

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity, Fragment and Notification](#)

[Task 3: Build a Content Provider](#)

[Task 4: Build a Sync Adapter](#)

[Task 5: Build a widget](#)

[Task 6: Show data in app \(using Loaders\)](#)

[Task 7: Implement Add To Watchlist Feature and Firebase Notifications](#)

GitHub Username: [lakshay-chhikara](#)

Events Guide

Description

Interested in events like Sports, Dance, Music, Coding, basically any type of event but can't find a way to know where they are happening around you?

Define the type of events you are interested in and get the details of any such event happening around you, add interesting events to your watchlist and share them with your friends.

Intended User

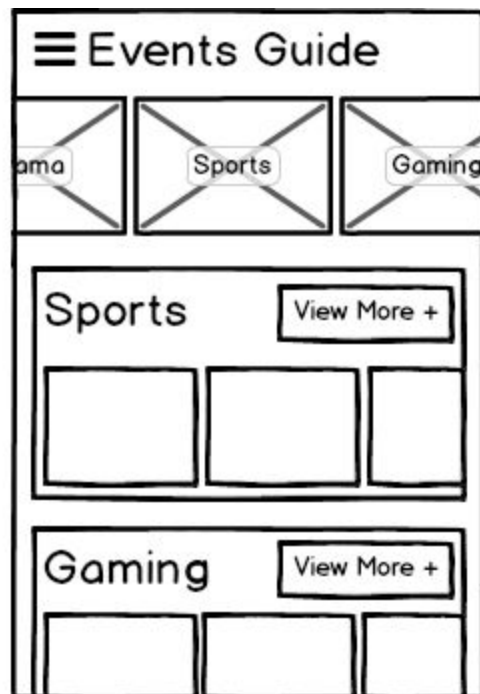
An app for those event fanatics who are interested in exploring interested events happening around them.

Features

- Get details of events happening around you
- Add interesting events to your watchlist
- Get notified of any changes about events in your watchlist
- Share events with you friends

User Interface Mocks

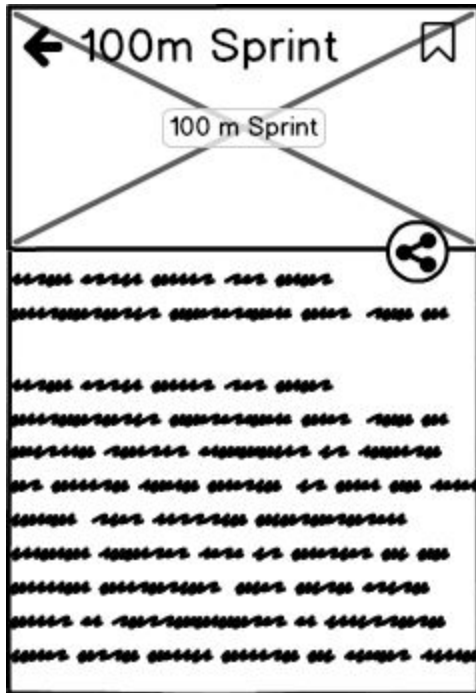
Screen 1



The top (ViewPager) contains a list of event categories. Click on any of the category to browse the events of the particular category.

Then, every event category along with their 3 latest events with a View All button to explore all events of that category.

Screen 2



This contains the event details like name, description, location, time along with the options to share or add to watchlist.

Widget

WatchList	
100 m Sprint Place	Date
Blind Coding Place	Date

Widget contains the user's watchlist events

Key Considerations

How will your app handle data persistence?

A content provider will provide an interface for the local database built on device.

Describe any corner cases in the UX.

- Take user directly to event detail page if he clicks on a notification showing information about a particular event. Same for an event category as well.
- Clearly indicate to user if internet not working.
- Show data from cache if internet not working but warn him that data may be outdated.
- Update Added to watchlist UI element only after the operation to add event in watchlist is successful.
- Inform the user clearly if there is no data available for any of the event category or any piece of information is missing for an event.

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

- Firebase: Transfer data from server to app.
- Glide: Loading and caching of images.
- Design Support Library: Implement Material Design.

Describe how you will implement Google Play Services.

- Firebase Analytics: Analyze user behavior within the app to make further decisions
- AdMob: Monetize the app

Next Steps: Required Tasks

Task 1: Project Setup

- Create Firebase console project
- Connect Firebase project with application
- Configure rest of libraries

Task 2: Implement UI for Each Activity, Fragment and Notification

Build UI for:

- Main Screen
- Category Events Screen
- Event Details
- Notification

Task 3: Build a Content Provider

Create a content provider to provide an interface for operations on local database used to:

- Cache event data.
- Store user watchlist events.

Task 4: Build a Sync Adapter

Sync Adapter to periodically sync data from cloud

- Create layout
- Use content provider interface to update incoming data

Task 5: Build a widget

Design and implement a widget to show upcoming user watchlist events

- Create a layout for the widget
- Connect widget with Content Provider

Task 6: Show data in app (using Loaders)

Use loaders to move data to app views from content provider.

- Implement loaders on all screens

Task 7: Implement Add To Watchlist Feature and Firebase Notifications

Add event watchlist feature to keep user updated with an event changes.

- Sign Up user for notifications regarding the events added in watchlist.
- Implement Firebase Notifications in the application to show any changes made to an event in the user's watchlist.