Phineas Software Design Document

Steeve Joseph

7 December 2019

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Purpose

Currently, with creative fields being as saturated as they are, it is difficult for a newcomer into the "service arts", such as photography, commision painting, and singing, to find clientele. The typical creative has difficulty marketing their services, and this is likely because their services are not being marketed "properly". In agile software development, there is an emphasis on finding a demographic of people to solve a problem for, and then developing a product that acts as a solution to that problem. With this methodology, the organization developing the product has sought out and matched their target clientele.

However, in more "creative-oriented" fields, this does not seem to be the representative approach. The default approach from the creative seems to be to cast a wide net so to speak, and hopefully net some buyers.

If it were possible to filter/pre-select potential clients on some objective and/or measurable metric, the artist would presumably have higher success pitching themselves the clients with whom they "match".

1.2 Scope

The scope of this project thus far is to be used as a photographer looking for likely candidates for clientele, using social media. Specifically, the product would allow a photographer to single out people that are interested in the genres that the photographer is seeking to shoot, based on hashtags, thereby facilitating the filtering process for the photographer.

1.3 Overview

A representative use-case for the product is given below:

1. The user signs into Phineaswith their Instagram account.
2. The user searches hashtags matching what they want to take pictures of.
3. The user gets a curated list of potential clients that they can then contact.

1.4 Reference Material

1.5 Definitions and Acronyms

Term	Definition
IG	Instagram
FB	Facebook
IGAPI	Instagram Platform API
FBGA	Facebook Graph API
FBBDA	Facebook Basic Display API
MVP	Minimum Viable Product

Table 1.1: Table of Definitions and acronyms used

Chapter 2

System Overview

Chapter 3

System Architecture

3.1 Design Constraints

3.1.1 Instragram Platform API

Deprecation

As originally anticipated, the Basic display portion of the IG Platform API was set to be deprecated by early 2020, accelerating the timeline for this project significantly. Essentially, the intent was to get a working prototype/ minimum viable product (MVP) released before deprecation. Of course, deprecation would still have to be accounted for, but the benefits of the MVP could have been reaped in the meantime. A potentially unaccounted for change is the effect of the deprecation of the Public Content endpoints on the ability to query for images based on hashtags. This is mentioned because it is a critical feature of Phineas.

Discontinuation of New Client Registration

An unanticipated hiccup was the discontinuation of new client registrations by the IG Platform, essentially forcing the use of the Instagram Graph API. This was retroactively discovered after the discontinuation in October 15, 2019.

3.1.2 Instagram Graph API

The Instagram Graph API is the successor to the Instagram Platform API (sometimes referred to as the Legacy API) Presumably, this API implements GraphQL in order to limit the amount of data sent with each request.

The benefit of this is better control of privacy issues across the platform. The potential drawback is that the Graph API may not have the full functionality that the IG Platform API had, for good reason as well. Facebook seems to be moving in a direction wherein the level of user information is limited, most likely to mitigate potential harm from bad actors.

Limitations

The critical functionality of Phineasis using Instagram's API to gather a set of pictures based on hashtags, and filter for users that meet a certain set of requirements. These requirements are predicated on metrics such as: user follower count, user following count, likes on said picture, comments on said picture, etc. It seems that the Graph API is solely concerned with business accounts, as of December 13, 2019, and that regular personal accounts do not have public API endpoints exposed. The following are points of interest in the new API.

IG User Endpoints

Represents an Instagram Business Account or an Instagram Creator Account. Some relevant returnable fields from this endpoint are:

- biography

- followers_count
- follows_count
- name
- username
- website

Relevant edges are:

- Business Discovery: Allows you to get data about other Instagram Business or Creator IG Users.
- Insights: Represents social interaction metrics on an IG User.
- Media
- Recently Searched Hashtags

IG User: Insights

IG Media

Represents and Instagram photo, video, story, or album.
Relevant returnable fields from this object are:

- caption
- children (if IG album)
- comments
- comments_count
- like_count
- permalink
- username
- media_type
- timestamp

IG Media: Insights

Represents social interaction metrics on an IG Media object. Relevant metrics include

- Photo and Video Metrics
  - engagement: Total number of likes and IG Comments on the album IG Media object.
  - impressions: Total number of times the album IG Media object has been seen.
  - reach: Total number of unique Instagram accounts that have seen the album IG Media object.
  - saved: Total number of unique Instagram accounts that have saved the album IG Media object.
  - video_views
- Album Metrics
  - carousel_album_engagement: Total number of likes and IG Comments on the album IG Media object.
  - carousel_album_impressions: Total number of times the album IG Media object has been seen.

  - carousel_album_reach: Total number of unique Instagram accounts that have seen the album IG Media object.
  - carousel_album_saved: Total number of unique Instagram accounts that have saved the album IG Media object.

- Story Metrics
  - exits: Number of times someone exited the story IG Media object.
  - impressions: Total number of times the story IG Media object has been seen.
  - reach: Total number of unique Instagram accounts that have seen the story IG Media object.
  - replies: Total number of replies (IG Comments) on the story IG Media object.
  - taps_forward: Total number of taps to see this story IG Media object's next photo or video.
  - taps_back: Total number of taps to see this story IG Media object's previous photo or video.

IG Hashtag

Represents an Instagram hashtag. Relevant edges from this node are

- **Recent Media** Returns a collection of media recently tagged with this hashtag.
- **Top Media** Returns a collection of the most popular media tagged with this hashtag.

IG Hashtag: Recent Media

Represents a collection of the most recently published photo and video IG Media objects that have been tagged with a hashtag.

Comments

Represents a collection of IG Comments on an IG Media object. Not readily apparent what value these have, may be possible to perform aggregate sentiment analysis on an IG Media Object's comments.

3.1.3 Instagram Basic Display API

The Instagram Basic Display API seems to replicate the functionality of the Legacy API's Basic panel, which allowed the reading a user's profile info and media. While there is a plethora of useful information accessable, some core components that seem to be missing are:

- user follower count
- user following count
- user engagement metrics

The lack of the above fields indicate Instagram's focus on privacy and security for regular accounts. As a result, Phineaswill be primarily focused on Creator and Business Instagram accounts, which have more information readilt available.

Limitations

3.2 Prior Attempts

This section focuses on prior attempts at solving the problem mentioned in the Introduction.

3.3 Similar Software

To date, (December 13, 2019), there does not seem to be any similar products or services in this space. Specifically, many of the applications using Instagram's API are using the Legacy version, either from presumable reluctance to migrate, or due to perceived lack of functionality in the Graph and Basic Display APIs.

Chapter 4

Data Design

4.1 Database Design

MongoDB will be used for Phineas, using MongoDB Atlas as the database provider.

The databse currently looks like this:

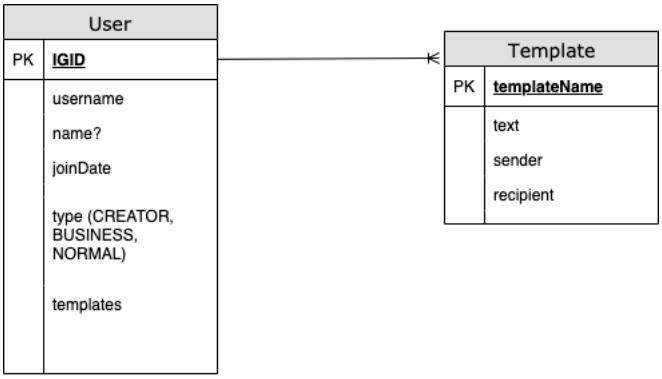

Figure 4.1: Phind ERD

4.2 User Model

The User model will look like this:

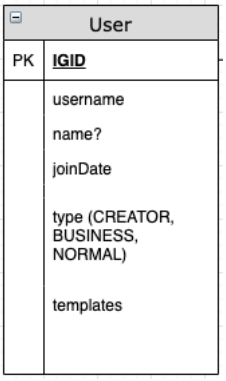

Figure 4.2: Phind user model

4.3 Data Interaction

Phineaswill read and write to a MongoDB database via an API written in ExpressJS. The API will communicate with a ReactJS frontend.

Chapter 5

Component Design

5.1 API

This section covers implementation details for the API of Phineas.

5.1.1 Backend Language

ExpressJS was chosen to be the backend language of choice for Phineas. The reasons why are numerous:

- written in JS
- simple, flexible, extensible
- "One language to rule them all"

5.1.2 Routes

At the very least, the necessary routes are:

- login: goes through Instagram's auth flow
- query: takes in a query object, returns a list of users

5.2 User Interface

This section covers implementation details of the UI portion of Phineas. Phineaswill primarily be web-based. ReactJS will be used as the UI framework. Rationale:

- It is the framework I know best
- Fits into "one language to rule them all" paradigm

Chapter 6

User Interface Design

6.1 UI Overview

6.2 Screenshots

6.3 Screen Objects and Actions

Chapter 7

Requirements

Chapter 8

Milestones

Chapter 9

Testing

9.1 API Testing

9.1.1 Postman

Collections

A collection is used to run multiple requests (and their coresponding tests) simultaneouslty

Tests

In addition, fields from the response of one request can be used in another request via chaining.

9.2 GUI Testing

9.2.1 Selenium

9.2.2 Squish Tools