

## How to Use this Template

1. Make a copy [ File → Make a copy... ]
2. Rename this file: “**Capstone\_Stage1**”
3. Replace the text in green

## Submission Instructions

1. After you’ve completed all the sections, download this document as a PDF [ File → Download as PDF ]
2. Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
3. Add this document to your repo. Make sure it’s named “**Capstone\_Stage1.pdf**”

---

[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 and Fragment](#)

[Task 3: Implement Google Maps](#)

[Task 4: Implement App Indexing](#)

[Task 5: Fetch Data](#)

[Task 6: Janus Library adaptation](#)

**GitHub Username:** oscarbujinkan

# event2Stream

## Description

Event2Stream is a platform where you can stream your own events. This app has the goal to inform the participants about the event and make they more aware about what kind of topics are in discussion.

The app has a list of events and you can see them, one by one with the ability to go back in the time line and repeat some important message that you miss.

## Intended User

Attendant to the event or any person interested in the topics to discuss.

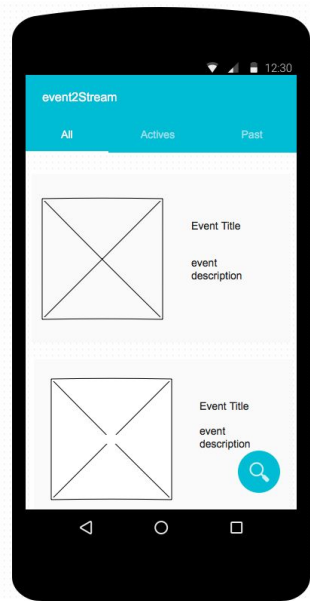
## Features

- List and search events hosted by the platform
- Play event stream
- Go back in the stream.
- Share moment in the stream.
- Show highlights by the editor
- List the exhibitors of the event and information about them
- Show with google maps the location where it will take place the event.
- List the event schedule.
- Play a talk from the schedule list
- Reminder for a talk listed in the schedule.

## User Interface Mocks

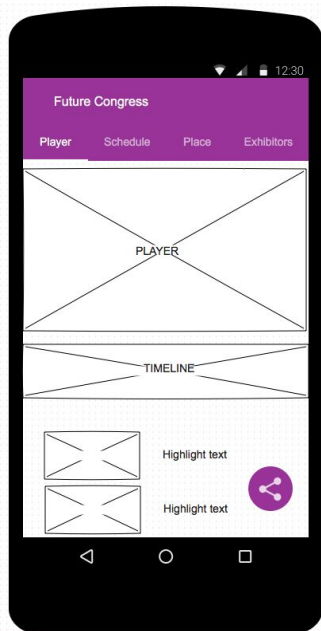
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

## Screen 1



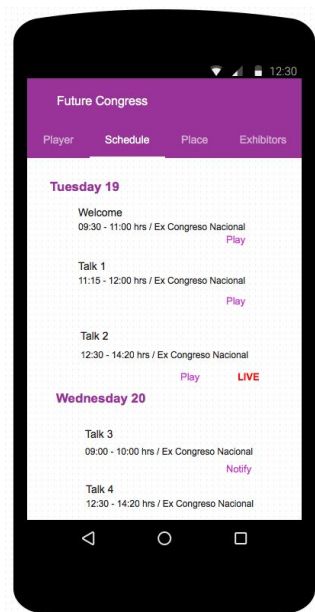
Event List: This view list the events available. We can search by event name or filter by “all events”, “active events” or “past events”).

## Screen 2



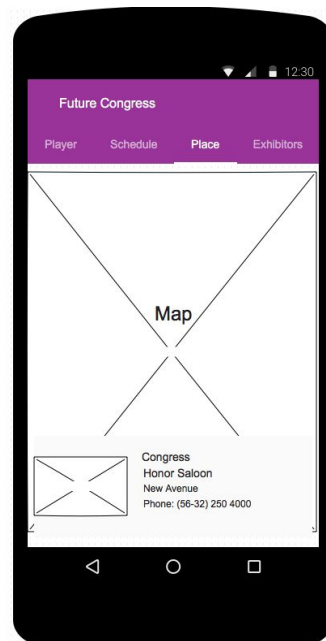
Event Player: This view has the player capable to get back in the stream using janus technology.

## Screen 3



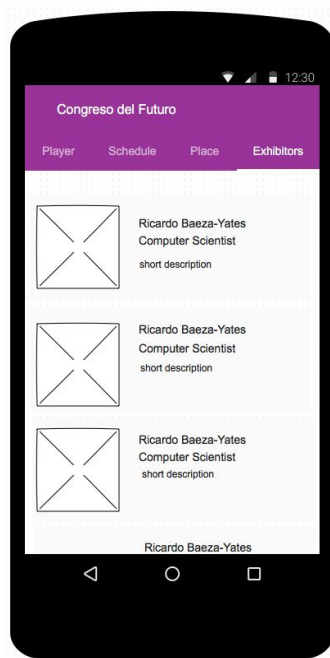
**Schedule:** This view has the schedule of the event and you can put an alarm that notify when the talk begins. If the talk has already been dictated you can go to that specific moment and re-watch it.

## Screen 4



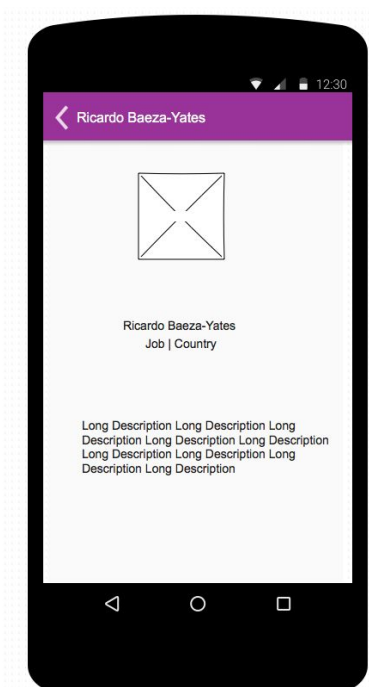
**Place:** This view has a google map instance with a pin in the place where the event is taking place.

## Screen 5



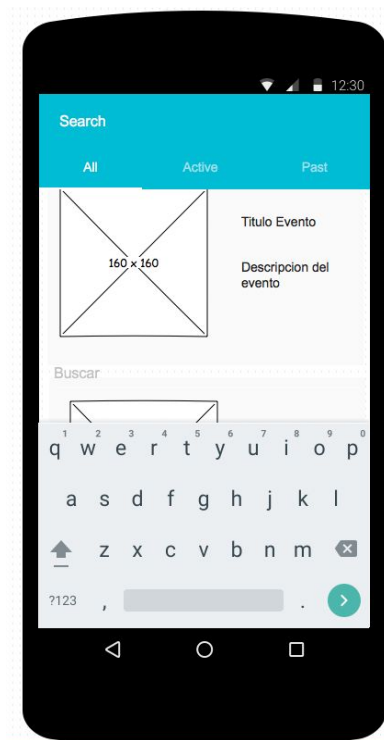
Exhibitors: This view has the list of the exhibitors that will talk in the event

## Screen 6



Exhibitor detail: This view has detail information about the exhibitor.

## Screen 7



Search: This View is used to search for an event.

## Key Considerations

**How will your app handle data persistence?**

The persistent code between the app will be handle with content provider and sharepreference.

**Describe any corner cases in the UX.**

The application will use the fragment back stack for navigate between the app. You can always come to the list of event.

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

- Picasso for caching images.
- Janus for implementing the player and play streaming

- android-shape-imageview: A github library that is use for making circular image in the app.

**Describe how you will implement Google Play Services.**

Google maps: For display the location of the event.

App Indexing: For indexing the app information, so we can be able to find talks in the device search tool.

## Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

### Task 1: Project Setup

- Create project in Android Studio.
- Configure Janus librarie
- Build the architecture of the app

### Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for for Event Activity
- Build UI for Player Fragment
- Build UI for Schedule Fragment
- Build UI for Location Fragment
- Build UI for expositor Fragment
- Build UI for expositor detail fragment

### Task 3: Implement Google Maps

Use the google maps api for display the location of the event.

- Fetch data from backend
- Use api for displaying the position
- Display additional information

## Task 4: Implement App Indexing

- Initialize Api
- Choose some events from the user
- Use API for index them for future search.

## Task 5: Fetch Data

All the data that the app shows is provided by an online platform. For display this, we need fetch and parse the data.

- Fetch and parse exhibitors
- Fetch and parse the location
- Fetch and parse the schedule
- Fetch and parse the event stream configuration

## Task 6: Janus Library adaptation

We need to adapt the Janus Library in order to support diferents stream configurations.

- Janus basic configuration
- Dynamically send the stream configuration to Janus
- Implement Callback from the library

Add as many tasks as you need to complete your app.

---

### Submission Instructions

1. After you've completed all the sections, download this document as a PDF [ File → Download as PDF ]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone\_Stage1.pdf**"



