Software Engineering Product Requirements Version 3

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

April 16, 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 display pictures they have received in direct messages from other Instagram users. 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.

Retrieving images from DMs

After the user is authenticated in our application, they should be able to have access to a comprehensive overview of all images send to them by DMs. This overview contains two options: first option consists of displaying a list of other Instagram accounts that have recently sent DMs to the Juicy Story user, who is able to select any of the before mentioned accounts in order to see what images they have received from that specific Instagram user. The second option is to have a complete overview of all images received by the Juicy Story user via DMs, regardless of the Instagram user who sent the content. The images will be displayed in chronologically ordered grid with their corresponding date and sender.

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 time-line. The user can also deselect images from the time-line. The selection process occurs on the same web page as the page displaying the incoming pictures, so the user can instantly choose what photos they want to add to their time-line.

Edit

After the user is satisfied 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 time-line in this window.

Upload

When the user is done editing their time-line they will be able to upload the individual pictures as they appear on the time-line 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 time-line 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 will be related to likes, reach, taps back, completion rate, replies and followers.

Scheduled posts

The goal of this feature is to allow the Juicy Story user to schedule the post of their selected pictures. This would mean that every post can be submitted according to predefined scheduling, providing a greater degree of freedom and scalability to companies. This function is implemented in connection to the upload button, in such a way that the user can simply schedule a post or upload immediately from the same web page.

Non functional requirements

Maintenance

The Juicy Story developer will need to maintain the application in order to keep it function optimally. The application frameworks and source code should be understood by the Juicy Story developer, to easily achieve this all work during initial development will have to be clearly documented. Also, the application should be very modular this allows for easy adjustments to the application without having to change the entire source code.

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

It is crucial that the application limits its possible downtimes and failures, as it will be used by corporations who will need reliable access and face financial consequences due to inconsistencies in the app. Therefore, the up time of the application will be constant, also during maintenance the app will be online.

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 necessary to make sure, as the user base 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.)
14.03.2018	——-

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
16.04.2018	- Updated main feature in to correspond with newly received information from client. - Updated minor features.