Software Engineering Product Requirements Version 2

Juicy Story Project Group
Hichem Bouakaz (TA)

Tolga Parlan
S3051072

George Argyrousis
S3175243

Alina Matei
S3080722

Casper Schmit
S3194299

Michiel de Jong
S2550768

George Rakshiev
S3015564

Wim de Jager
S3215210

March 12, 2018

Introduction

The Juicy Story project is aimed at companies to facilitate their use of Instagram: creating user-generated stories, getting statistics, scheduling the upload of Instagram posts. This way, each company could benefit from the content posted by their customers which would be helpful for marketing departments worldwide.

Most companies use Instagram as a means of mass market penetration. That of course means that their content needs to be curated and meticulously managed. In a more recent update, Instagram introduced stories as a means of interaction with their target audience in a more meaningful way. Company held events require coverage from multiple angles requiring multiple employees. Company representatives and media curators consider that to be a viable option, although optimal results are not guaranteed.

In this report, we will present the main feature, other secondary requirements, a meeting log and the decisions made with the client.

Users and their stories

Main Feature(User Generated Stories)

Conventionally, a company needs to create their own marketing or coverage for events and products. In the rise of social media a lot of the marketing is being done via applications like Instagram. Companies do not have a way of tapping into user content through the Instagram application that might be relevant for marketing. With Juicy story, the main focus is to provide a solution to this problem.

A company would be able to login to our web application and link it with their Instagram accounts. From there they can search for relevant pictures via hash-tags or locations. When the user finds content that is relevant, for example a photo from a person who attended one of their events, the user should be able to select that photo, edit the photo and then queue that photo for upload onto their instagram story.

Search

After the user is authenticated in our application, they should be able to search for specific content in Instagram through a search bar in our application. In the search bar hash-tags or locations can be inputted and the output will be a configurable number of photos maximum per each page, with ability to continue to further pages.

Selection

After the images are the displayed on our web application a user has the option to select multiple pictures and click a submit button for whatever images they want to be added to the timeline. The user can also put other inputs into the search bar and add results from those searches to the same timeline or even deselect images from the timeline. The selection process occurs on the same web page as the search process so the user can instantly choose what photos they want to add to their timeline.

Edit

After the user is satisified with the selection of images they will be redirected to an edit window by clicking another button where they can edit each pictures individual links,description, and meta-data. They will also be able to rearrange the timeline on this window.

Upload

When the user is done editing their timeline they will be able to upload the individual pictures as they appear on the timeline to the story of the Instagram account they have selected by clicking the upload button found on the editing page of our web application. The user has the ability to preview the timeline they created before uploading the pictures to the profile's story.

Minor Features

Statistics

Diagrams providing useful information regarding to posts and stories. The information can extend to likes, comments and reposts containing @ prerequisites. Gather statistics about your posts. For example: who has seen your story, how many people have seen your story, at what time do people see your stories, how long have they looked at your story, etc.

Scheduled posts

The user would be able to have a preview feed in an Instagram format. Attributes such as descriptions, links and hashtags need to be included to help promote the post even further. Every post can then be submitted according to predefined scheduling, providing a degree of freedom and scalability to companies as never before.

Non functional requirements

Maintenance

The developer of the Juicy Story company (Sid) will have to maintain the application. Sid will have to be able to understand the application source code and perform updates. The code should be very modular, so in order to understand and maintain one part Sid will not have to understand and maintain all of the code. Also, the application will be sufficiently documented and will provide Sid with a clear and detailed overview.

Compliance

Due to legal issues, we may not use Instagram's native API for some of our operations such as Scheduled Posts. Therefore with the functionalities that we implement we should be wary of the legal issues that might arise and comply with the relevant rules when designing.

Reliability

Since the customers are going to be corporations who will be using the product for professional purposes, it is rather important that the product is reliable because failures and down times can have financial consequences for the customers. The up time of the application will be constant, during maintenance the app will be online as well.

Usability

Taking into consideration that the customers are going to be companies worldwide, we should also consider the human factor: the user interface should be easily usable by all people independent of their technological knowledge. So, the interface will be designed in such a way that it is intuitive and easily accessible.

Extensibility

The final product should allow further developments. Even though the main goal at the moment is to create a MVP, the client would like to further extend the product: adding features to the stories (polls, text, location), scheduling the uploading of stories, keeping the product up to date according to future Instagram updates etc.

Documentation

As the product aims for paying corporate customers who would likely want to use all the features flawlessly and to the fullest, a solid documentation of the features is probably necessary to make sure as the usage grows, new users can learn the features on their own and do not need extensive support.

Meeting Log

Date	Discussed
23.02.2018	- Clarified that use generated stories are the main selling point and have the highest priority. - Statistics and scheduled posts have secondary priority. - The product is to be a web app, with no aspirations to be a mobile app at the moment. - Start from scratch, the only thing already present is the demo of the website. - The product is geared towards companies. - Recommendations for the back end: PHP with laravel or Python with Django. - Recommendations for the front end: Vuejs or AngularJS. - No public API's to get the stories. - Stories are retrieved by hashtags or dm.
1.03.2018	- A legal grey area with regards to the APIs - Continuous integration - Virtualization - What languages to use? (Recommendations of the previous meeting discussed again.)

Change Log

Date	Done
23.02.2018	- Created introduction, minor features and main features.
12.03.2018	- Noted which users are involved with the app and what their roles are. - Updated user stories with respect to what is available in the app at the moment. - Looked into feedback and changed commented parts. - Rewrote the main functionality requirements