CAMPUS ISSUES
(PROJECT NVCTI)

DOCUMENTATION AND SCHEMATICS OF APPLICATION - PUPPALA PRANAY(19JE0661)

DESCRIPTION OF APPLICATION

- Campus Issues is an app which provides a platform where the students of campus can raise their issues regarding many different topics and at the same time management authorities can check the issues raised by students and take an action against it.

- This app is built with "Flutter" (UI toolkit by Google) integrated with Google's Firebase services for "Authentication" and "Real time database" .

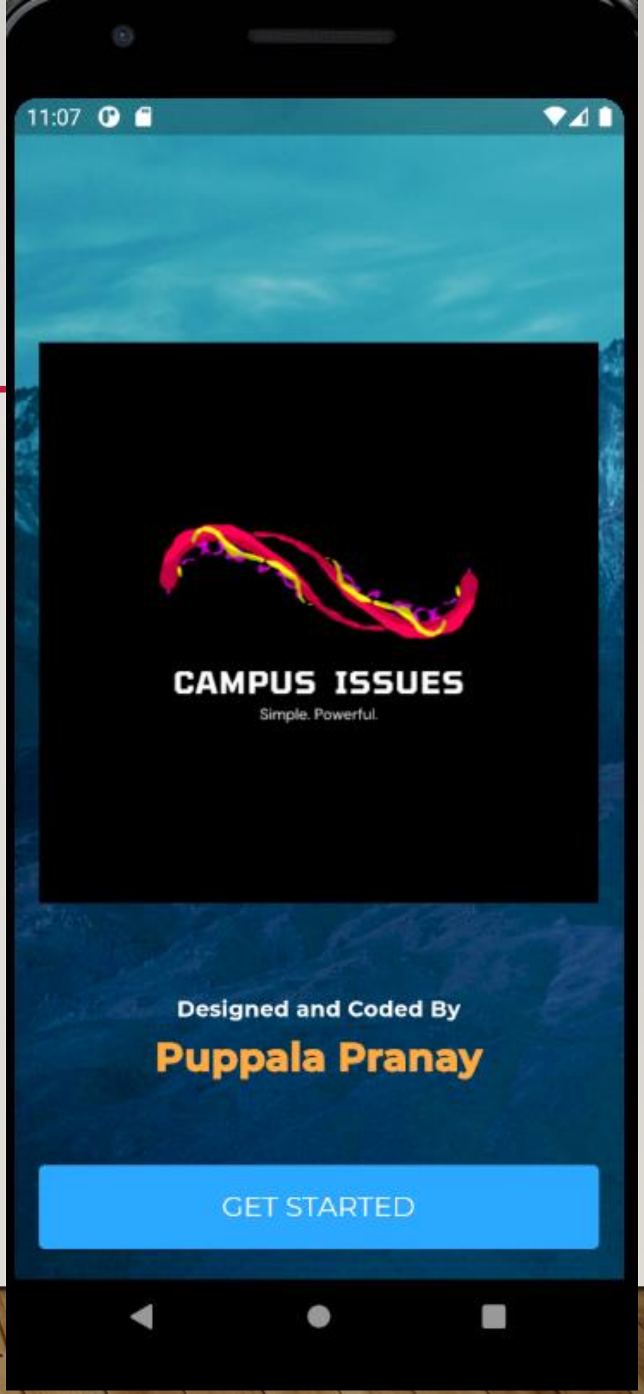
MAIN FILE SYSTEM OF APPLICATION

Screens	Functions	Widgets	Constants
onboarding. dart	auth_functions.dart	drawer.dart	Theme.dart
signIn.dart	database.dart	drawer-tile.dart	
register.dart		Input.dart	
registercomplain.dart		navbar.dart	
Yourcomplaints.dart		card-horizontal.dart	
Checkcomplaints.dart			

SCREENS

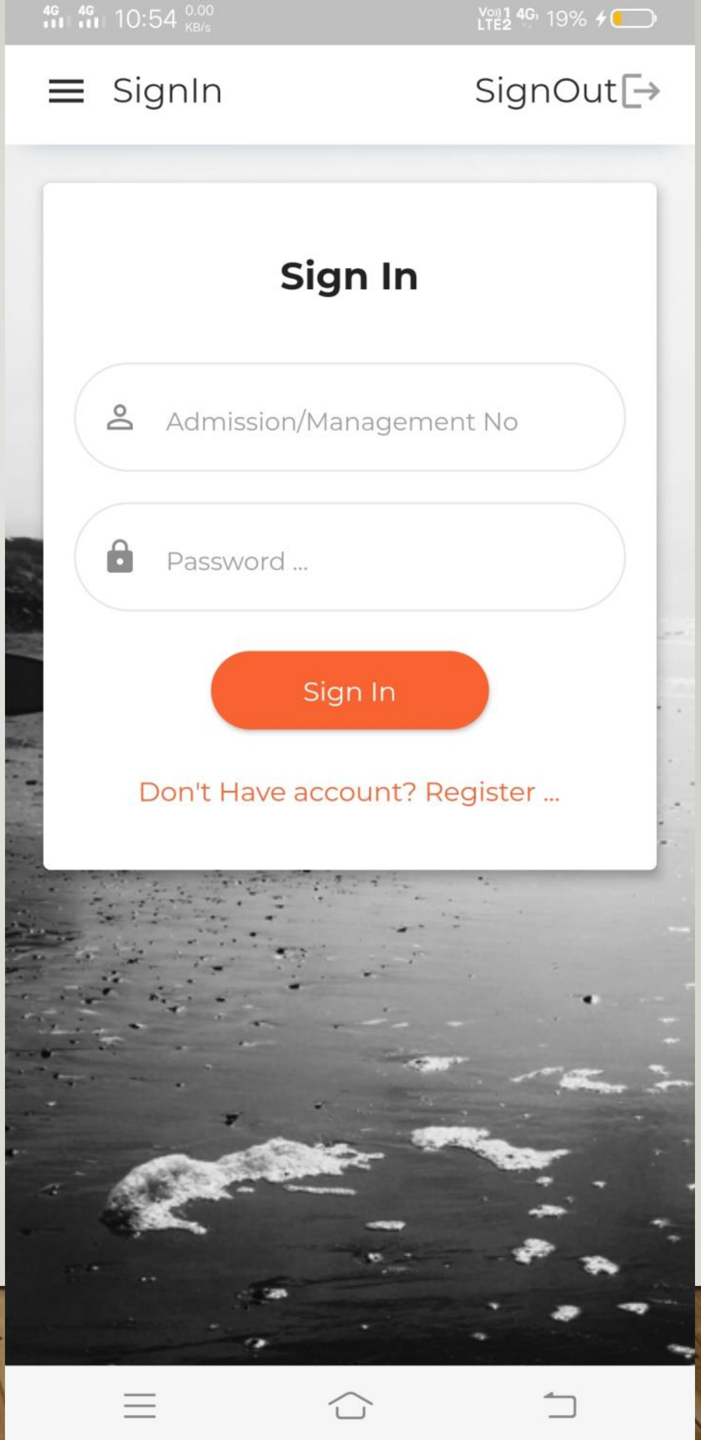
1) Onboarding Screen

This is the opening screen of the app which
Displays the Logo of App and the credit for developer.
Followed by a button with text "Get Started" which on
clicked takes the user to next state.



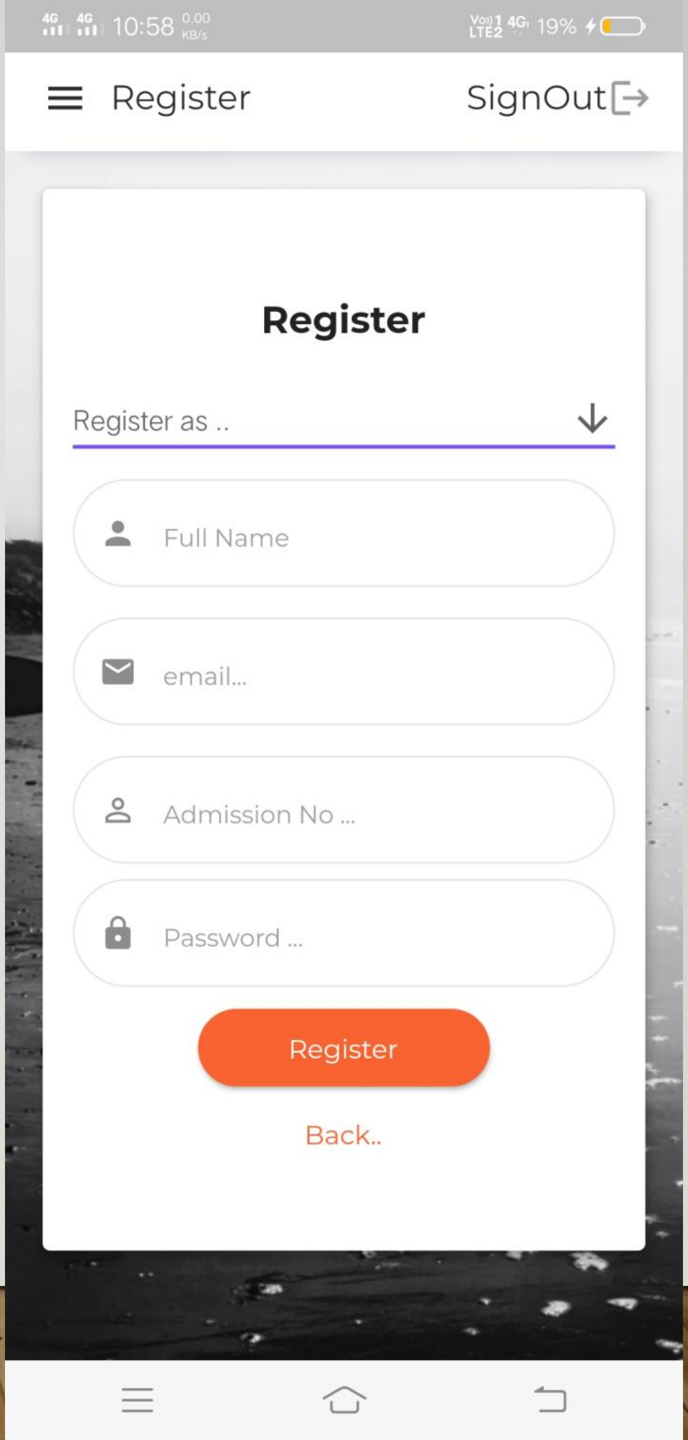
2) SIGN IN PAGE

- When user enters the app , if he is not logged in , he/she will be redirected to this sign in page.
- Here , the user is required to enter admission no (if student) or management number (if management authority) along with the password that is set by the user itself while registration .
- If User has not registered to app . He can use the text button below to navigate to the Registration page.
- On pressing the Sign In button , If authorized by firebase service , He will be directed to Home Page , else error is displayed and stays on the same page for the user to recheck.



3) REGISTER PAGE

- When User is not Signed in, He can register In app by using the register button in sign In page or in side-drawer.
- Here , The user can Register as Student or Management by entering the required fields like Full name, email, Admission/Management number and Password.
- Upon pressing register, the app will register him as a user in firestore (database in firebase) and creates a authentication record.
- When the registration is successful , he will be directed to home page .



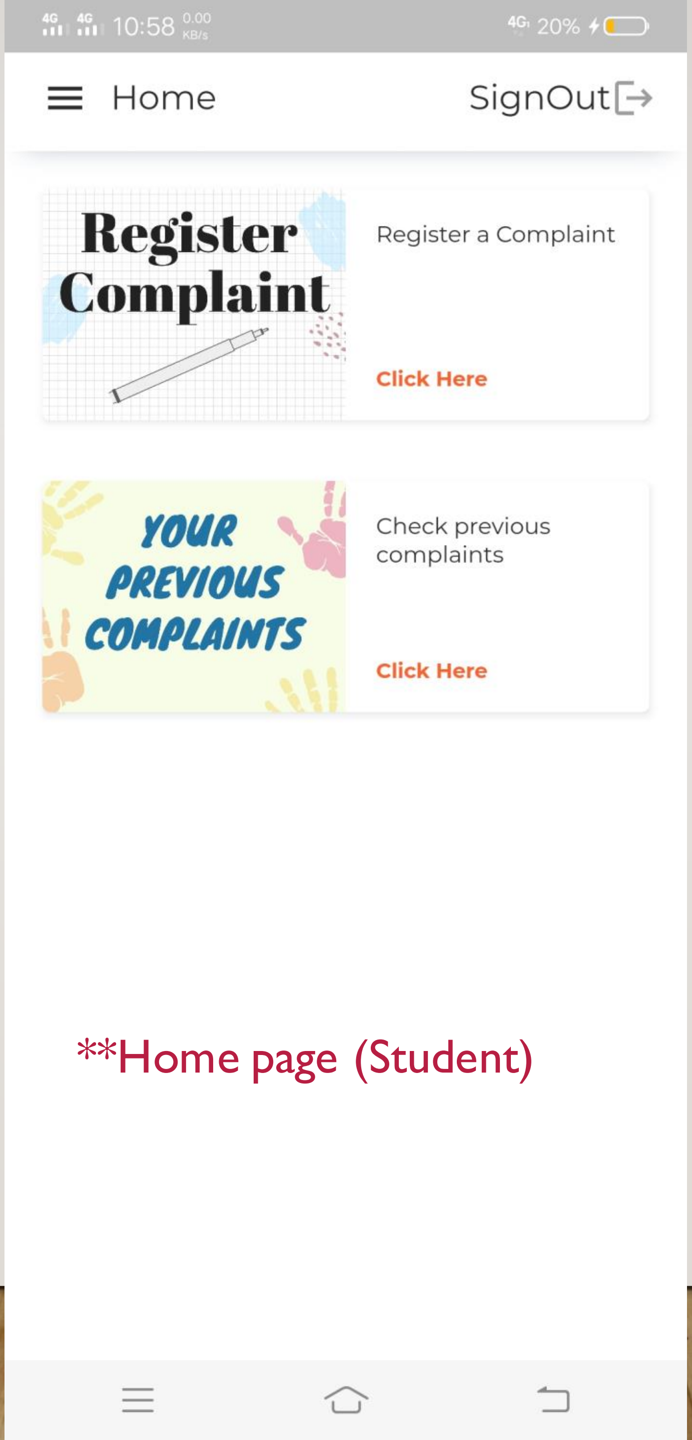
4) Home Page (Student)

- When User is Signed in, he is directed to home page .

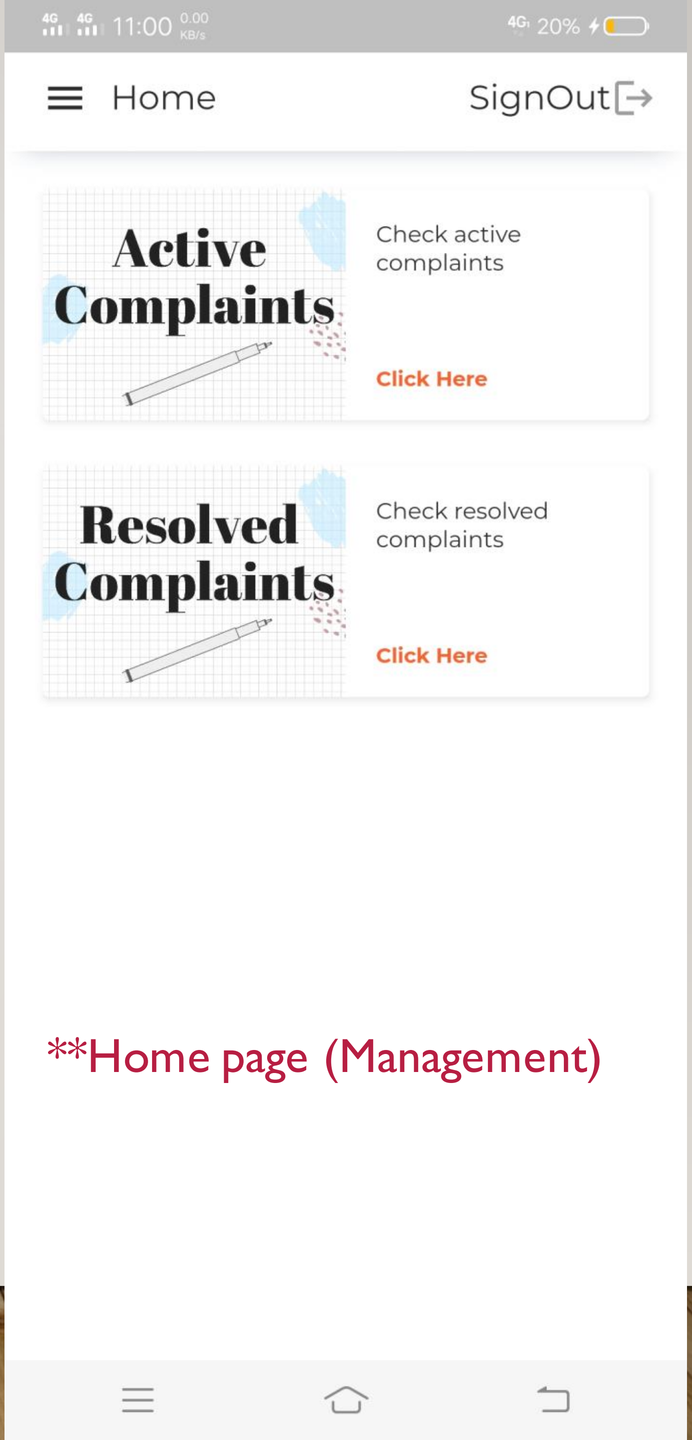
- The ***home page*** contains the features accessible by the user based on his role (Student/Management) .

- For ***Student*** ,The home page contains features to *Register complaint* and to *check previous complaints* .

- For ***Management*** ,The home page contains features to Check *active complaints* and to check *resolved complaints*



**Home page (Student)



**Home page (Management)

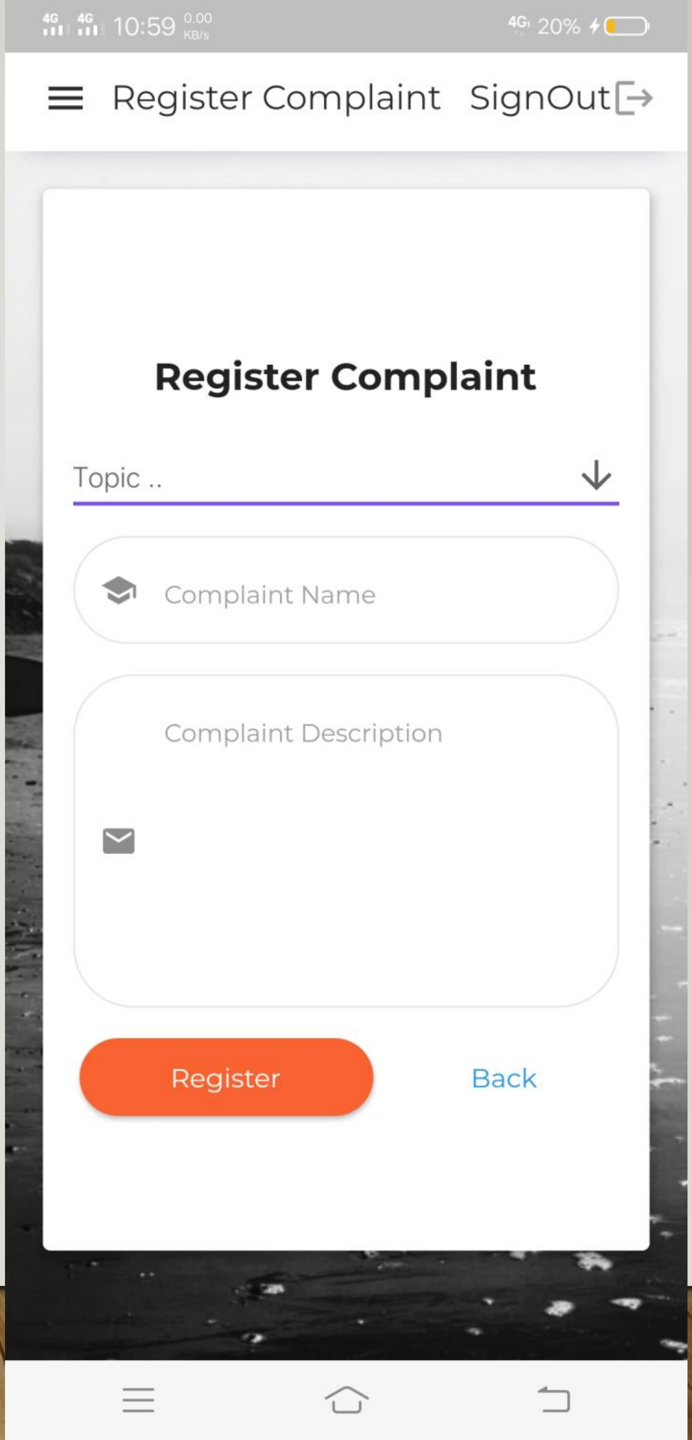
5) REGISTER COMPLAINT PAGE (STUDENT)

- If , the user is Student Then he can access the feature to Register complaint from home page or side drawer

- Here , The registered Student can select the topics of complaint from drop down menu .
  The topics include :
  1) Hostel
  2) Academics
  3)Security
  4)Administration
  5)Co-Curriculum
  6)Miscellaneous

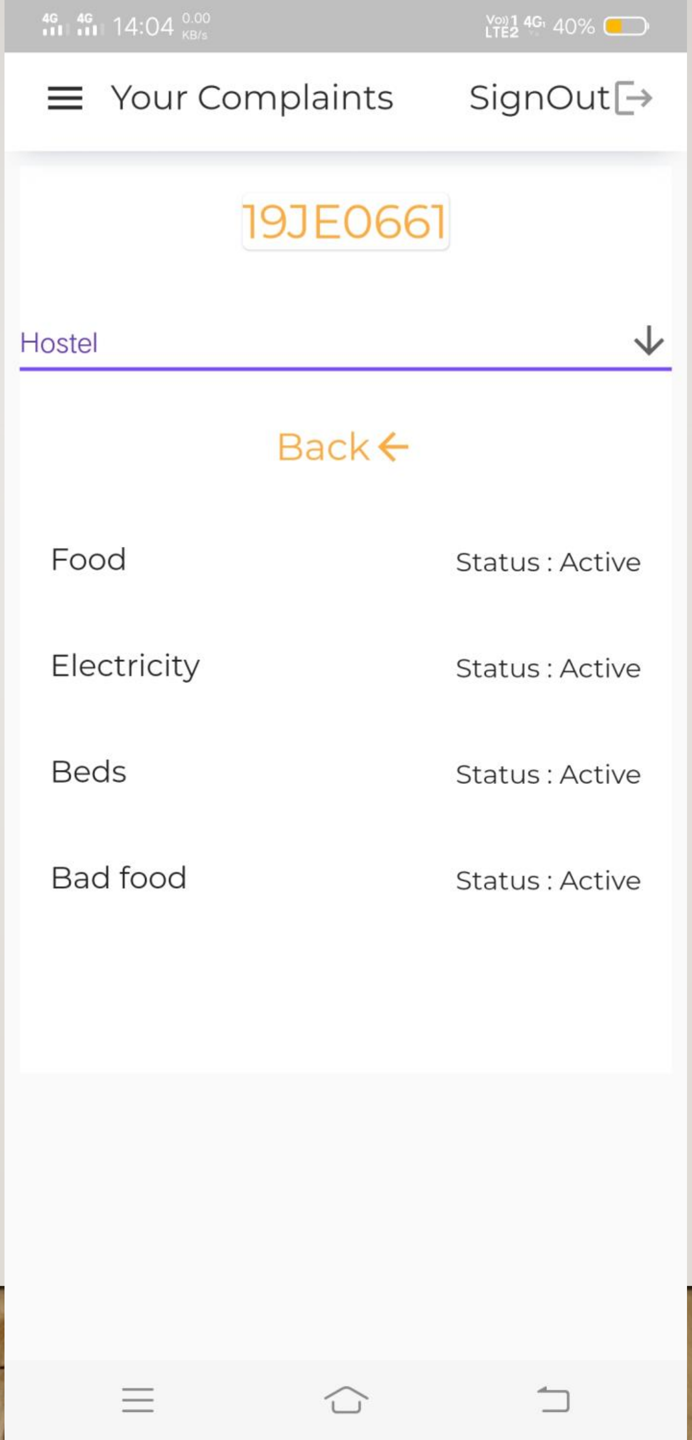
- Then the user needs to fill the details regarding Complaint – Name and Description.

- On Submission ,The Complaint is saved in firestore database under the topic selected.



6) YOUR COMPLAINTS PAGE (STUDENT)

- If , the user is Student Then he can access the feature to *Check Status of his/her complaint* from home page or side drawer
- When the user enters this page , he/she should select the topic of complaint from drop down menu .
- Then , all the complaints registered by user under that topic/domain can be seen as a list in descending order of date of complaint.
- Upon clicking any of the complaint , its complete details can be seen on dialog box.(name, description, status).
- This way , one can track the status of complaint.

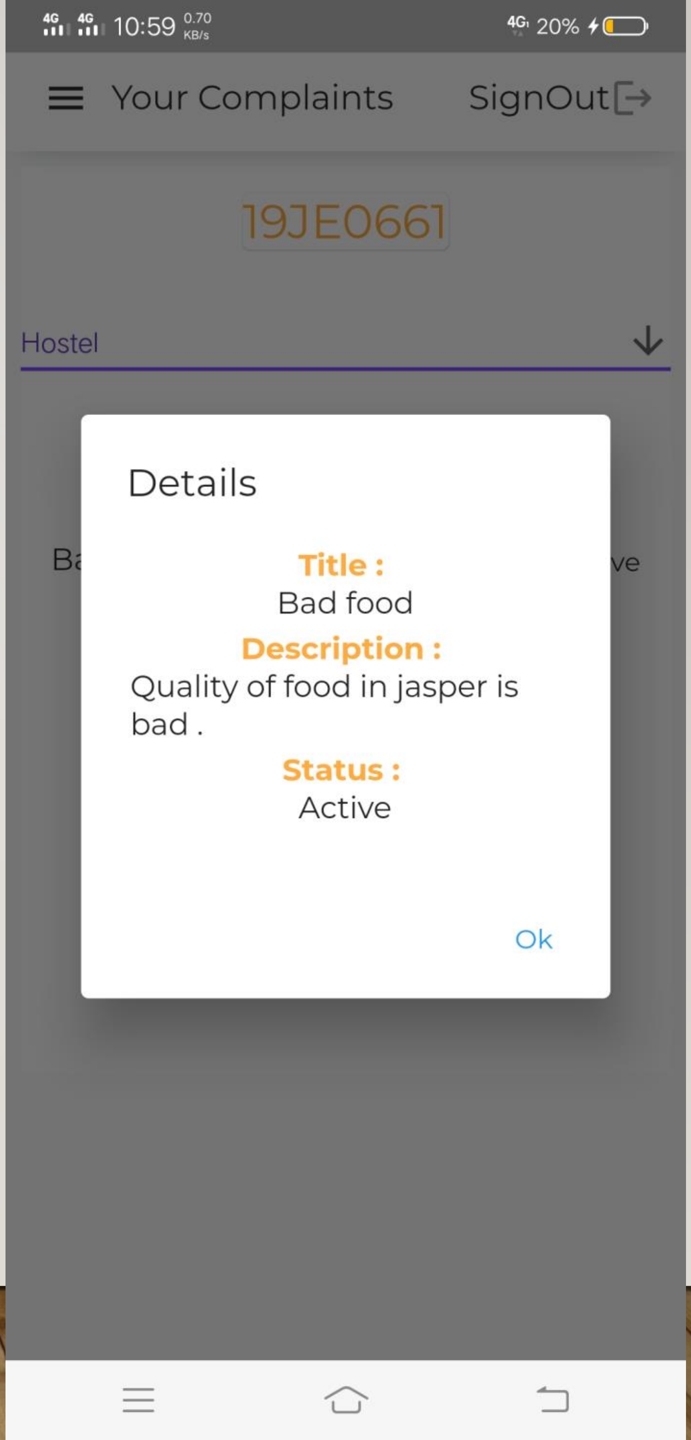


Food Status : Active

Electricity Status : Active

Beds Status : Active

Bad food Status : Active



Details

Title :
Bad food

Description :
Quality of food in jasper is bad .

Status :
Active

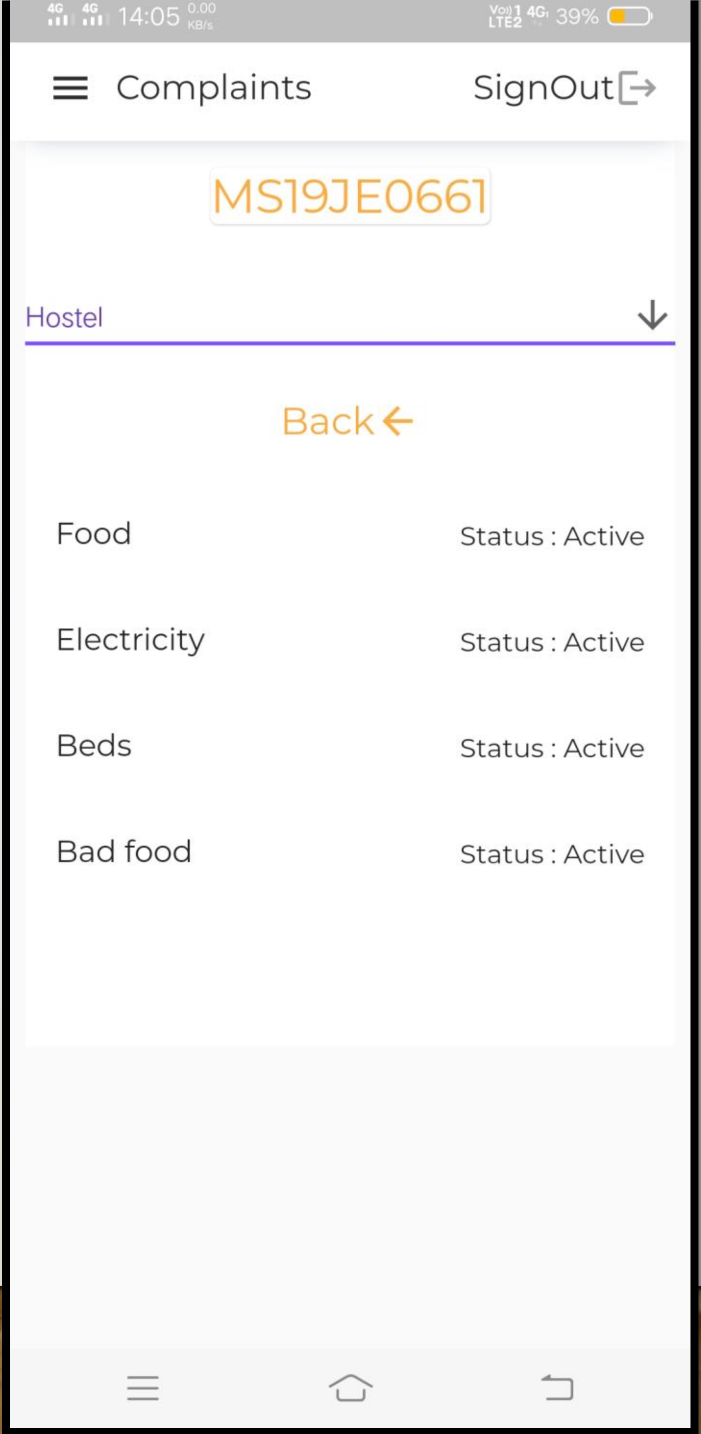
7)CHECK ALL COMPLAINTS (MANAGEMENT)

- If , the user is Management .Then he can access features to *Check the active complaints* and to *Check resolved complaints* from homepage or side-drawer.

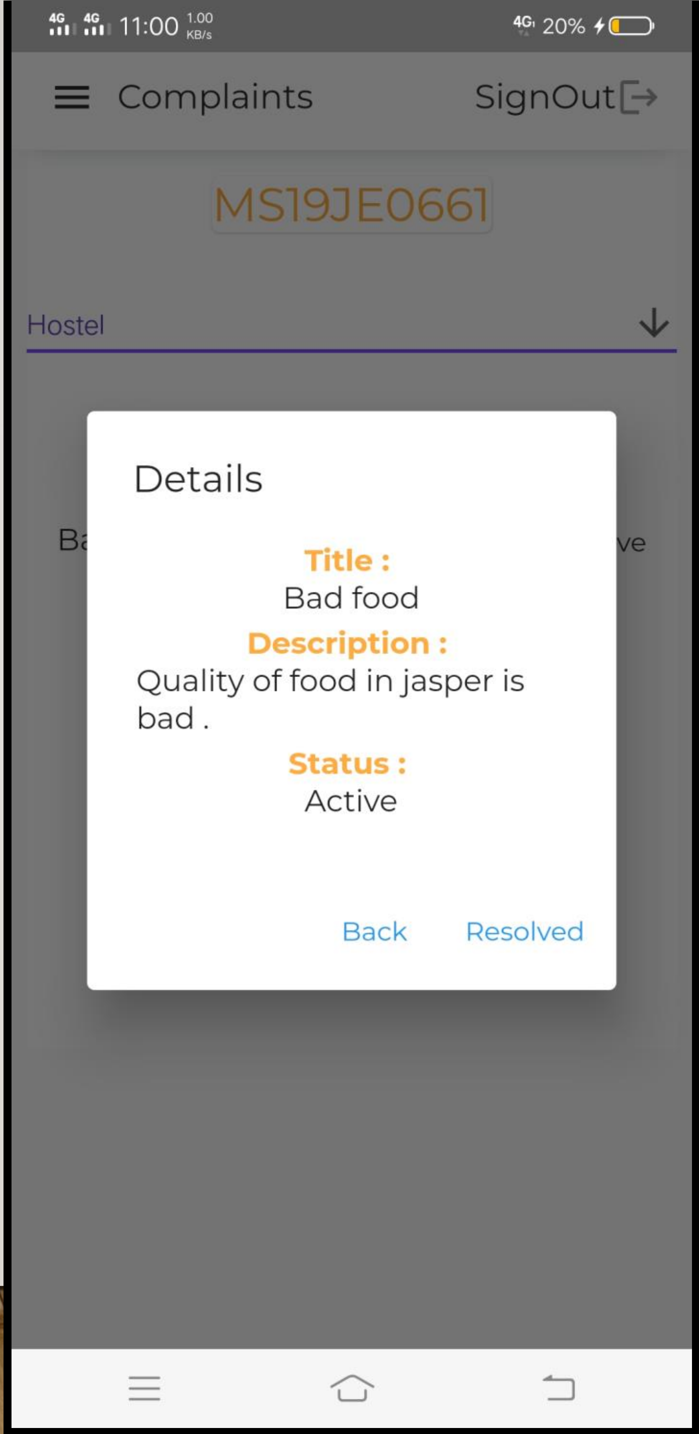
- When the user enters this page , he/she should select the topic of complaint from drop down menu .
- Then , all the complaints registered by all students under that topic/domain can be seen as a list in descending order of date of complaint.

- Upon clicking any of the complaint , its complete details can be seen on dialog box.(name, description, status).

- If complaint is active ,Then user gets access to a "Resolved" button clicking which changes the status from Active to Resolved.This way management can edit the status of Complaint.



Food	Status : Active
Electricity	Status : Active
Beds	Status : Active
Bad food	Status : Active



Details

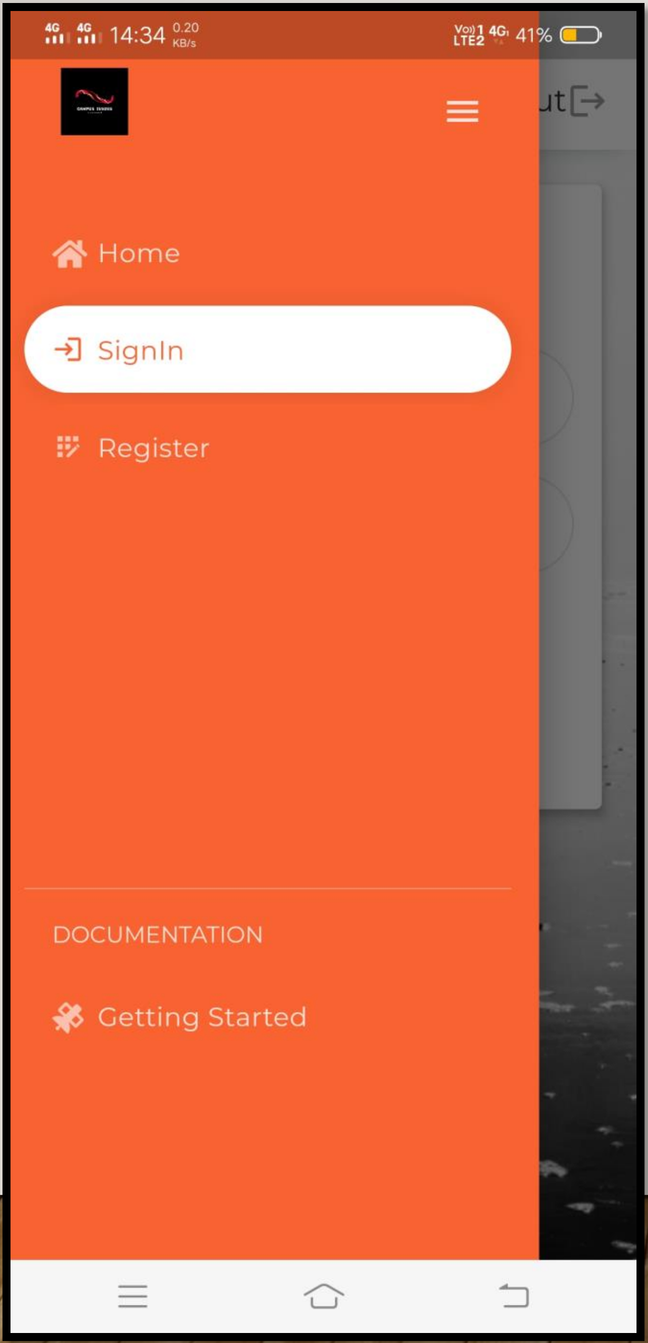
Title :
Bad food
Description :
Quality of food in jasper is bad .
Status :
Active

Side Drawer

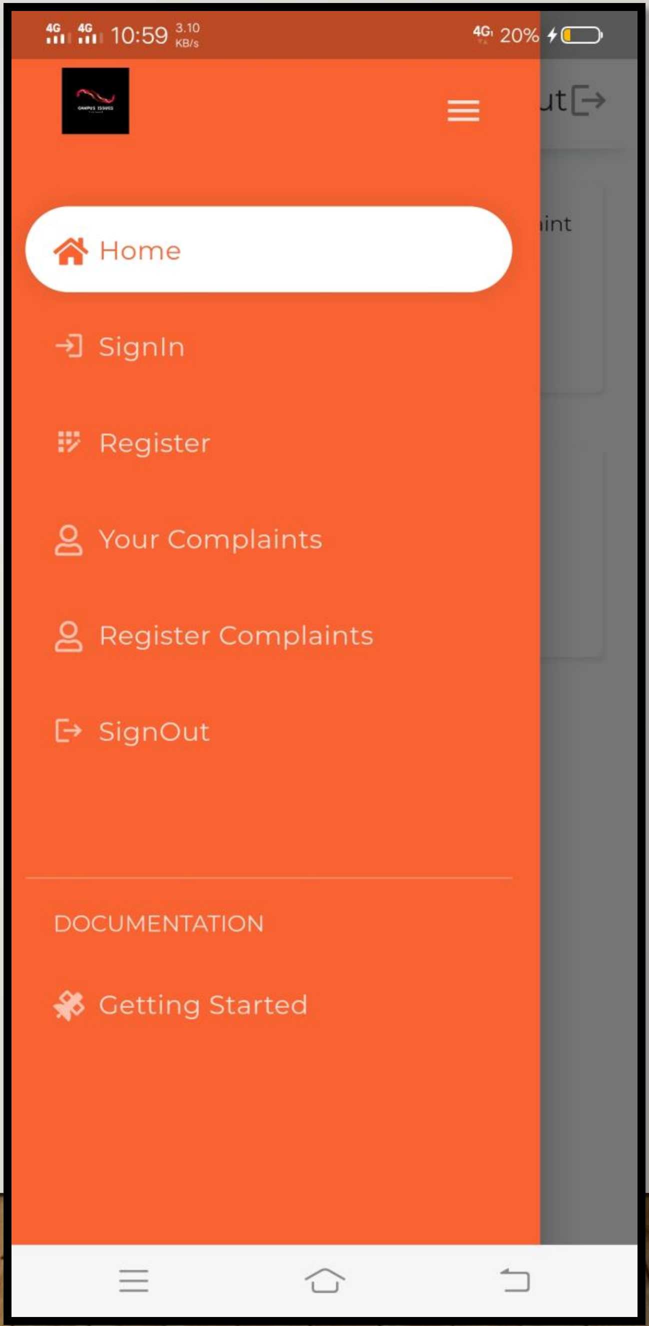
Side Drawer provides an easy way to navigate through different states and access different features.

This drawer is customized to different states of the app.

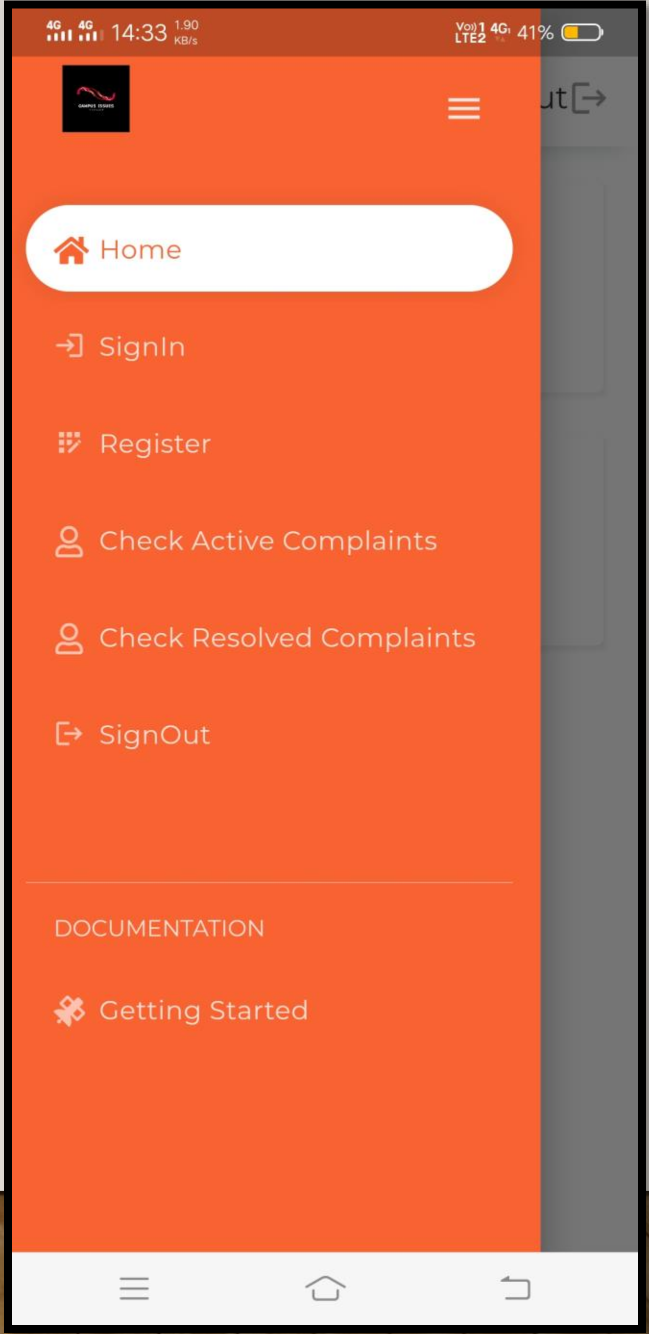
Drawer when not signed in

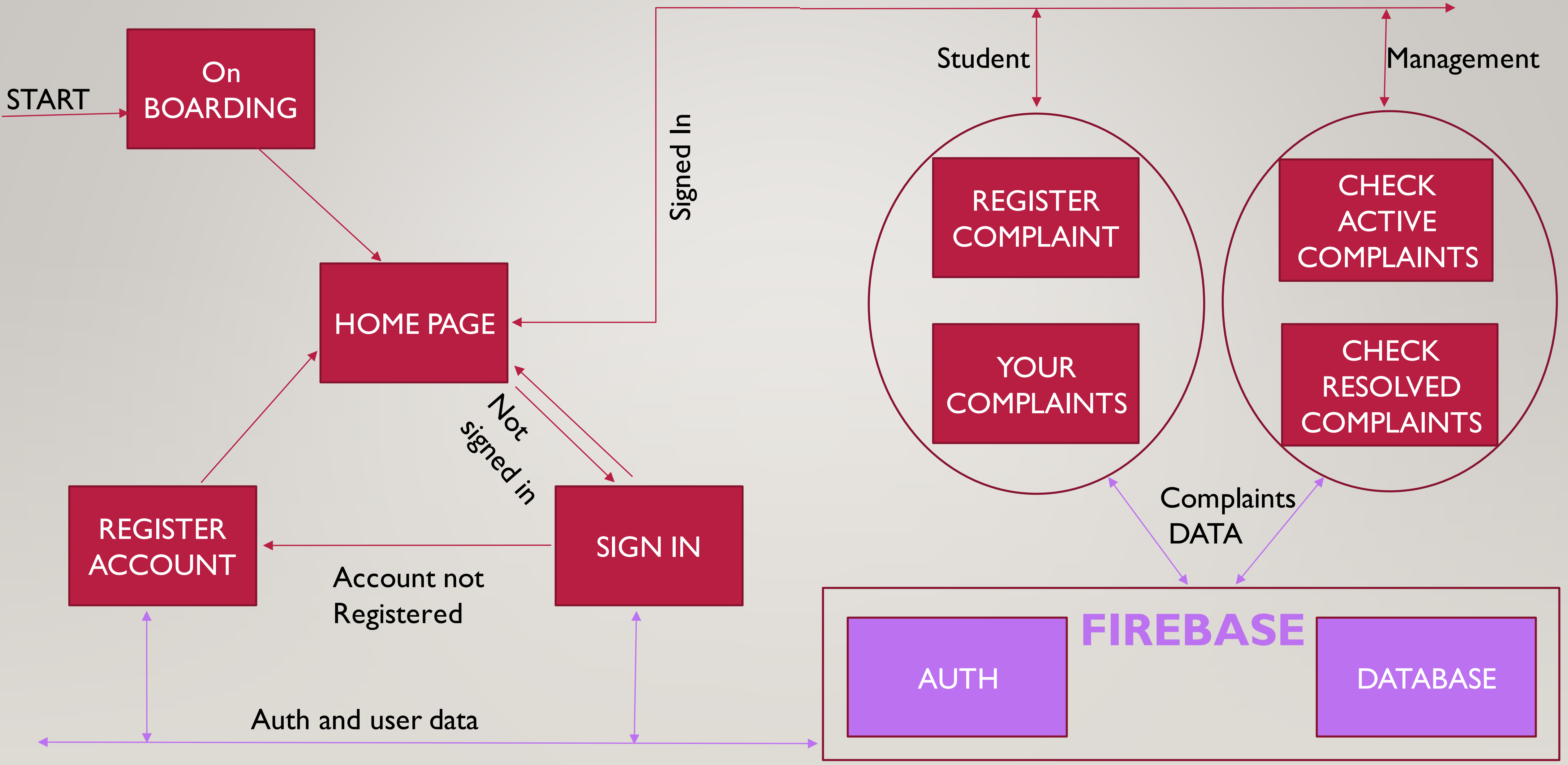


Drawer : user is Student

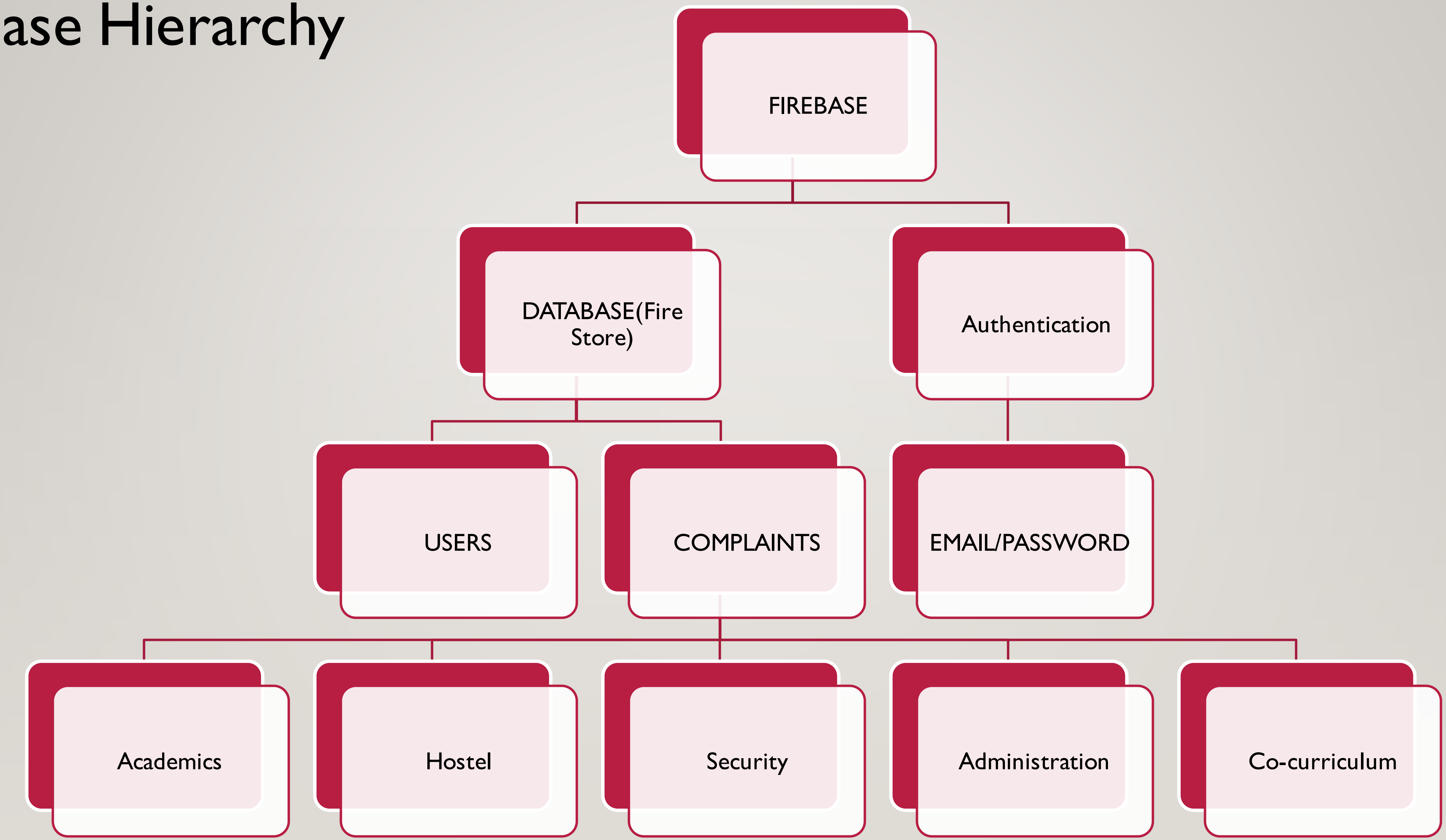


Drawer : user is management




START
On BOARDING
HOME PAGE
Signed In
Student
Management
REGISTER COMPLAINT
YOUR COMPLAINTS
CHECK ACTIVE COMPLAINTS
CHECK RESOLVED COMPLAINTS
Not signed in
REGISTER ACCOUNT
SIGN IN
Account not Registered
Complaints DATA
FIREBASE
AUTH
DATABASE
Auth and user data

Firebase Hierarchy



FEATURES :

1) Simple and user frendly UI
2) Security from Firebase Authentication
3) Reliable Database services from Firebase(Firestore).
4) Developed with Flutter UI toolkit which provides support to both IOS and Android with Just some basic addition of code.

Status of App :

1) Fully Functional
2) Requires some addition : Text field validation
3) NOTE :
In current application , During registration any one can be signed up as management . But this can be taken care by pre creating some management accounts before hand and sharing the credentials with management only and changing the registration process , where users can register only as Student and management can directly sign in without creating an account