

Requirements Document

Project Pijper Media

Version 2.0

Client: Kevin Klomp, Eelco Luurtsema, Sanne Vast



PUPER MEDIA
a family business

Team names:

Daniël Scheepstra

Julian Pasveer

Medhat Kandil

Dilan Adel

Jeroen Klooster

TA:

Alina Matei

Table of Contents

1 Introduction	3
2 Actors	4
3 Use Cases	5
Use Case 1 (requirement 1.1): Receive notification	5
Use Case 2 (requirement 1.2): Logging in	5
Use Case 3 (requirement 1.3): Navigate to source page	6
Use Case 4 (requirement 1.4): Viewing social media posts	7
Use Case 5 (requirement 2.3): Personalize dashboard	8
Use Case 6 (requirement 2.1): Check off post	9
4 Requirements	10
Must have	10
Should have	10
Could have	10
Won't have	11
Non-functional requirements	11
MVP Requirements	11
5 Meeting log	12
6 Change log	12

1 Introduction

This project has been introduced to us by the family business Pijper Media. Pijper Media is a family business located in Groningen. It functions as a printing company, a magazine publisher and a large online branch. Pijper Media employs a total of 100 people, with headquarters in Groningen and editorial offices in Amsterdam. It consists of Pijper Druk, Pijper Publishing and WeBLog media. This project will be created for WeBLog media. The editorial team of WeBLog media would like to produce content based on viral posts to uplift the brands they publish content for.

These days almost everyone is in some way involved in social media. It could either be by texting each other or posting pictures on Instagram. Certain topics or posts go viral, which means they get a huge amount of attention. Our client would like a quick and easy overview of such topics which are trending/viral. To do this we're going to build a social wall on which all posts will be shown and categorized for easy access. This social wall is going to be shown on a desktop and optionally on a (non-interactive) tv screen.

In this document we will use the term viral and trending, but these terms have a vague definition. To combat this we will define how we use these words. Viral is when something becomes popular in a really short timeframe (e.g. within 30 minutes). Trending is when something is popular over a longer period of time (e.g. 3 days).

2 Actors

- **Main user:** The main users of the application will be the editors that write articles for Pijper Media.
- **Secondary user:** The other users for the application are the admins.

3 Use Cases

In this section, we will elaborate on the must have requirements.

Use Case 1 (requirement 1.1): Receive notification

User Story

As an editor, I would like to receive a notification when a post gets above average attention.

Rationale/Context

The editors want to receive a notification in case a post gets above average attention. This could either be via a pop-up or an email. If the editor is logged in, the editor will get the notification via a pop-up. However, when the user is not using the system, the user will get the notification via an email.

Frequency of Occurrence

A few times per week. Depending on the amount of posts that are getting above average attention.

Primary Actor

The editorial team.

Preconditions

The User is a member of the editorial team and is authorized to use this use case.

Postconditions

The editor is notified via either a pop-up message or an email that a post is getting above average attention.

Main Success Scenario

1. The System checks if a post is getting above average attention.
2. The System checks whether a User is using the system or not.
3. The System sends a pop-up to the Users who are logged in and an email to the ones that are not.

Use Case 2 (requirement 1.2): Logging in

User Story

As an editor, I would like to log into my own account.

Rationale/Context

Editors need to log into their own account so they can add posts from the dashboard to their activity list.

Frequency of Occurrence

Every time the user opens the application.

Primary Actor

The editorial team.

Preconditions

The user is a member of the editorial team and has already registered themselves.

The credentials of the user stored inside the database.

The user is on the login page.

Postconditions

The main dashboard will load for the user to view. The dashboard is the view that every logged in user uses to view and save posts.

Main Success Scenario

1. The user is connected to the system.
2. The user enters their credentials in the login form.
3. The system validates the credentials.
4. The system determines the user's role.
5. The system loads the dashboard view.
6. The user gets redirected to the dashboard.

Alternate Scenario

The user enters unregistered credentials in the login form, so the system tells the user to do it again.

Use Case 3 (requirement 1.3): Navigate to source page

User Story

As an editor, I would like to be able to click on a post to go to the source page of the article, this way I can easily get more information about the article/post.

Rationale/Context

Before choosing an article to write about, the editor wants to get more information about it. If an article looks like something they would want to write about, they should be able to go to the source

page of the article for more information. This way they can find an article which they like, and they can write about it.

Frequency of Occurrence

Every time a user wants to (possibly) write about an article, during the working hours of the editors this will be about 10 times (for selecting a good article) per 20 minutes for every editor at work.

Primary Actor

The editor/writer

Preconditions

User is logged in, and is on the home page.

There exists a source page to the article in the database.

Postconditions

The user is sent to the source page of the article.

Main Success Scenario

1. The system gives the user a “social wall” of articles from which he can choose one.
2. The user indicates to the system the wish to see more information about a certain article.
3. The system sends the user to the source page of that article.

Use Case 4 (requirement 1.4): Viewing social media posts

User Story

As an editor, I would like to see the timeline posts from the accounts on the social media platforms that are registered in the system.

Rationale/Context

On the dashboard the posts from the registered social media account have to be shown. Every post contains a title, image, caption, social media logo, a button that redirects the user to the source page, a button that adds the post to the user’s activity list and the date of when the post was added.

Frequency of Occurrence

Every time the user is on the dashboard page.

Primary Actor

The editorial team.

Preconditions

The user is part of the editorial team and is on the dashboard page.

Postconditions

The user seeing the posts on the page.

Main Success Scenario

7. The user is connected to the system.
8. The user is on the dashboard page.
9. The system fetches the posts saved in the database.
10. The system puts the data in the view.
11. The system loads the view.
12. The user sees the posts on the dashboard page.

Use Case 5 (requirement 2.3): Personalize dashboard

User Story

As an editor, I would like to be able to change the categories that are displayed to me on the social wall.

Rationale/Context

Articles/posts can be sorted into different categories. Editors have different preferences as to what topics they like writing about. To show the most relevant articles we want them to be able to choose the categories that are shown to them.

Frequency of Occurrence

Once at the registering process and later it will probably be once every week.

Primary Actor

The editor.

Preconditions

The user is part of the editorial team and is on the main page.

Postconditions

The preferred categories of the user are stored in the database categories.

Main Success Scenario

1. User indicates that he/she wants to change his preferences
2. The system gives the user a list of categories with checkboxes
3. The user checks the boxes of the categories he/she wants to see and unchecks the others.
After which he/she presses submit.
4. The system will store these preferences in the database.

Use Case 6 (requirement 2.1): Check off post

User Story

As an editor, I would like to be able to check off a post in order to avoid duplication in the stories which are being written about.

Rationale/Context

The feed contains a lot of posts. These posts need to have an option to be removed from the feed once an editor decides to write about it. This is in order to make sure no article is being written about more than once.

Frequency of Occurrence

Several times a day

Primary Actor

Editors

Preconditions

The user is a member of the editorial team.

Postconditions

The post is removed from the feed and added to the "written about" section.

Main Success Scenario

1. The User is on the homepage of the web app.
2. The User checks off the posts he/she wants to write about.
3. The System removes the post from the feed.

4 Requirements

Requirements listed in bold are requirements yet to be implemented.

1. Must have

- 1.1. As an editor I would like to receive a pop-up or an email notification when a post is going viral, so I will be one of the first people to write about it.
- 1.2. As an editor I would like to be able to log into my own account, so I will be able to personalize my own dashboard and be able to identify myself to the system.
- 1.3. As an editor I would like to be able to click on a post to go to the source page, so I can easily get more information about the article/post.
- 1.4. As an editor I would like to be able to see (article) posts from Facebook and Instagram, so I can see which topics are trending/going viral.
- 1.5. **As a user I would like to add/remove/edit specific Facebook, Twitter and Instagram sources and have the option to connect it to a category, which can be added/deleted/edited in a section of the admin tools.**

2. Should have

- 2.1. As an editor I would like to be able to check off a post I have written (or am writing) an article about, so there will be no duplication of work and this will make room for other posts to be shown.
- 2.2. **As an editor I would like to be able to remove some posts from my personalized dashboard if they are irrelevant, so I have a better chance of seeing relevant information for my work.**
- 2.3. As an editor I would like to be able to personalize my own dashboard, so I can look at the categories that are important for the work I deliver.
- 2.4. **As an editor I would like to be able to add and remove important (birth) days and events to the calendar.**

3. Could have

- 3.1. As an editor I would like to be able to search for a post/topic by name, so I can easily find the post I am looking for.

4. Won't have

- 4.1. As an editor I would like to be able to see (article) posts from Tiktok, Reddit, and Youtube, so I can see which topics are trending/going viral.

5. Non-functional requirements

- 5.1. The code should be implemented in such a way that it is easily modifiable, which is important for the algorithm that determines popularity/virality.

MVP Requirements

In our Minimum Viable Product (MVP), we will make sure that the following requirements are met. This is to make sure that the project is going towards the way the client wants.

- As an editor I would like to be able to log into my own account, so I will be able to personalize my own dashboard and be able to identify myself to the system.
- As an editor I would like to be able to see (article) posts from Facebook and Instagram, so I can see which topics are trending/going viral.
- As an editor I would like to be able to click on a post to go to the source page, so I can easily get more information about the article/post.

5 Meeting log

Date	Discussed
Wednesday 10-02	Kick-off & introduction
Tuesday 16-02	Functional Requirements
Friday 19-03	Technical Meeting
Tuesday 23-03	Live Demo and Question/Answer session
Tuesday 01-06	Live Demo and Question/Answer session

6 Change log

Date	Changes
Friday 05-03	Cleared up points said in feedback.
Sunday 07-03	Specify which requirements are going to be featured in the MVP.
Sunday 07-03	Continue: Clear up points said in feedback.
Saturday 27-03	Change some keywords according to the meetings and put the unimplemented requirements in bold.
Tuesday 18-05	Added Use Cases. Cleared up points given in feedback.
Friday 21-05	Added Use Cases.

