**BILL ORGANISER**

**OBJECTIVE:**

The purpose of the app is to store and organize bills as pictures. Bills could be of restaurants, shopping or any other relevant bills can be uploaded and organized in tile views.

**TECHNICAL SPECIFICATIONS:**

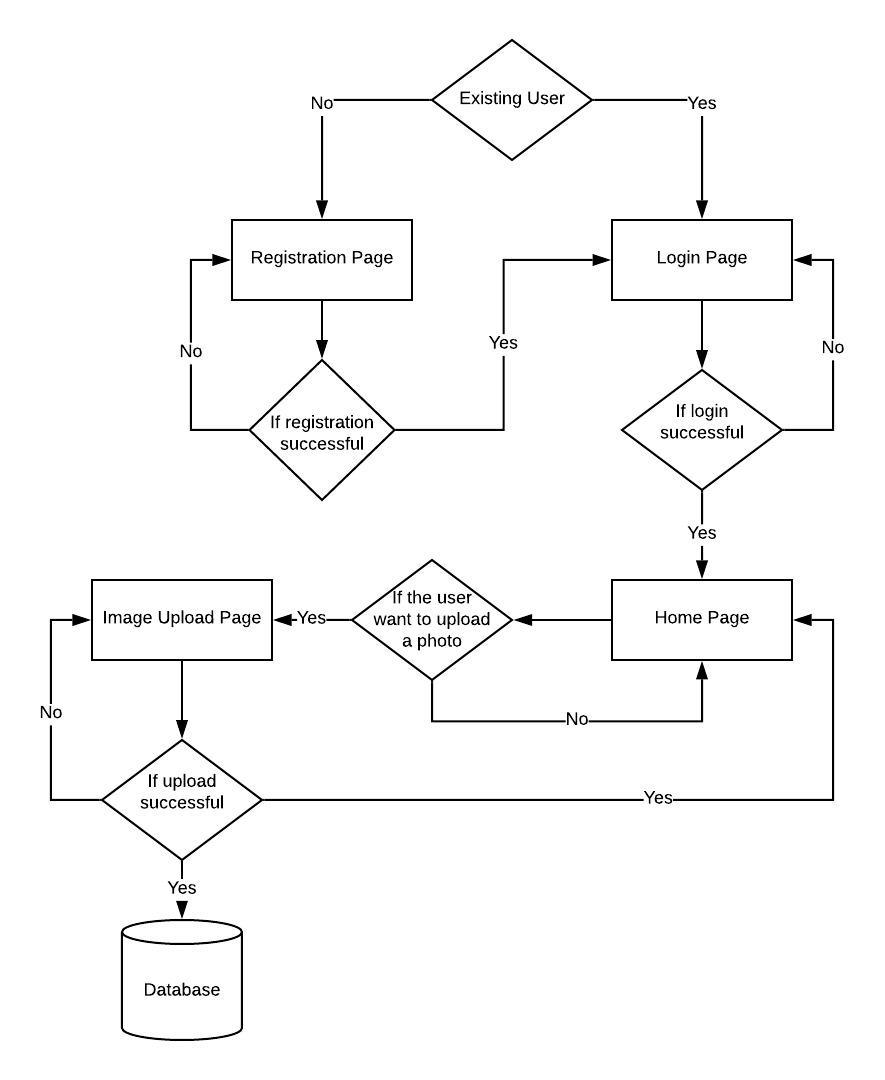
In this app, I have included Activity, Application, Fragments, Constraint Layout, Linear Layout, Drawer Layout, Toolbar, Floating Action Button, Coordinator Layout, Card View, List View, Navigation View, Rounded Image View, Recycle View, and Scroll View.

Apart from the android features, I have also included Amazon’s EC2 Cloud using Parse Server to connect the app to the cloud. Through this, I am able to store the user’s information on the cloud as well as the bills (pictures) in the cloud using parse server.

Also, I have done the validations for the Login and the registration and password change forms.

User authentication is required to make the use of the app. In the login page and the profile fragment where the user have the option to change the password. In the registration page, there are basic validations/checks were made so that a single user cannot register more than once.

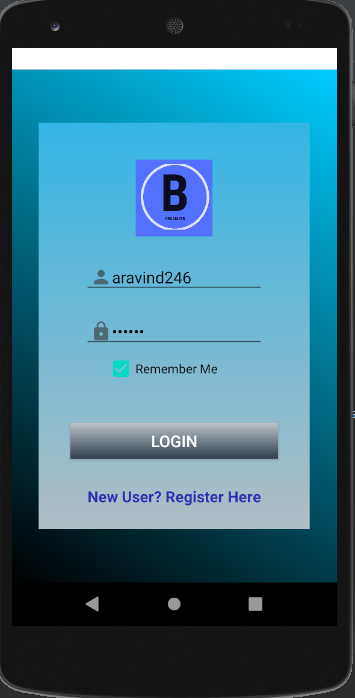
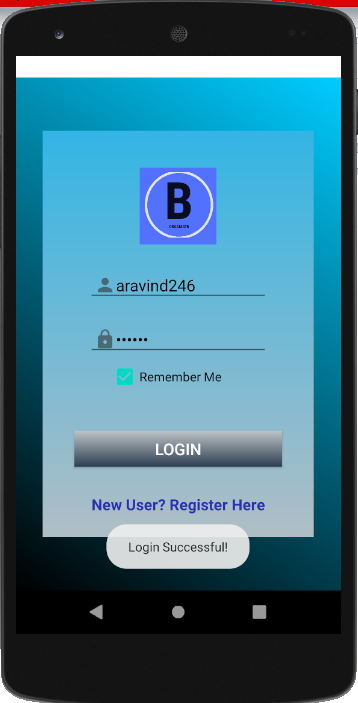
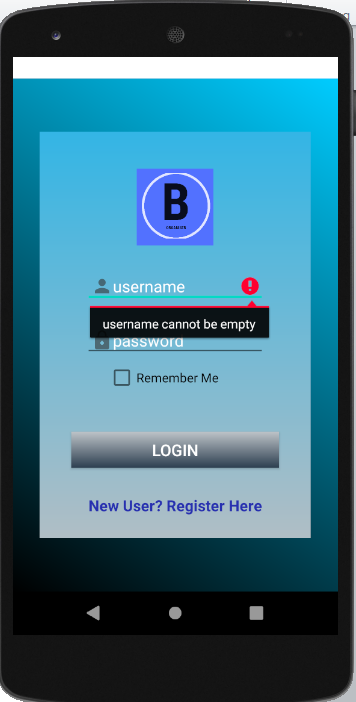
**WORK FLOW DIAGRAM:**



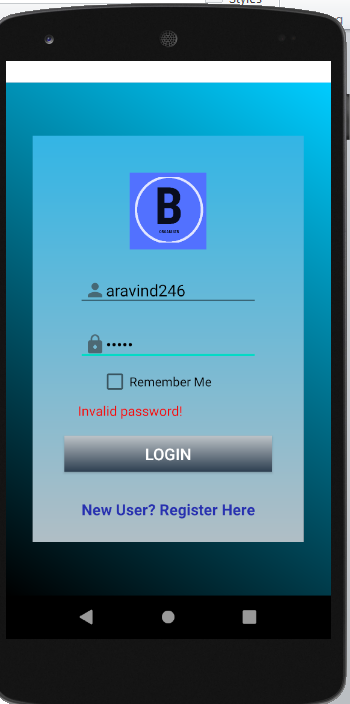
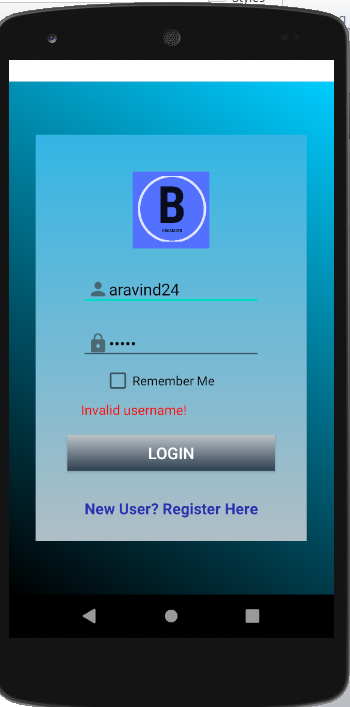
**SCREENSHOTS:**

The login page authenticates the user by using Amazon’s EC2 cloud with the help of parse server and performs actions accordingly.

Login Page On Successful Login Login Form Validations

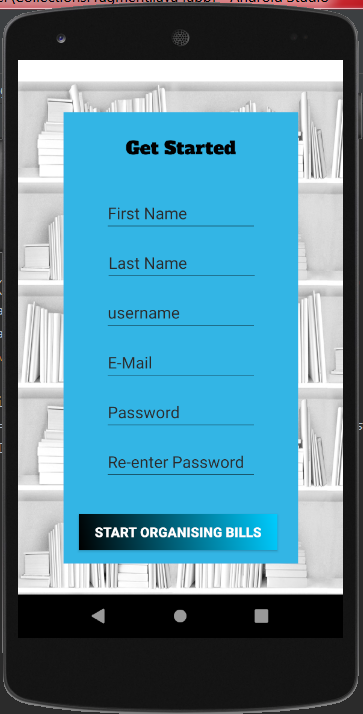
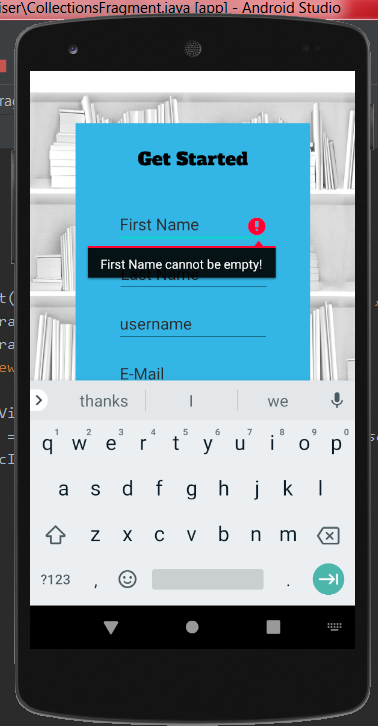
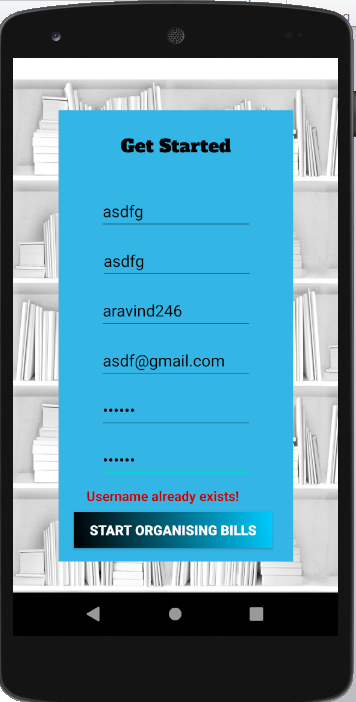
  

Invalid Password Invalid Username

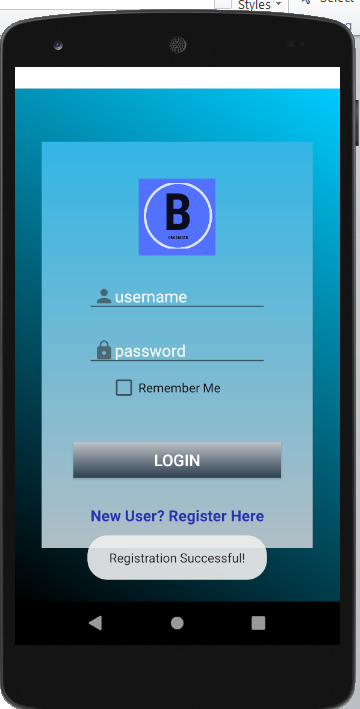
The Registration Form allows the user to register and provide app privileges. The username is verified from the Amazon’s cloud service to make sure there is no redundant data. The form is completely validated from the front end as well. For ex, email should be of the format [abc@def.com](mailto:abc@def.com). The form displays an error if the user fails to follow the validation rules.

Registration Form Registration Form Validations Existing username

Up on successful registration, the app redirects to login page leaving a feedback, a Toast Message, about the status of registration.

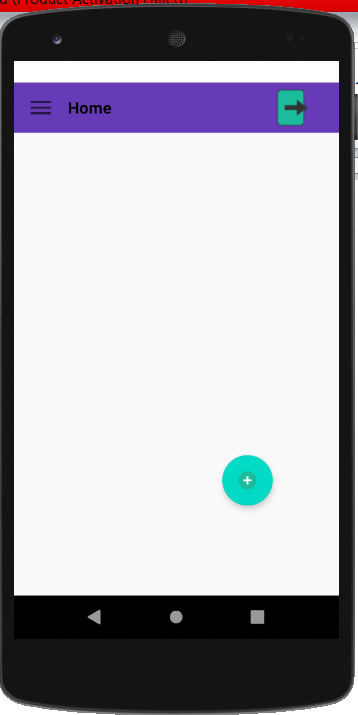
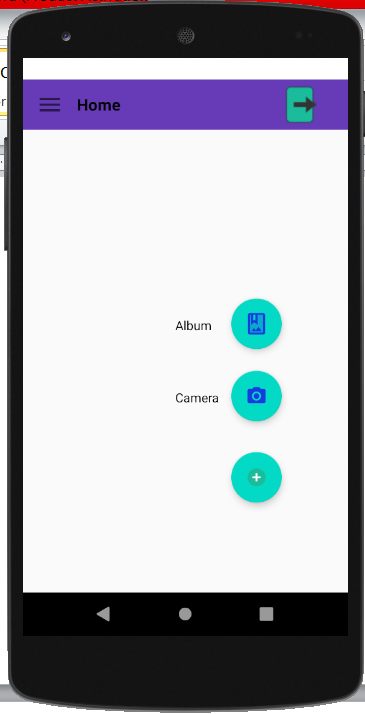
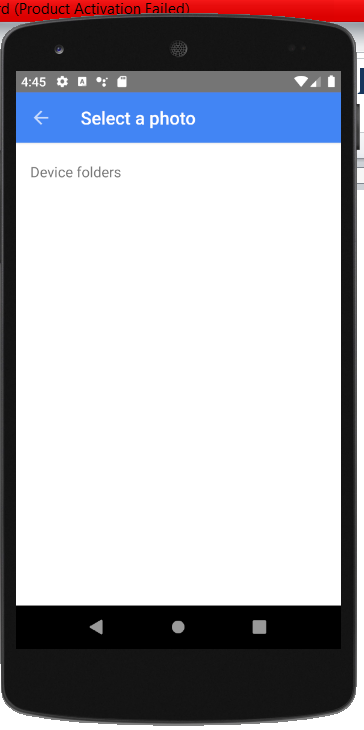
On successful registration



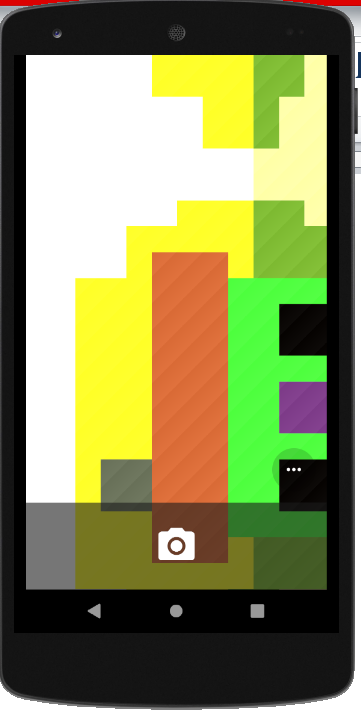
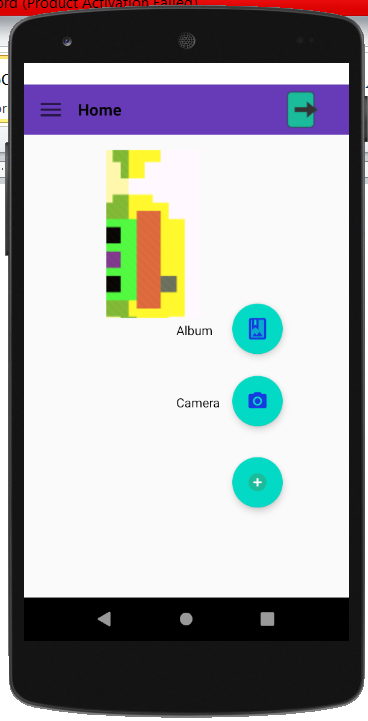
After Logging in, the home page displays the user’s uploaded pictures in the home fragment and also there is a floating action button allowing the user to upload pictures of bills from gallery/camera.

The top bar is the toolbar which provides option to logout and on the left side of the screen; the user can find the navigation menu to access other features of the app. The user can either swipe from the left side on the screen or click the options button on the left top corner in the tool bar.

Home Page Floating Action Button Open Gallery

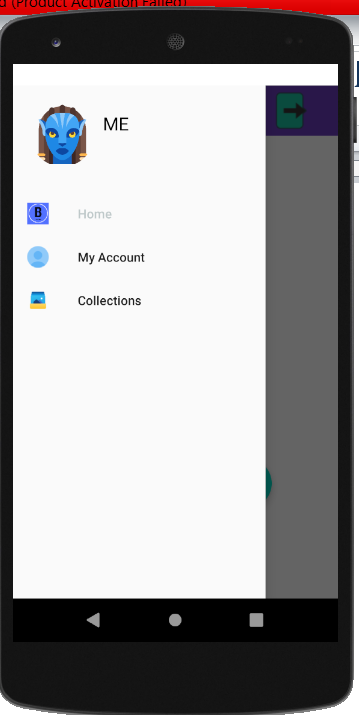
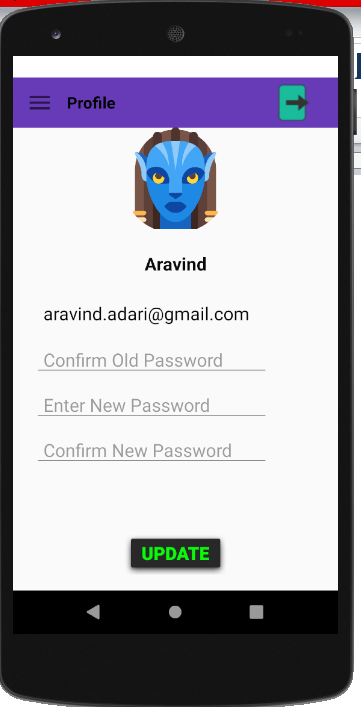
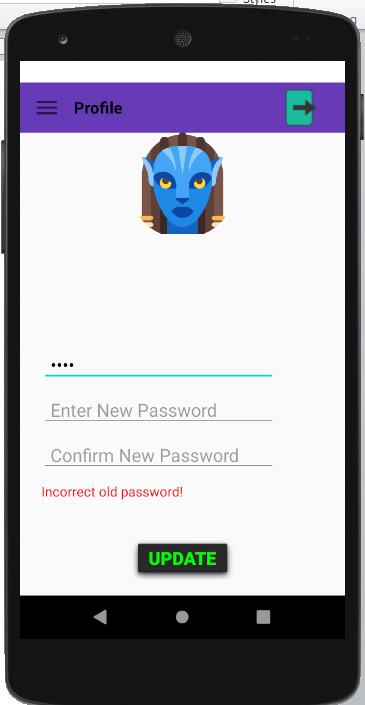
  

Open Camera Home Fragment after clicking a picture

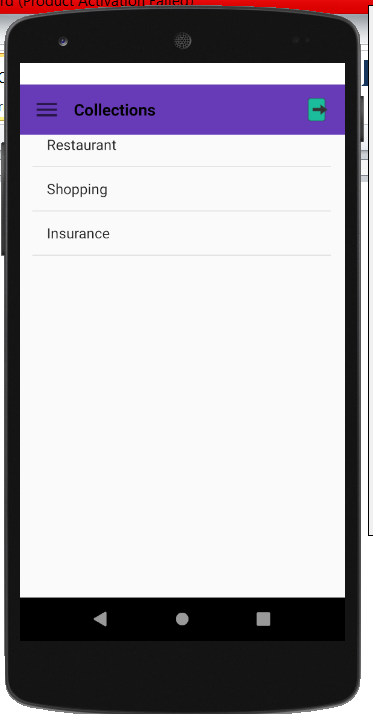
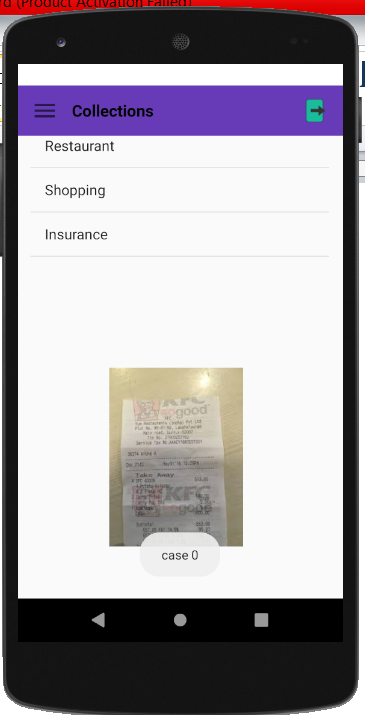
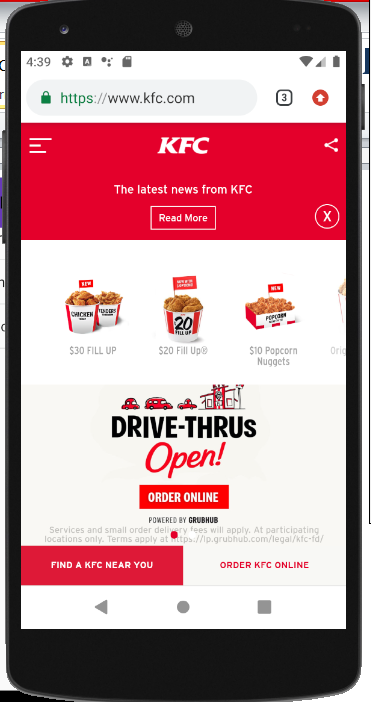
The navigation view consists of navigation header and other components to access the fragments like home, profile and collections. The Profile Fragment provides the user an option to change the password if they feel so. The app uses backend authentication to verify whether the authenticated user is using the app or not. And the password length should also be greater than 5 characters.

Navigation View Profile Fragment Profile Form Validations from backend

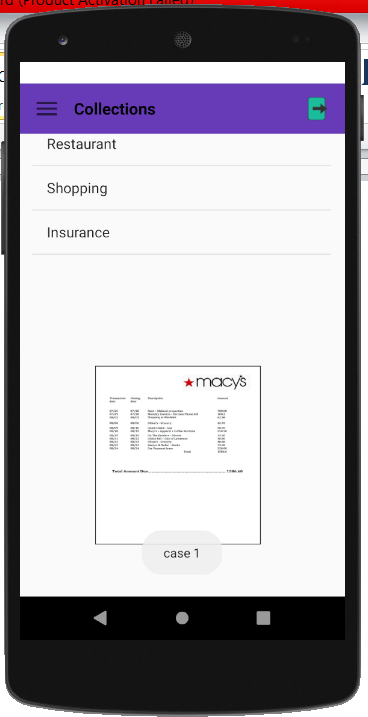
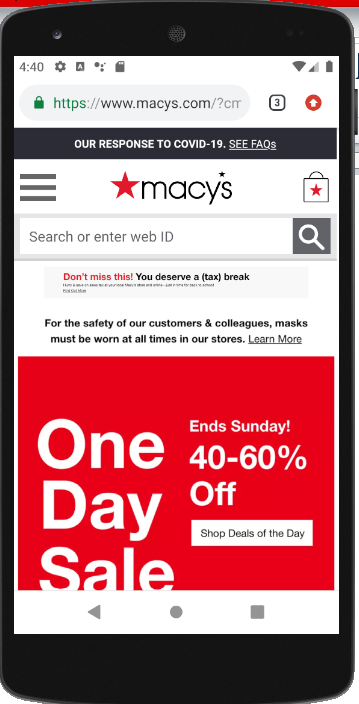
  

The Collections Fragment uses List View to display a list of possible bill items that this app supports and also provides sample images of bills and also allows the user to redirect to appropriate website on the click of the image.

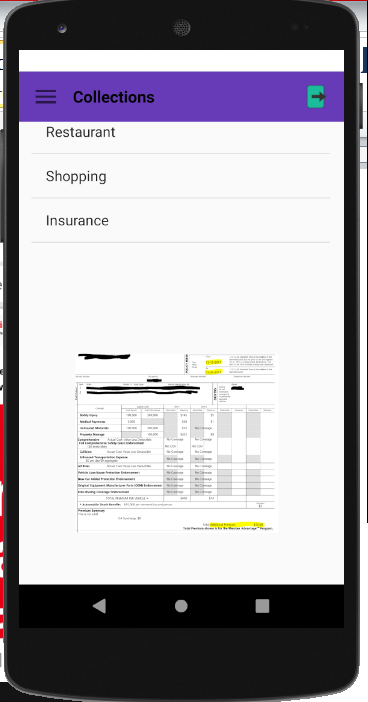
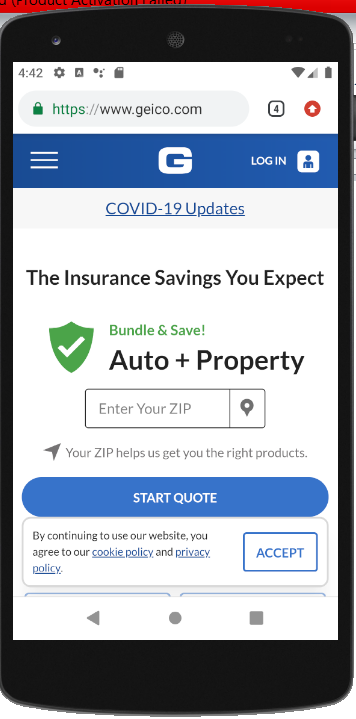
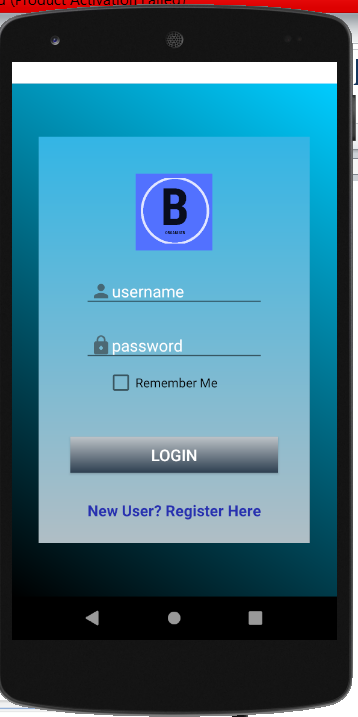
Collections Fragment On Selecting Restaurant Item On Image Click

On Selecting Shopping Item On Image Click

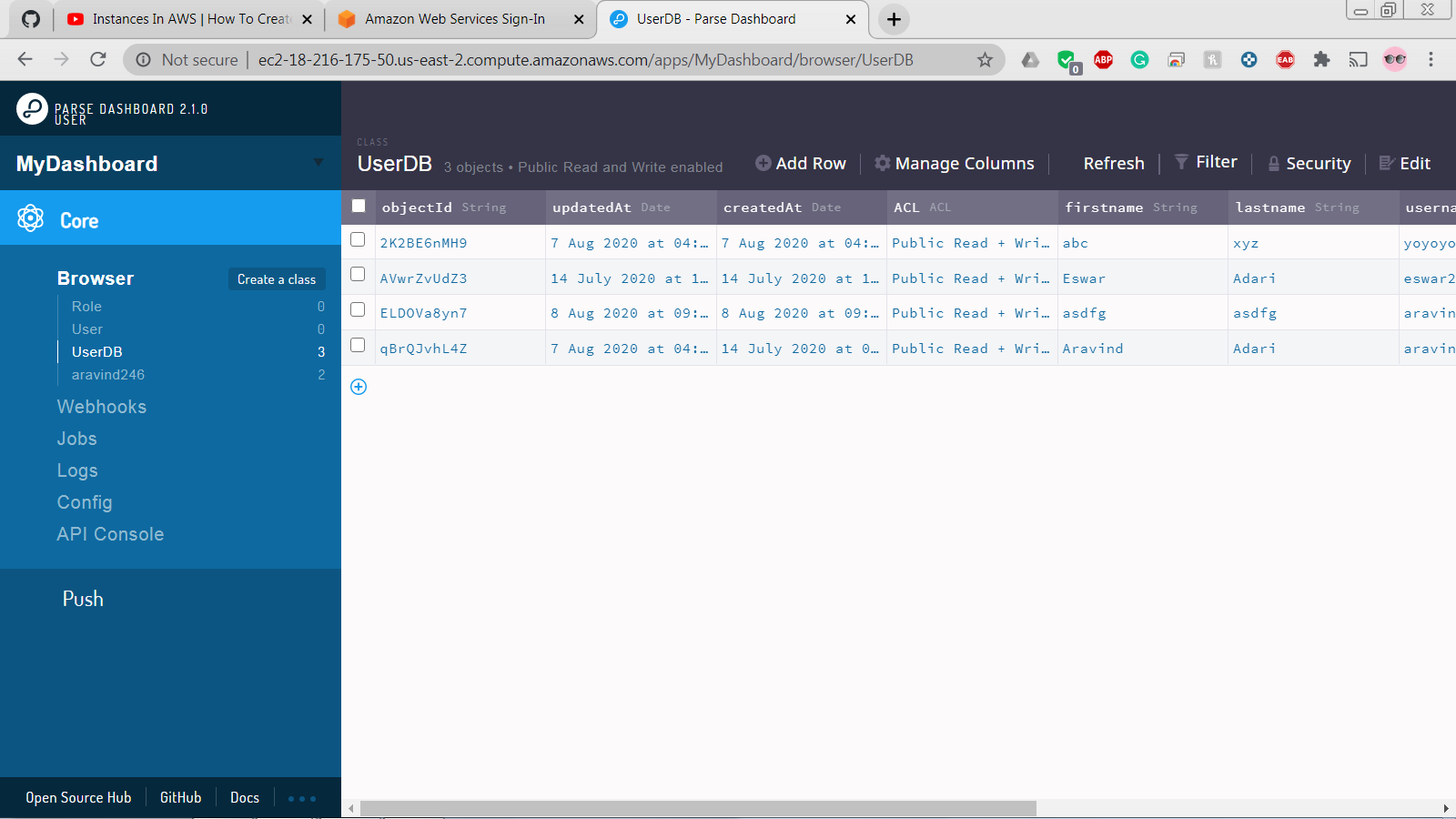
 

On Selecting Insurance Item On Image Click On Clicking Logout Button

Parse Server Dashboard

The Amazon’s EC cloud allows the user to store data in the cloud allowing the user to access it anywhere on the planet. The parse server dashboard is a medium which allows the administrator to see the data and able to manage it accordingly. Below, I have provided the screenshots of the dashboard displaying the user information.



As you can see on the right panel, the “UserDB” is the database instance created to store the user’s data. On the right side of the panel, each row/object resembles each user and their details.

Each user is considered as an object in the cloud database.

Although the app works fine with registering the user, here are some usernames and passwords to try and check the app.

username: “aravind246”

password: “asdfgh”

username: “eswar257”

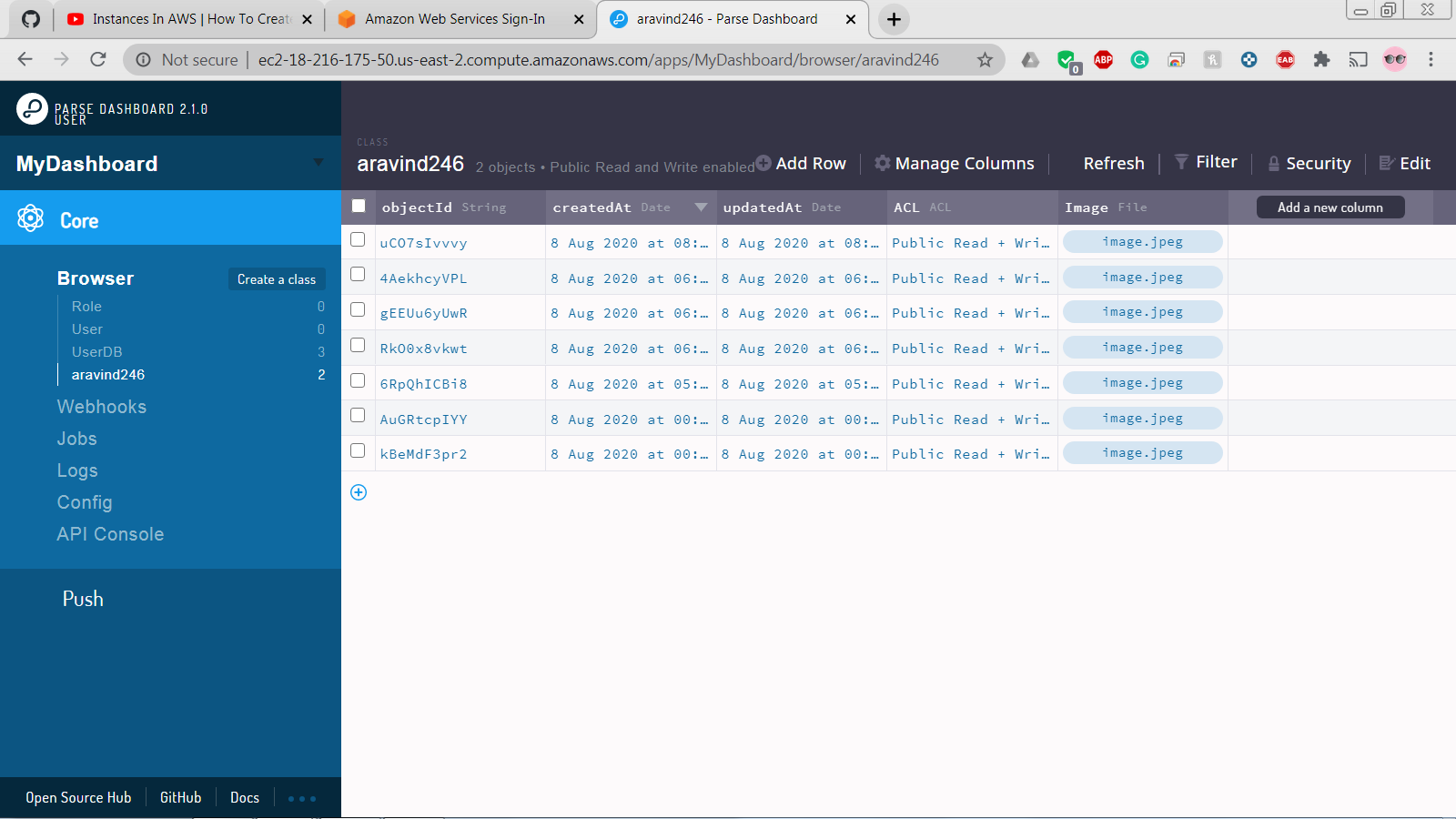
password: “itissecret”

username: “yoyoyo”

password: “123456”

Below the “UserDB” tab on the right, you can find “aravind246” which is nothing but another instance of database created up on uploading the images by the user “aravind246”. Meaning, I am creating an instance of database by the ‘username’ of the user which helps fast access to the data and also no need to perform join operations if the data is large.

You can see the screenshot of the table that user “aravind246”stored bills (images) in the Amazon cloud.



As you can see, there are no columns other than the implicit columns created by the parse server except the ‘Image’ column.

**CONCLUSION:**

My goal is to allow the user to upload the bills and organize them according the user’s choice. With this app, the user can easily register, login, upload images and verify them accordingly.