



# SOFTWARE ENGINEER TEST PROJECT

## Mock Web Application for managing advertising creatives on Moloco

1. Purpose.....	1
2. Time.....	1
3. Scope.....	1
4. Industry concepts.....	2
5. Specifications.....	2
5.1. Functional Specs.....	2
5.2. Web App Specs.....	3
5.3. Automated Test Specs.....	3
6. Evaluation Criteria.....	3
7. Deliverables.....	3
8. Project communication.....	3

### 1. Purpose

This project is meant for Bubbleye to assess your ability to **familiarize** with a range of new concepts, both functional and technical, to design and develop a web application fitting a customer's requirements. We are looking to get a sense of your technical skills as well as your ability to solve problems you haven't faced before.

### 2. Time

We hope the entire development will take less than 8 hours (though it is understood, candidates will be able to dedicate only fragmented portions of time to this project, so we allow up to one week of elapsed time to deliver the final application).

While we would like you to deliver **all** the specs listed below, depending on your experience and available time we understand you might not be able to do so; in such case, please use your critical sense to autonomously identify the highest priority specs and complete at least those. When you deliver your solution, you may then accompany your code with a few notes describing **what** could not be completed, **why** not, and **how** you would implement those missing features.

### 3. Scope

This test project consists of 2 components:

- 1) Build a program which has an user interface, these are what we recommend but not limited to
  - a) Python + [Streamlit](#)
  - b) Web frontend + backend in any programming language (Python is preferred)
  - c) Python + Google Spreadsheet



- 2) In the program, you may build **mock API** that simulates requests and responses to or from [Moloco](#) based on the [API document](#)

## 4. Industry concepts

1. "Advertiser" is any product maker who pays a certain amount of money to advertise their products (physical products or software products)
2. "Publisher" is any product maker who earns money by showing Advertisers' ads to its own users
3. "Campaigns" are advertising projects initiated by each advertiser to promote their products. For example, each "campaign" may be addressed to a specific audience, or a specific geographic targeting.
4. "Creatives" (also called "Ads") are the advertising content - a video, or a banner, or an interactive ad - which the Advertiser aims to show to Publishers' users, in order to promote its product.
5. "Advertising Platform" is a company that connects Advertisers with Publishers: it runs the Advertisers' campaigns by showing their Creatives to a broad audience of Publishers' users. Each advertising platform earns money through arbitration - ie. it makes a margin between the fee paid by the Advertiser for being promoted and the reward paid out to the Publisher for having the ads displayed to its users.
6. [Moloco](#) is an Advertising Platform. Technically, among the many types of advertising platforms on the market, Moloco is defined as a programmatic DSP. DSPs' effectiveness depends on their ability to place each ad on the best suited mobile phones and most attractive ad placements within each mobile phone, so that a small number of "ad impressions" may lead to a high number of conversions.
7. [Moloco's campaign structure](#) is very peculiar. For example, within each campaign, Moloco's platform distinguishes between ad groups, creative groups, and creatives. Without digging deeper into the reason for such entity structure, please familiarize with their campaign entities through their [official API documentation](#): this will be essential in order to build a web app that satisfies the needs listed in section [5. Specifications](#).

## 5. Specifications

### 5.1. Use CASE

- Imagine you are an Advertiser: you have built a mobile app for Android devices and you want to promote it through ads on the Moloco platform
- Every week, your team prepares 1 new "creative concept", and implements it in 2 new files. You constantly aim to find a new creative concept that may deliver more conversions, so that you may obtain more results from your advertising spend.
  - a. A creative concept is consist of:
    - Portrait video
    - Landscape video
  - b. A creative concept will then be used to create 1 creative group in Moloco, which contains 2 *VIDEO* (Moloco creative type) creatives, here are some information for the API call
    - Creative type: *VIDEO*
    - Video property: *auto\_endcard=true*
  - c. It takes some time (depending on the size of each video) to upload an asset in Moloco, you want to upload them (and/or create a creative) as soon as you receive a concept.
- Every 2 week you have 2 new concepts, you want to do the following:
  - a. Create every new creative concept to Moloco, each of them becomes 1 creative group with 2 creatives



- Creative\_group\_1
  - Creative\_1: portrait\_video\_a
  - Creative\_2: landscape\_video\_a
- Creative\_group\_2
  - Creative\_3: portrait\_video\_b
  - Creative\_4: landscape\_video\_b
- b. Attach them to a Moloco creative testing campaign (assume its name is: creative\_testing\_campaign)
- c. Attach another creative group, which has been tested and shows high performance, as control group
  - Assume the creative group is: good\_creative\_group
- d. Start the campaign, Moloco will make sure every creative group receives 10,000 impressions, then pause the campaign.
- e. You then evaluate the tested concepts, champions (good performing ones compared to the control group) will be used for the regular campaigns.
- Every month, you evaluate the regular campaign's performance: for each of them, you replace the worst concept with one of the champion concepts
  - a. Assume there are 2 regular campaigns
    - regular\_campaign\_a
    - regular\_campaign\_b
  - b. There could be more new champion concepts than the available slots in the regular campaigns, you want to put them in waiting lines.

## 5.2. PROGRAM SPECS

- Feel free to design the User Interface however you find effective and meaningful: users need to be enabled to take the consecutive steps described in section [5.1. Use Case](#), but the look of the interface is completely up to your creativity and common sense. Don't spend too much time making it pretty: **rather focus on making it useful.**
- Please implement a **mock API server**: don't hardcode the json response in the frontend.
- You're not required to store data in a file or database, but that's allowed if you prefer to do so: **what matters is that the user's intention is saved.**

## 6. Evaluation Criteria

We will evaluate several aspects of the final deliverables:

1. Effectiveness: can it meet the specifications?
2. User Flow: Is the user flow intuitive and convenient?
3. Elegance of the solution: use as few libraries and plugins as possible, to deliver an application that is simple and lightweight
4. Code quality: coding standards, informative in-line comments
5. Overall thoughtfulness of the user interface: while prettiness is not important, the UI must be clear, sleek and easy to use
6. Self management:
  - a. If you can't complete all of the tasks, try to autonomously prioritize all tasks based on your understanding of our purpose (we want to evaluate your technical skills and ability to use a diverse range of tools and frameworks - see [1 Purpose](#)) so you can at least deliver those tasks that you feel are more important, and leave the nice-to-have enhancements as lower priority



- b. If after proper investigation/research you conclude certain requested features are not technically feasible, you may still build your best alternative solution to address the same underlying business need as you perceive it

## 7. Deliverables

1. A zipped File containing all the Files necessary to run the application locally, on our computers
2. Instructions on how to run the application
3. If you can't complete all tasks, you may accompany your code with a few notes describing what part you could not complete, why not, and how you would implement those missing features if you had more time.

## 8. Project communication

Feel free to request further information by [Email](#), if needed.

**Good luck!**