**Task to be Submitted;**

1. Attached is a mockup of a Events App
   1. Events landing page
   2. Event View with details and tickets to book
   3. Payment page when customer proceeds to pay
2. Create a Web service passing static data to the App to Mimic how an API providing the data would work
3. You are required to build the Workflow for either Android or IOS

**What will be reviewed**

* 1. Execution of provided UI/UX
  2. Understanding of integration of Apps and API's
  3. Understanding of different elements on Android / IOS
  4. Quality of code-base
  5. Speed of execution

# Description

PesaApp is intended to be an event booking mobile application. The users should be able to

* 1. Browse through specific categories of events.
  2. Favorite events they fancy.
  3. View their favorites
  4. Check a specific detail of an events
  5. Pay for an event.
  6. Search for an event

# DESIGN

## UI

1. Login
2. Signup
3. Landing page
4. Events page - various categories like gaming, art, music, sports
5. History page – extra work
6. Specific event view
7. Payment view

## Service

1. Create tables
   1. Verification
   2. Categories
   3. Favorites
   4. All events – should take data from all table categories
   5. Featured events
2. Design table structures
3. Create endpoints

# Libraries and tools

* Gradle – v4.6. Build automation.
* Android – v3.4.1
* Picasso – stable v2.71828. Loading and cache of images.
* Livedata - v1.1.1. Lifecycle observable data holder.
* Retrofit – v2.5.0. Network calls.
* Adobe xd – design Login and Sign up page.

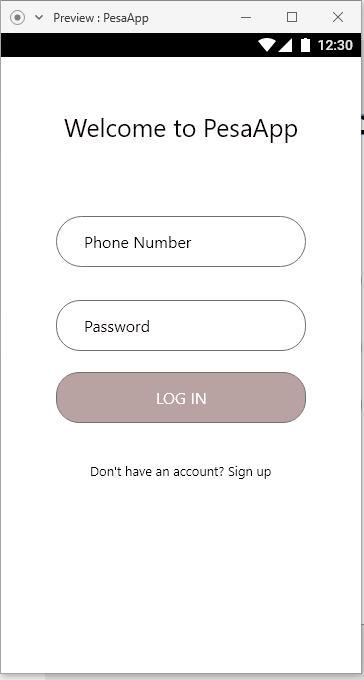
# Development core points

This will show the specific activities of the application and the tasks to be carried out in each task.

### Task 1: Project setup

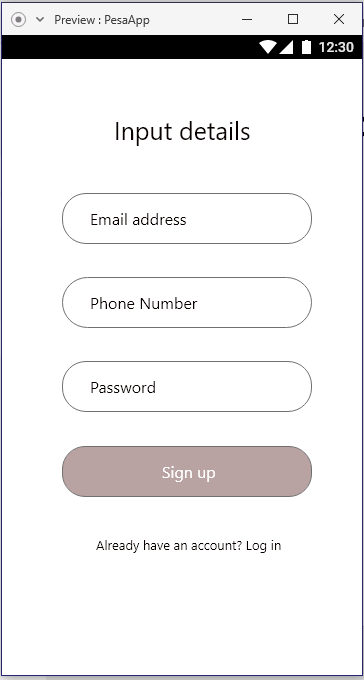
* Create new project and add dependencies.
* Enable internet access in manifest file
* Enable Use of Cleartext Traffic. This is because our node application communicates in HTTP which is not allowed by default
* Add icons and images

### Task 2: Login Activity



* Add phone textview for user phone number. Change input type to phone\_number
* Add password textview for password input. Change input type to password.
* Add log in button. Should validate user is registered. Display appropriate information on missing information, unsuccessful or successful login.
* Sign up textview to transfer to sign up activity.

### Task 3: Sign up activity



* Add phone textview for user phone number. Change input type to phone\_number
* Add password textview for password input. Change input type to password.
* Add an email textview. Change input type to email address.
* Add sign up button. Should validate user is registered. Display appropriate information on missing information, unsuccessful or successful login.
* Log in textview to return user to previous activity.

### Task 3: Events landing page.

Using the sent Events Landing V image:

* Add a search view. This will submit user searches on specific events.
* A recycler view to see featured events. Should use vertical Linear Layout manager. Design the layout for featured event items.
* A recycler view to categories of events. Should also use vertical Linear Layout manager. Accompanied by item layout.
* A recycler view to see a user’s favorited events. Use a vertical Linear Layout Manager. With its item layout. Should display only 3 items initially
* Textview to see all favorite events to take user to another activity.
* More events displayed in a recycler view. Design the layout. Include the heart icon to favorite.
* A button to view all events available
* Change the parent activity in manifest file.

### Task 4: Specific events activity

As per the sent Events Page Extended image, tasks will be:

* Change the parent activity in manifest file.
* Add icon for sharing (intent) and favorite in tool bar.
* Add image view.
* Add title text view. Change font and size.
* Add host text view.
* Add icons and text views for location and time.
* Display the information description of event.
* Add view to select ticket
  + Factor in the quantity of tickets to not go below zero
* Button to go to payment activity.

### Task 5: Payment activity

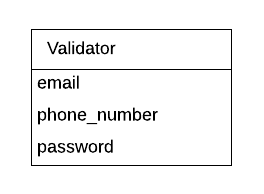
* Display event organizer account name and total to be paid.
* Provide buttons pay with a card or with M-Pesa.
* Button for users to add new card
* Button to complete payment

#### Task 6: Create service endpoints

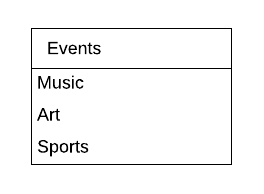
* Create sign up service. Check if user is already registered and notifies the application to handle.
* Create log in service. Check if user exist else and notifies the application to handle. Provide toast or Snack Bar.
* Search endpoint to return an event looked up by user.
* Featured event endpoint
* Categories endpoint to pull data for specific category events.
* An endpoint to supply all available events.

#### Task 7: Create tables

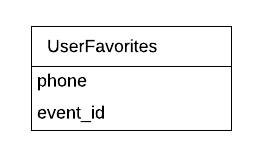
Validator table to hold user credentials.



Categories table to hold all the events available



User favorites table to hold the events that a user like.



Categories table has data on events of one category



Early\_bird\_price and Advance\_day\_1 fields and the other 5 fields in category table