GitHub Username:pavankoduru

FUNAPP

Description

FUNAPP is an android application which will consume your time for your Entertainment Purpose. This app will displays some random jokes to entertain for some time. This app will contain authentication to login into and user can add his favourite joke into widget which will display joke on Home Screen. This app uses JokesAPI to fetch some random jokes and display to the user.This app will display ads provided by the Google.

Intended User

This app will be use full to any kind of people.

Features:

User can register and login through his email and password.
User can enter his desired No.of jokes between 1 to 100.
User can view Ads and react to that Ads.
User can add his favourite joke to Widgets which will display on Home Screen.
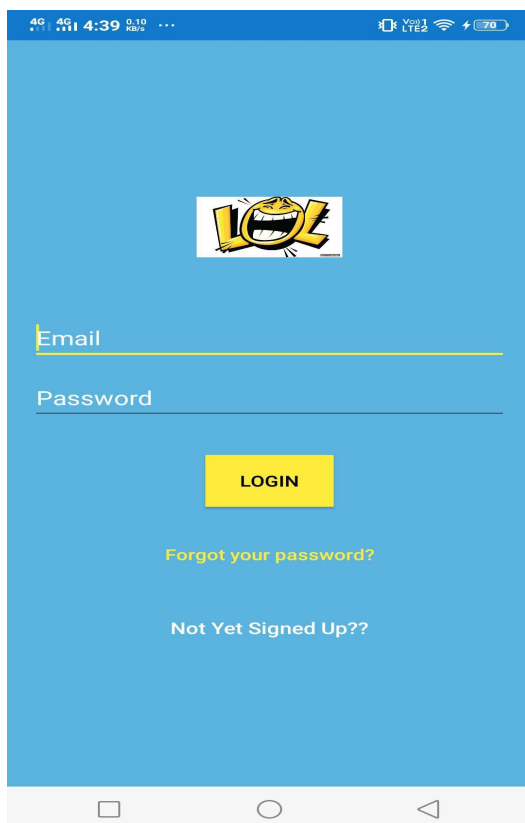User can Sign out and Remove his Account by himself.

APP Design Specifications:

All the strings used in Application will be extracted from strings.xml file.
All the colors used in Application will be extracted from colors.xml file.
All the dimensions used in Application will be extracted from dimensions.xml file.
All the themes used in Application will be extracted from themes.xml file.
All the static images used in Application will be extracted from drawbles.xml file.
Application is written solely in the java programming language.
Application supports right-to-left(RTL) layouts.
Application utilizes stable release versions of all libraries,Gradle,Android Studio.
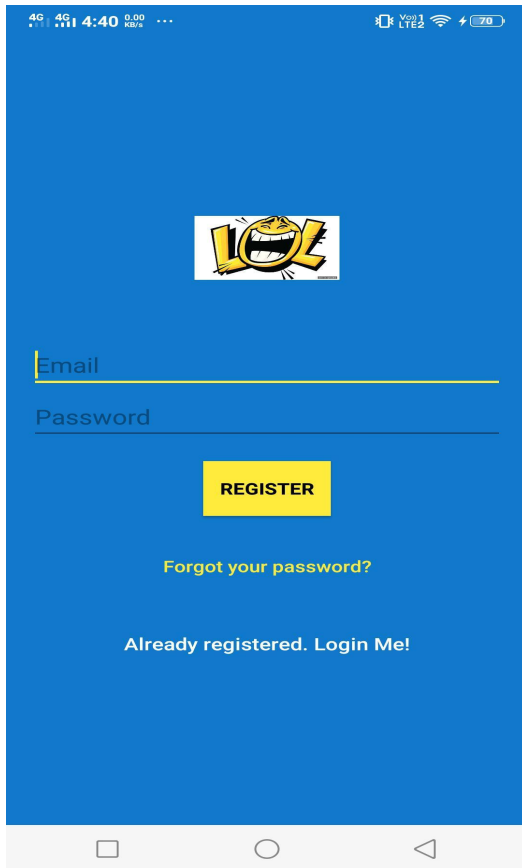
User Interface Mocks:
Screen 1

This screen is Splash Activity which will displayed a text called Have Some Fun.
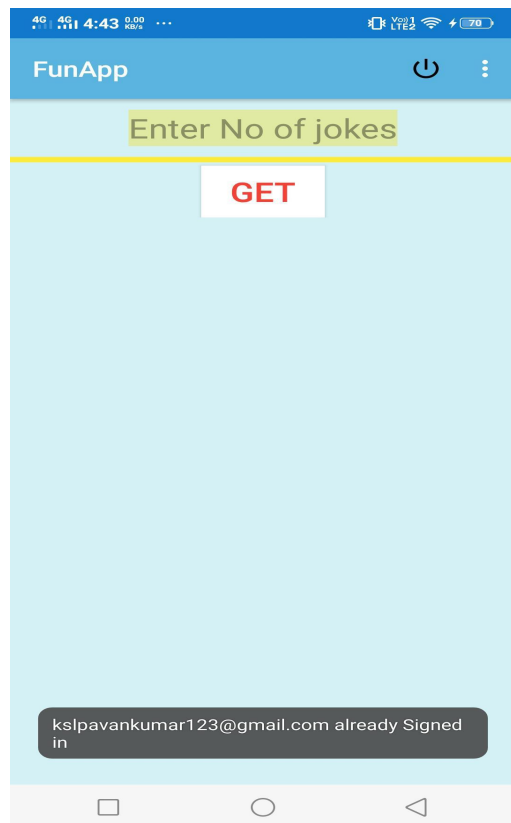
Screen 2

This screen will display two Text boxes to enter his email and password, and login button to login into his account. User can also register a new account if he is not registered yet.
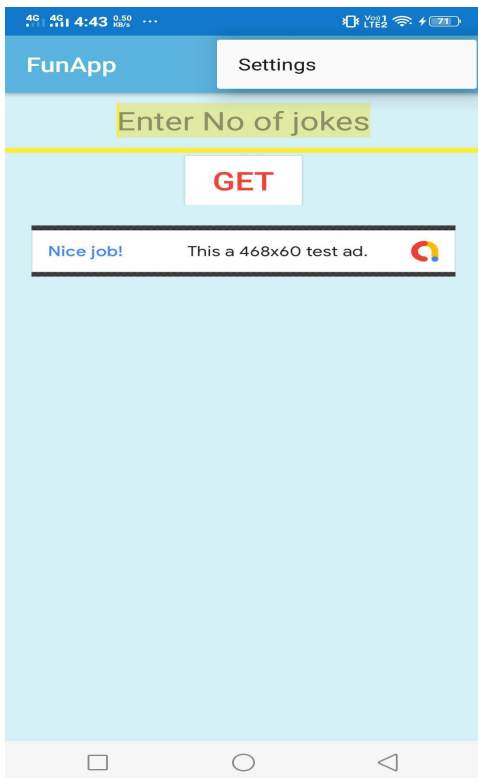
Screen 3

This screen will display the two Text boxes and Register button to register and link for forgot password task.
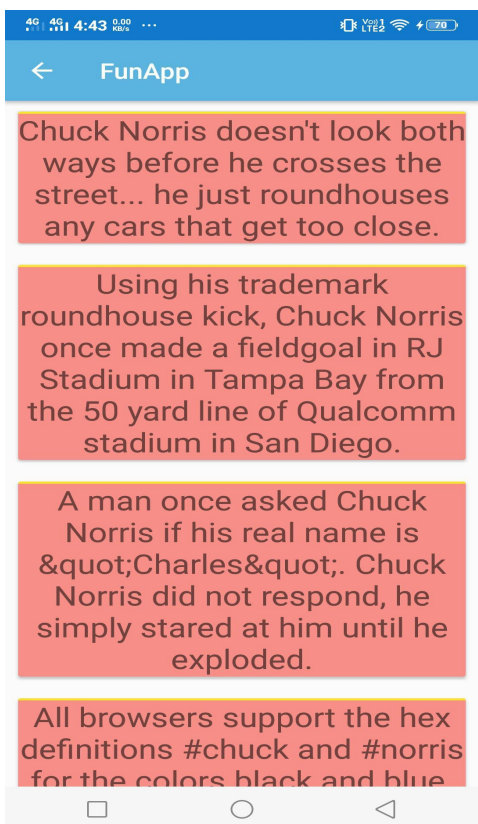
Screen 4

This screen will contains a Text box which will take number as a input from the user and a button to drive into another screen(detail screen),will also contains a sign-out button and action overlay button on the right corner. Screen will also display Ads bottom to the get button.
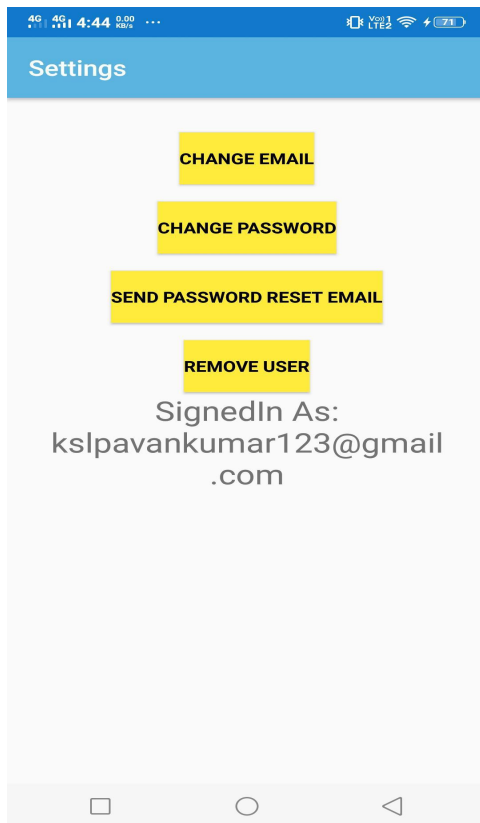
Screen 5

This screen will display the menu items whenever the user clicks on action overlay button.
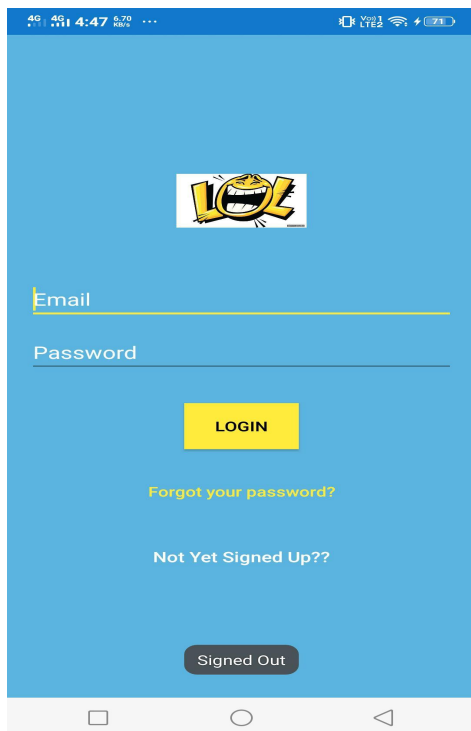
Screen 6

This screen will displays the list of jokes and user can click on his liked joke to display on widgets. Screen will also contains Ancestral Navigation Button to go back to Homepage i.e, screen 4.
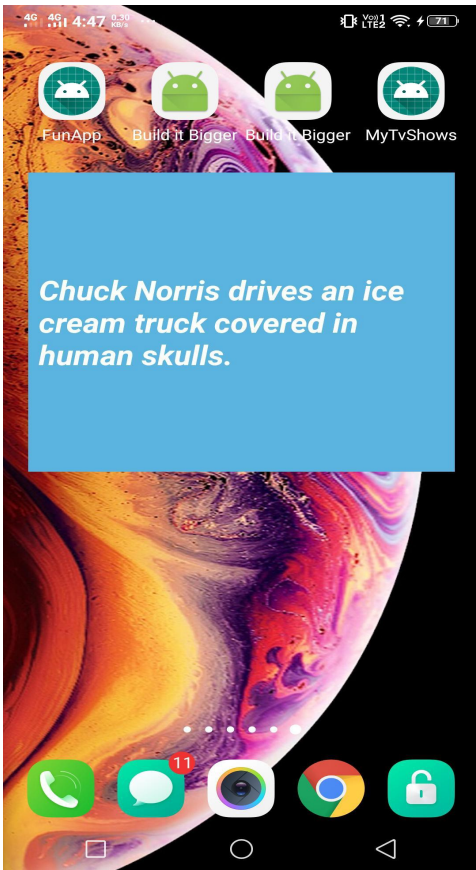
Screen 7

This is the screen when user clicks on the settings option which is located on Screen 4.

Screen 8

This is the Screen looks like When user clicks on the sign out button which is located on Screen 4.

Screen 9

This is the screen with Home screen widgets when user clicks on jokes displayed in Screen 6.

Key Considerations:

How will your app handle data persistence?

This app will use Async Loaders for fetching the data from the server and uses Firebase Authentication to Authenticate the user.

Describe any edge or corner cases in the UX.

This app will contain a Ancestral Navigation button which will be fixed on left corner of detail screen(Screen 6).
This app will contain a Toast Messages to help the User.

Describe any libraries you'll be using and share your reasoning for including them.

Firebase-Auth : To authenticate the user -Version 16.0.2,
Google Ads :To display Ads(Banner,Interstitial) -Version 15.0.0,
Butterknife for binding the views -Version 8.2.0,
Recyclerview:To display items in a recyclerview.

Describe how you will implement Google Play Services or other external services.

Firebase Authentication to authenticate the user
AdMob to Display Ads.

Next Steps:

Required Tasks:

Task 1: Project Setup

Creating the new Project and naming it as FunApp. Configure the libraries, dependencies for app functionality.

Task 2: Implement UI for Each Activity and Fragment

Design layout for Splash Screen.
Design layout for Login Activity.
Design layout for Register Activity.
Design layout to enter the input from the user..
Design layout for displaying the jokes in a recyclerview.
Create menu with options Settings,Sign-out.

Task 3: Authentication

Implement authentication using Firebase in order to use this app.

Task 4: Data Fetching

Fetching the data from the network.

Task 5:Make the App material

Add Toolbar AppBar and check the color Scheme for Attractive user interface Use of icons in the project will be appropriate to the context.
Text will be readable and attractive in design.

Task 6:Testing

Write Espresso UI tests to the application.

Task 7:Generating signed APK

Sign the app in release mode and submit application.