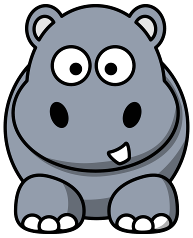

Hippo

finding business ideas using AI and big data analysis

Requirements Document

Document revision 2.0

27 February 2018

Client

Sjoerd Bekius (Juicy Story)

Teaching Assistant

Hichem Bouakaz

Team

Levi van Rheenen
Jean Paul Donovan Meijer
Said Faroghi
Natalia Karpova
Thijs Baksteen
Andreea Glavan
Sardor Khashimov

Table of contents

Introduction

The ability to find market-validated business ideas is an issue many entrepreneurs are faced with. Even if a certain idea they come up with is seemingly good, proving its' relevance to the market can be difficult. What is more, it is often difficult to put a number to its' demand. For example, finding a group within the target market who all have a necessity for the specific product/service pitched is not only time consuming, but also may not be enough to prove this necessity to others.

Hippo is a web app which allows entrepreneurs to not only find products the general public wishes for, but also prove its' relevance and the demand for it. The collecting of ideas is done through social media crawling, Twitter specifically. Twitter is a great platform to collect potential business ideas from as people can voice their wishes as well as engage with other ideas they find interesting or useful. Given the structure of Twitter and likeliness of people to share ideas on this platform, it is possible to show the demand for each idea/wish by counting its' interactions, such as likes, retweets, and replies.

The goal of Hippo is to provide entrepreneurs a platform of possible business ideas collected from Twitter, as well as the interaction data for each idea. As the target users are entrepreneurs looking for innovative business ideas, we aim to make an easy to use system which presents all the found ideas and their metadata in a understandable manner. The platform itself is to offer the ideas as well as packages including the interested users' names, and the idea's statistics(likes, retweets) which can be purchased by interested entrepreneurs.

Functional Requirements

Critical

Retrieve tweets from twitter.

Using proper authentication to Twitter's API, the software should be able to automatically crawl through tweets and retrieve them, for the duration of the application's lifetime after launch.

Filter tweets based on certain terms.

In order for tweets to be extracted, they must be queried with keywords that, if present in a tweet, makes that tweet have a high chance of containing a business idea. Because of Twitter's API being rate-limited, the choice of keywords for the queries are important. The ratio of relevant business tweets to irrelevant tweets (contains no business idea) should also be maximized.

Group tweets together based on ideas.

Multiple tweets may contain the same idea but formulated differently. Using Natural Language Processing (NLP) techniques, the application should merge tweets with the same idea together

Cluster ideas together based on category.

After grouping tweets into ideas, the ideas themselves should be categorized as well. This will also use NLP techniques.

Store the tweets including interaction and metadata in a database.

The data is stored should also be able to be queried efficiently, so a good relational model must be potentially used so that searching the database is quick.

Access a data set consisting of tweets related to a certain idea.

The user should be able to easily navigate through tweets that belong to a certain idea, and be able to view the metadata of each tweet that a user would find useful, such as the contact information of the tweet’s poster. This data set is only available to paying customers.

Important

Implement user accounts and access controls.

All users should have personal account, and their access to data (tweets and associated metadata) should be controlled.

Implement idea package that gives an access to more the more detailed data set.

Application should have several example ideas that would be shown to a customer to provide an insight of what Hippo application sells. Majority of the ideas and associated metadata have to be stored in the idea packages, that are supposed to be purchased by a a client.

Explore all available ideas based on category.

There should be idea's category filters that will filter and show tweets based on (a) chosen category(s).

Useful

Idea sharing and comment functionality.

User should be able in some way to exchange ideas or comment in the application, may be related to an idea it's data set.

'Like' functionality.

There should be a 'like' button for each idea, similar to the functionality many modern social networks have. These 'like' stats can also be added up to the total stats(number of retweets, etc.) of the idea.

Removing ideas.

If ideas are not searched for or interacted with for a certain period of time, they should be removed from Hippo. The period of time is yet to be determined.

Non-Functional Requirements

Minimal maintainability

The maintenance required when the application is deployed should be minimal. Daily operations should only require minimal interference. Software upgrades of the underlying platform (possibly database, containers) on which the application runs should be a simple task that can be performed by a system administrator. Thus, not often require interference from the development team in the sense of major adjustments to the codebase.

Data protection law conformity

We should ensure that we satisfy any obligations under the EU General Data Protection Regulation (GDPR). Since it's very likely that in some way we will process personal data, although it is public information from Twitter. Consequently, the question that arises is whether we need to satisfy the rules about personal data defined in the GDPR when processing and storing it.

Scalability

The application should be able to handle large streams of data from Twitter streaming API of which the volume is still unknown. This will be a real-time feed that needs to be processed almost instantaneously if not the feed will be closed by Twitter. Furthermore, the entire application should also be easy to scale in the future. By taking these things into account we can prevent future issues that might arise related to this.

Flexible deployment

Deploying the application it's components should have some kind of flexibility such as the usage of containers (Docker, Kubernetes). This partially ensures the required future scalability and maintainability.

Extensibility

The project should be engineered in such a way that takes into mind any future development, i.e. adding features. An modular design, consisting out of loosely coupled components that can potentially be reused in new functionality.

Excluded Requirements

Use any other social media platform than Twitter.

Twitter will be the only platform for which we will collect the data. Platforms like Facebook, or any other platforms will not be used.

Meeting Log

The records of all client meetings with the decisions both parties agreed on.

Date	Description
23-02-2018	- Discussed concepts such as business ideas and relevant categories. - Introduced paid packages, and determined they should include usernames and idea backlog. - Decision to only crawl Twitter for ideas, due to it being ideal for idea sharing. - Established web app end goal. - Focus on client interaction and feedback.
05-03-2018	- Agreed on continuous integration and code coverage target 85-95%. - Introduced the ‘like’ functionality. - Introduced ideas being removed if not interacted with. - Set block based goals: 1 - working app, 2 - demographics. - Delimited relevant demographics and discussed the possible privacy issues.

Change Log

The record of the changes made to the requirements document tagged with date and author.

Author	Date	Description
Jean P.D. Meijer	21-02-2018	Create the requirement document and layout the structure.
Andreea Glavan	21-02-2018	Wrote an introduction for the requirement document.
Jean P.D. Meijer	25-02-2018	Added a meeting log and change log to the requirements document.
Andreea Glavan	25-02-2018	Updated meeting log and introduction after first client meeting
Jean P.D. Meijer	26-02-2018	Listed most of the functional and non-functional requirements.
Said Faroghi	26-02-2018	Added descriptions to the critical functional requirements section.
Sardor Khashimov	26-02-2018	Worked on non-functional requirements.
Natalia Karpova	27-02-2018	Added descriptions to the important functional requirements section.
Levi van Rheenen	27-02-2018	Added excluded requirements.
Jean P.D. Meijer	27-02-2018	Prepared the document for first release, elaborated the non-functional requirements.
Andreea Glavan	10-03-2918	Added new requirements and meeting log after client meeting in the second sprint.