android Widgets	30
Description of the Fragment	30

About the project

Introduction

This project was developed as a part of CS243 System Programming Lab to demonstrate the working of a Faculty Student Interaction Portal as an Android Application.

The Project is made in Android Studio with the aid of Java JDK (inbuilt with Android Studio). A Firebase Database is used. The program is divided into sections, each of which contains Activities. For verification of identity, only people with IITG email ids are able to register as a User.

Definitions, Acronyms and Abbreviations

- App/Application and any synonym refers to the application developed in this Assignment and all statements talk in the context of the same.
- The application was developed using *Android Studio* with *JAVA JDK* and keywords and terms relating to it carry the same meaning here.
- Database talks in reference to the Firebase database that is used in the application and words/statements relating to *Firebase Database* carry the same meaning here
- Knowledge of computer related terms like *Login*, *Sign-up* etc. is assumed.

Overview

//Replace with a flowchart of the project (to be done)

Intended Users

Anyone willing to Study, Evaluate, Use, Modify, Deploy or work with the application in any way is free to use this document.

System and Software requirements

The use and development of the application needs some requirements to be fulfilled.

Hardware

- To run the built application:

Any android device with API 21 and a resolution of 720p or higher, storage of 1GB or higher, RAM 1GB or higher.

- For development:

Any device supporting android Studio App development Android Studio 3.3.2 or higher.

Software

- To run the built application:

Any Android phone with Android API 21 or higher.

- For development:

Any device supporting Android Studio App development with Android Studio 3.3.2 or higher.

Description of the Project

//Replace with a flow chart of the Application

CLASSESS AND OBJECTS used in this Application

1. COURSE MATERIAL

- String Title;
- String URL;
- String FileName;
- java.util.Date Date;

2. COURSE PROJECT

- String Title;
- String URL;
- String FileName;
- String Deadline;
- String Description;

3. COURSES

- String CourseID;
- String CourseKey;
- String Description;
- Date DateOfCreation;
- ArrayList<Event> events;
- String Fullname;
- String MarksDistribution;
- String Professor;
- String StudentList;
- String Syllabus;
- String TimeSlots;

- ArrayList<Thread> threads;
- ArrayList<CourseMaterial> CourseMaterial;

4. EVENT

- Date DateOfCreation;
- String Title;
- String Description;
- String DateOfEvent;
- String TimeOfEvent;
- String Venue;
- String Type;

5. MESSAGES

- String sender;
- String receiver;
- boolean read;
- String subject;
- String date;
- String time;
- String body;
- String uniquid;

6. NEWUSER

- boolean verified;
- String fullname;
- String username;
- String usertype;
- String rollnumber;
- String email;

- String occupation;
- String department;
- String year;
- String password;
- ArrayList<Messages> messages;
- ArrayList<String> courses;

7. OPTIONS

- String optiontext;
- int votes=0;
- int totalvotes=0;
- Integer usernum;
- ArrayList<String> userslist;

8. POLLS

- String creator;
- String creatertype;
- String question;
- String uniqueid;
- Integer usernum;
- ArrayList<String> options;
- Integer[] optionvotes;
- ArrayList<Options> options;
- ArrayList<String> users;
- Integer totalvotes=0;
- boolean isactive = true;

9. REPLIES

- Date DateOfCreation;

- Date LastModified;
- String ReplyContent;
- String Username;
- String Usertype;

10. THREAD

- boolean ThreadClosed;
- String Username;
- String Usertype;
- String ThreadContent;
- String Title;
- Date DateOfCreation;
- Date LastModified;

ACTIVITIES IN THE APPLICATION

AddCourseContent Activity

Java file: AddCourseContent.java

Xml file: activity_add_course_content.xml

ANDROID WIDGETS AND FEATURES USED IN THIS ACTIVITY

Types	Names	Description
TextView	CourseIDTextView	To display prompt CourseID
	CourseNameTextView	To display prompt course title
	KeyTextView	To display prompt key for the course
	DescriptionTextView	To display prompt course Description
	SyllabusTextView	To display prompt course syllabus
	MarksTextView	To display prompt course MarksDistribution
	TimeslotsTextview	A clickable TextView to choose TimeSlots
	textView5	To display the chosen time slots
DatabaseReference	databaseReference	To get reference to the database
Spinner	spinner	To use as a combo box for entries in TimeSlots
Toolbar	toolbar	Toolbar of the Activity
EditTexts	textView	Entry for Course Name
	textView1	Entry for Description
	textView2	Entry for Syllabus
	textView3	Entry for Course key
	textView4	Entry for Marks entry
Intent	intent	To start a new Activity

BUTTONS USED IN THIS ACTIVITY

Name	Description	New Activity on Click if created
------	-------------	----------------------------------

btn	It allows to add the TimeSlots	None
CourseAddButton	Adds a course if the entered data is valid	On valid entries CourseMainPageStudent.class

FUNCTIONS USED IN THIS ACTIVITY

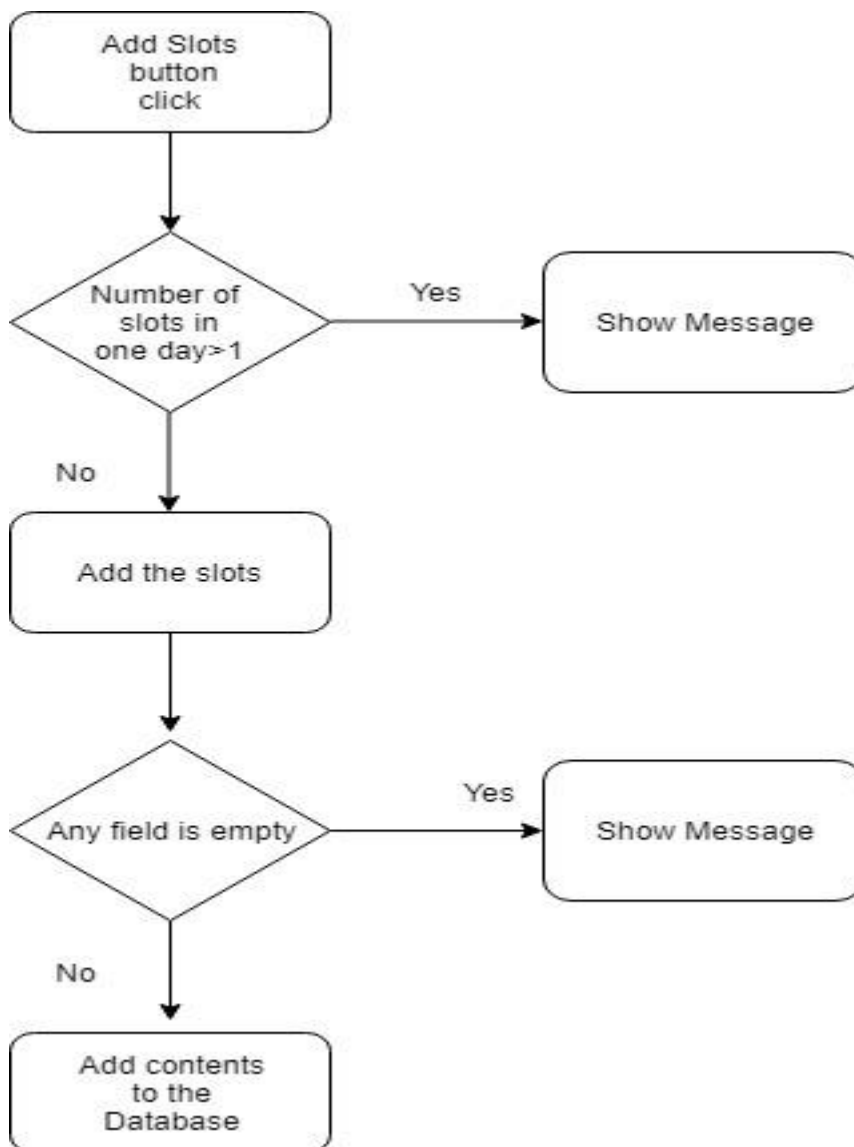
Name	Arguments	Description	New Activity if Created	Return value
getCount	None		None	Integer constant 3

DESCRIPTION OF THE ACTIVITY

After successful verification of courseID, the user will need to put the details of the course to add it to his/her list of courses. Database reference `databaseReference_users_add_course.child("users").child(UserInfo.username).child("courses").setValue(UserInfo.courses)` is used where `UserInfo` is a class mentioned above.

FLOWCHART OF THE ACTIVITY

Add time slots



ComposeMessage Activity

Java file: ComposeMessage.java

Xml file: activity_compose_message.xml

ANDROID WIDGETS AND FEATURES USED IN THIS ACTIVITY

Types	Names	Description
EditText	reciever	To enter the reciever
	subject	To enter the subject of the message
	body	To enter the body of the message

FirestoreDatabase	database	To get Instance to the database
DatabaseReference	databaseReference	To get Reference to the database

BUTTONS USED

Name	Description	New Activity on Click if created
(FloatingActionButton) sendbtn	calls sendMessagefunc on click	None

FUNCTIONS USED IN THIS ACTIVITY

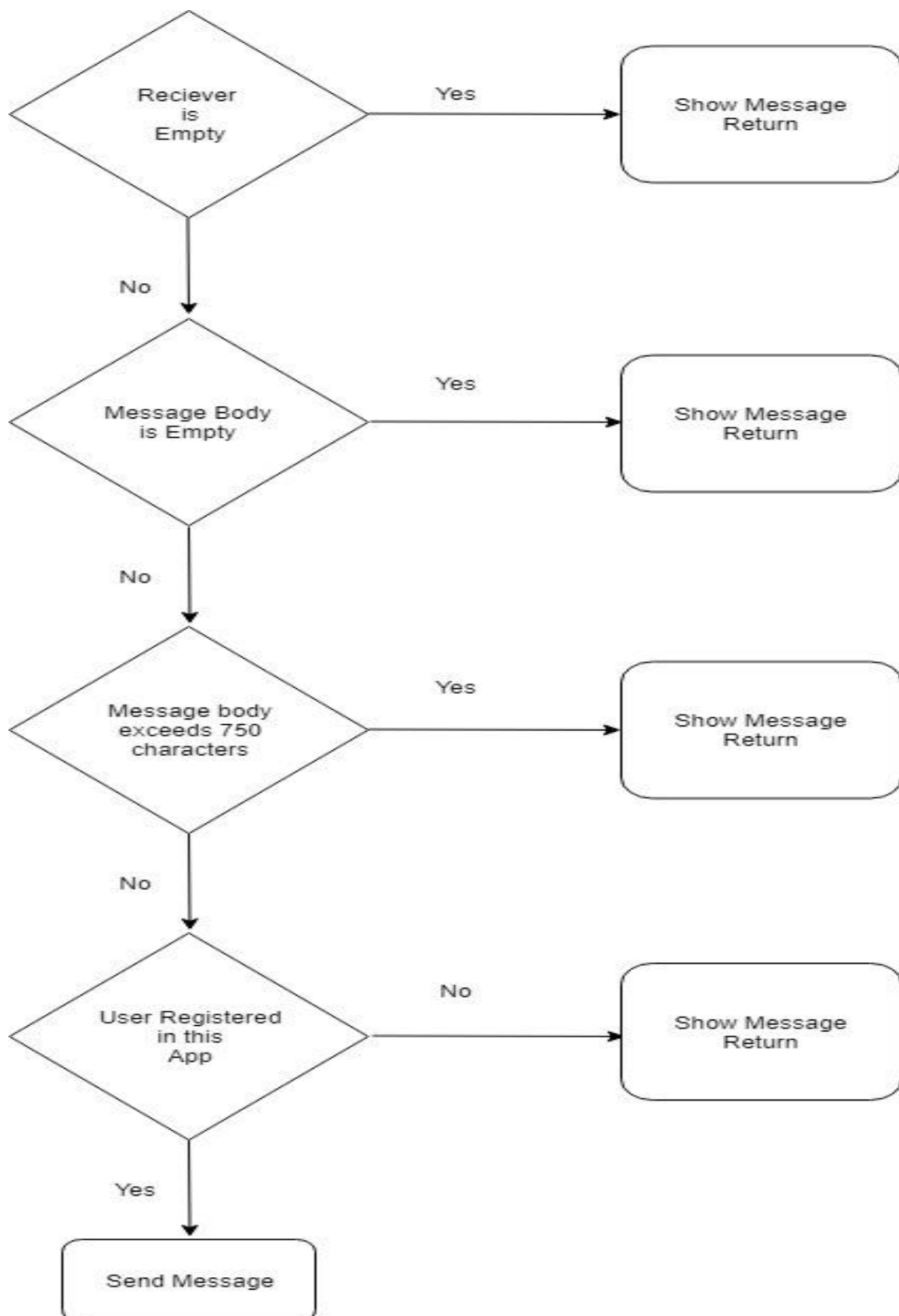
Name	Arguments	Description	New Activity if Created	Return value
sendMessageFunc	None	Sends the Message on valid entries	None	NULL

DESCRIPTION OF THE ACTIVITY

It can be used to send message to other users.

It sets the Title to “Compose Message”, “Reply”, or “Forward”. On clicking send, it calls sendMessageFunc, which takes reference to the child Reciever name in the Database with child Users, which if exists send the message if successful.

FLOWCHART OF THE ACTIVITY



Course Add Activity

Java file: CourseAdd.java

Xml file: activity_course_add.xml

ANDROID WIDGETS AND FEATURES USED IN THIS ACTIVITY

Types	Names	Description
Database Reference	Database Reference	To get reference to the Database
Edit Text	CourseID	To take input for the courseID
Toolbar	toolbar	Toolbar of the Activity

BUTTONS USED IN THIS ACTIVITY

Name	Description	New Activity on Click if created
button	It will attempt to create a new course with the mentioned courseID onclick	If a new course is successfully created AddCourseContent

DESCRIPTION OF THE ACTIVITY

To Add a course, user needs to type the CourseID which is not already present, and not null. It checks if the CourseID entered is Valid or not and if Valid leads to a AddCourseContent activity.

```
databaseReference = FirebaseDatabase.getInstance().getReference()
```

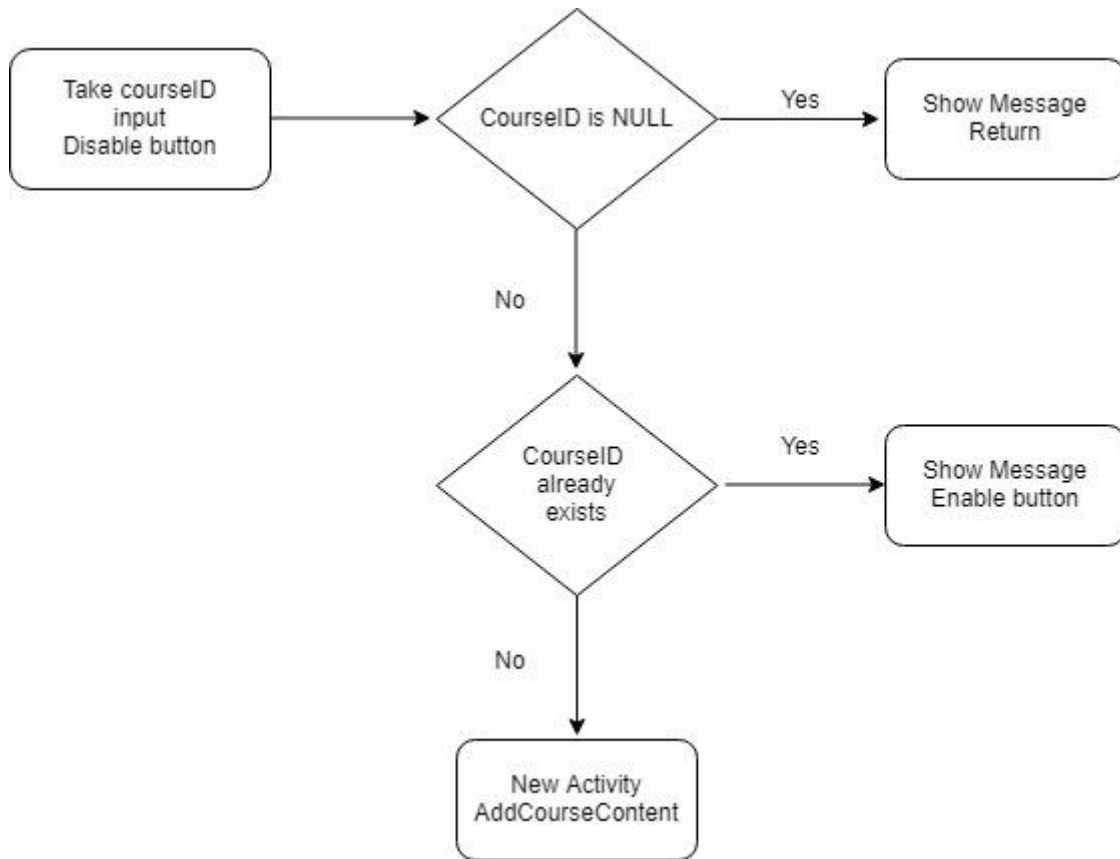
gets reference to the child Courses of the database

```
databaseReference.child("Courses").child(course_id).setValue("")
```

adds a new Course with key CourseId in the Database on Courses child.

BUTTON CLICK FUNCTION FLOWCHART

button



CourseMainPageStudent Activity

Java file: CourseMainPageStudent.java

Xml file: activity_main_page_student.xml

FUNCTIONS USED IN THIS ACTIVITY

Name	Arguments	Description	New Activity if Created	Return value
getItem	position	To place and switch between the fragments in the MainPage	None	Fragments
getCount	None	A constant to help in fragment display	None	Integer Constant 3

DESCRIPTION OF THE ACTIVITY

This Activity loads the Main Page of a profile in the Application which has a fragment. The adapter decides the fragment to be displayed. The left is a toolbar and it can be used to navigate through different properties in the Application.

Case1: If user is a student return courseinformation fragment student, else return CourseMainPageProf fragment.

Case2: Returns CourseDiscussionFragment

Case3: Returns pollActiivity

CreatePollActivity

Java file: CreatePollActivity.java

Xml file: activity_create_poll.xml

ANDROID WIDGETS AND FEATURES USED IN THIS ACTIVITY

Types	Names	Description
EditText	et	To store poll value
	question	quesiton
FirestoreDataabse	database	Database
DatabaseReference	databaseReference	Database

BUTTONS USED IN THIS ACTIVITY

Name	Description	New Activity on Click if created
addoptionbtn: sendbtn	This is to add option	Null
submitbtn	After creating poll conform buton	Null

FUNCTIONS USED IN THIS ACITVITY

Name	Arguments	Description	New Activity if Created	Return value
Add Line	None	This features set a new id to added layout	None	NULL
Submit Function	None	Get Reference to the Database with child "Users"	None	NULL

DiscussionThreads Activity

Java file: DiscussionThreads.java

Xml file: activity_discussion_threads.xml

DESCRIPTION OF THE ACTIVITY

Using databaseReference = FirebaseDatabase.getInstance().child(course).child("Threads"), the details of the discussion forum is loaded.

Home Activity

Java file: home.java

Xml file: activity_home.xml

ANDROID WIDGETS AND FEATURES USED IN THIS ACTIVITY

Types	Names	Description
Databasereference	databaseReference	Databse refernce
DrawerLayout	drawer	To implement a drawer
ActionBarDrawerToggle	toggle	Toggle
SectionsPagerAdapter	mSectionsPagerAdapter	Add pages list to this adapter
ViewPager	mViewPager	In this tool bar pages are added
NavigationView	navigationView	To add nav bar to left side

OPTIONS AVAILABLE ON DRAWER

Name	Description	New Activity, if Opened
nav_profile	used to check personal profile	ProfileActivity
courses	used to check registered courses	CourseSearchTabbed
exam_timetable	used to check exam timetable	Is to Copy
nav_FAQ	used to see the FAQ	Specific questions for user To guide

nav_logout	used to logout as the current user, UserInfo class is cleared of its contents	LoginActivity
nav_message	used to view messages	MessageActivity

FUNCTIONS DESCRIPTION

Name	Arguments	Description	New Activity if Created	Return value
onCreateOptionsMenu	Menu	Adds items to the action bar if present	None	Boolean true
getCount	None		None	Integer constant 3

DESCRIPTION OF THE ACTIVITY

Using reference to the database this one uses the information to create a new instance of class UserInfo and thus create the application details.

Login Activity

Java file: LoginActivity.java

Xml file: login_screen.xml

ANDROID WIDGETS AND FEATURES USED IN THIS ACTIVITY

Types	Names	Description
EditText	etusername	To take input of the username
	etpassword	To take input of the password
FirebaseAuth	firebaseAuth	To get Authentication details of the user from firebase
OauthProvider.Builder	provider	To get data from Microsoft Outlook
FirebaseDatabase	database	To get Instance of the Databsae
Handler	handler	To handle between different activities
Runnable	runnable	To enable btn_login button
	runnable2	To enable btn_customlogin

BUTTONS USED IN THIS ACTIVITY

Name	Description	New Activity if created
btn_login	calls loginfunction	Depends on function
btn_signup	allows the user to go the signup activity	SignupActivity
btn_customlogin	calls customloginfunction	Depends on function

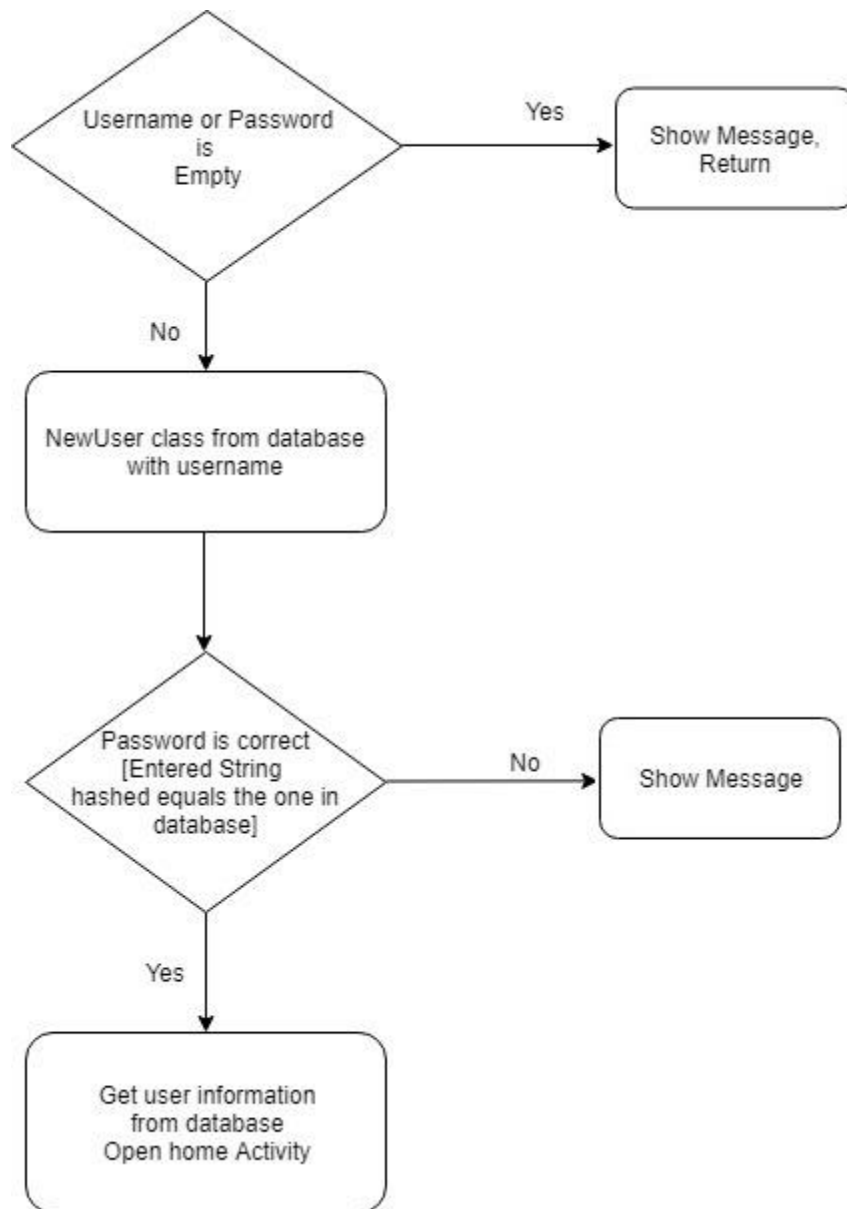
FUNCTIONS DESCRIPTION

Name	Arguments	Description	Return value
getvaluesfromoutlook	FirebaseUser loginuser, AuthResult authResult	It is used to read available data from outlook	None
loginfunction	None	It is used to validate login	None
customloginfunction	None	It is used to validate customlogin	None

DESCRIPTION OF THE ACTIVITY

This is the Launcher Activity of the Application, the one that starts with the start of the application.

FLOWCHART OF FUNCTIONS:
customloginfunction



MessageActivity

Java file: MessageActivity.java

Xml file: activity_message.xml

BUTTONS USED IN THIS ACTIVITY

Name	Description	New Activity on Click if created
composebtn (floating action button)	This allows a new message to be composed in the App.	ComposeMessage

Messagebox Activity

Java file: messageboxActivity.java

Xml file: activity_messagebox.xml

ANDROID WIDGETS AND FEATURES USED IN THIS ACTIVITY

Types	Names	Description
TextView	subjecttv	Description of the type of sender, receiver,date, etc. of the mail.
	sendertv	
	recievertv	
	datetv	
	bodytv	

BUTTONS USED IN THIS ACTIVITY

Name	Description	New Activity on Click if created
deletebtn	To delete a message	None
replybtn	To reply to a message	ComposeMessage
forwardbtn	To forward a message	ComposeMessage

DESCRIPTION OF THE ACTIVITY

After successful creation of the activity, on clicking the delete button the dataref gets reference to UserInfo.username.child("messages").child(uniqueid) and value is removed using removevalue.

PollLayout Activity

Java file: PollLayoutActivity.java

Xml file: layoutpoll.xml

ANDROID WIDGETS AND FEATURES USED IN THIS ACTIVITY

Types	Names	Description
TextView	question	To display the questions

DatabaseReference	databaseReference	To get reference to the poll details database where username is stored if voted
--------------------------	-------------------	---

BUTTONS USED IN THIS ACTIVITY

Name	Description	New Activity on Click if created
editchoicebtn	To change the choices of the poll	None
closepollbtn	To close the poll	None

DESCRIPTION OF THE ACTIVITY

Using `dataref.child(currentcourseid).child("Polls").child(clickedpoll.uniqueid)` is used to get reference to the database course polls. For an user its checked if he/she has already chosen an option else a new entry in the database is generated.

Profile Activity

Java file: ProfileActivity.java

Xml file: activity_profile.xml

ANDROID WIDGETS AND FEATURES USED IN THIS ACTIVITY

Types	Names	Description
TextView	fullname	To display the personal information in the Acitivity.
	username	
	email	
	occupation	
	rollnumber	
	year	
	department	
ImageView	profilepic	To put image(profile pic) in the activity

DESCRIPTION OF THE ACTIVITY

UserInfo class has all the required data already stored. So if the user is a professor his details is shown and if it's a student then details along with roll no and year is shown.

SignUp Activity

Java file: SignupActivity.java

Xml file: signup_layout.xml

ANDROID WIDGETS AND FEATURES USED IN THIS ACTIVITY

Types	Names	Description
EditText	name	To get name entry
	email	To get email entry
	rollnumber	To get roll number entry
	occupation	To get occupation entry
	year	To get the year entry
	password	To get password entry
	pass2	To get confirmation password entry
Spinner	department	
FirestoreDatabase	firebaseDatabase	To get instance of the Database
DatabaseReference	dataref	To get location of the Database

BUTTONS USED

Name	Description	New Activity if created
signupbtn	calls signupfunction on try part of try catch block	Depends on function

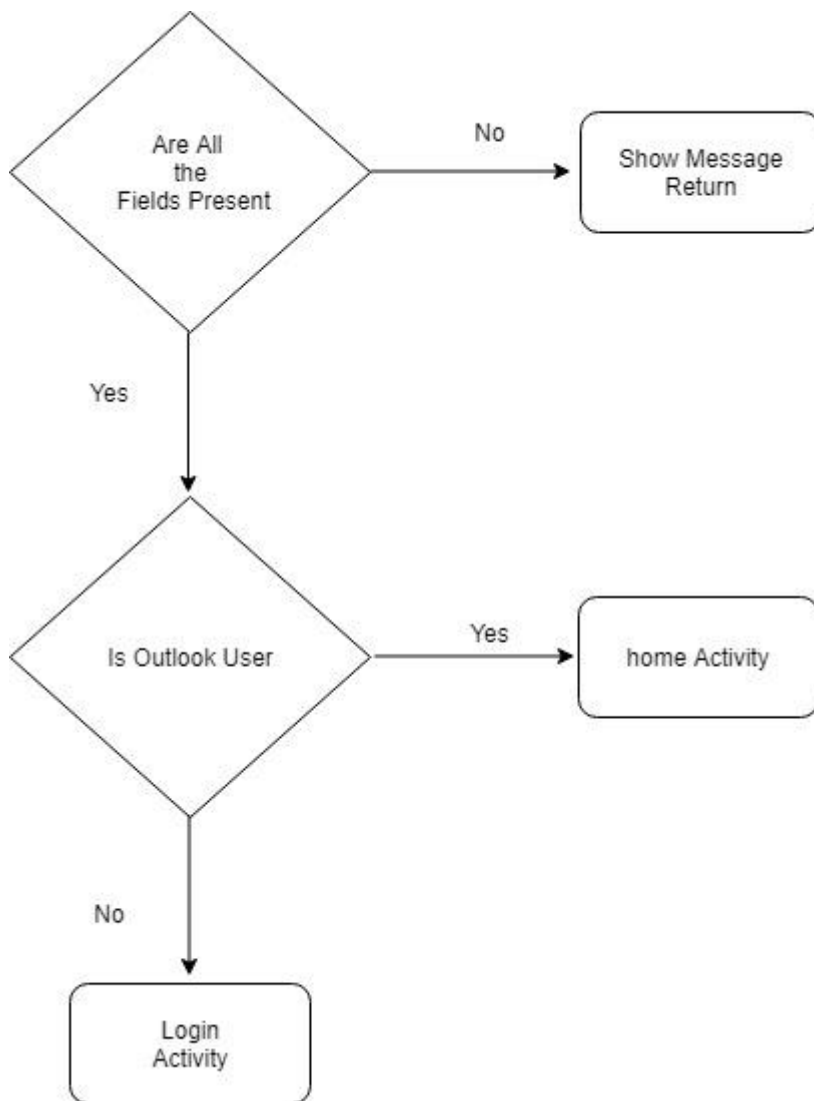
FUNCTIONS DESCRIPTION

Name	Arguments	Description	New Activity if Created	Return value
signupfunction	None	Allows someone to sign up as a new user	MainActivity or LoginActivity depending upon the details entered	None

DESCRIPTION OF THE ACTIVITY

dataRef is used to get reference to the database with child "Users" to write the details of the users. If attempted signup via outlook it responds and checks if any field is empty. Depending upon the users entry of the occupation if "Prof" then a professor is created, and if any other department is chosen, it is created as a student.

signupfunction



Splashscreen Activity

Java file: Splashscreen.java

DESCRIPTION

This is used for the delay caused due to signing in as a new User in the database.

Shal Custom Activity

This Activity is Basically used for hashing of password so that even administrator cant see it



Functions used in this Activity

Name	Arguments	Description	Return value
SHAI	String	It takes string and hash it	String (hashed one)

Splash Screen

This Activity is basically for showing loading screen while it system is accessing database

It takes time to Access it

Student Enroll Activity

This activity mainly check whether the enroll key Provided by student is correct or

Not

Database

```
pk=db.getReference().child("Courses").child(name.getText().toString()).child("coursekey")
```

This reference given enroll key value and check it with key provided by student

ANDROID WIDGETS AND FEATURES USED IN THIS ACTIVITY

Types	Names	Description
TextView	Name Desc Time Title	Everything used is textview only

	Syl Prof	
--	-------------	--

FUNCTIONS USED IN THIS ACTIVITY

Name	Arguments	Description	New Activity if Created	Return value
openDialog	Null	Open dialog box for providing key	Here new alert box is created	void
ApplyText	Key (provided by string)	This will check key	Null	void

Time Table Fragment

ANDROID WIDGETS

Types	Names	Description
Array<String>	al,al_time,al_title	To store list of timetable
List view	r	To store array list in list view
Button	choose	Open calendar date picker

DESCRIPTION OF THE FRAGMENT

Choose button to open date time picker

Calendar c= Calendar.getInstance();

This will open date time picker

c.get will give the values picked in date time picker

After selecting date time picker we get date and time in required format;

And we will add that string to array list to show

We used simple_list_item_1 for layout for list ie layout has textbox to show user name and other stuff

db.getRefrence().child("Cources").child(value).child("timeslots") give the acess for database for time slots and we will call filltable() to fill time slots in database

To Do List

ANDROID WIDGETS

Types	Names	Description
Adapter	Todo	To link array and list view
Dialog	Dialogview,dialogview2	Store to be written content in todo list
Array<list>	Todos,ids	To store content of todo

DESCRIPTION OF THE FRAGMENT

```
FirestoreDatabase().getInstance().getReference().child("users").child(username).child("todo");
```

This will give database of access list

NewThread button is for adding new thread

New Thread. Set On Click Listener will open a dialog box ie alert box this uses add_todo_dialogview for layout

on long click its gona open new alertbox for delete or update list ("getItem(psotion)");

delete list is happening through ="databseRefrence.removeValue()"

