Technical Report - **Product specification**

What's new?

Course:	IES - Introdução à Engenharia de Software
Date:	Aveiro, 04.01.2023
Students:	104171: Luís Carlos Afonso 112018: Zuzanna Sikorska 103696; Catarina Costa 103182; João Matos
Project abstract:	What's New? - A website that compiles and aggregates news from different sources like Reddit or Hacker News around the world. It is focused in every single user that wants to be well-informed about certain topics like (sports, beauty, games, etc).

Table of contents:

1 Introduction

2 Product concept
- Vision statement
- Personas
- Main scenarios

3 Architecture notebook
- Key requirements and constrains
- Architetural view
- Module interactions

4 Information perspetive

5 References and resources

1 Introduction

For our IES final project we had to propose, conceptualize, and implement a multi-layer, enterprise-class application. Our team decided to create a web application related to collecting and displaying personalized news. We visualize our app to be helpful for ordinary people, so they can save time on searching for interesting news. In what follows, we will describe the detailed vision of the application and the core architectural concepts.

2 Product concept

Vision statement

Our application will be used by ordinary people. No matter how old they are or what job they have or what they are interested in - because there is one thing that connects us against all differences - **the news**.

We always want to be informed about the current situations and latest trends in our world or our country. However, we have also to take into consideration that each person has different interests and priorities. That's why our application is gonna personalize the news for each and every single user, because everyone is important to us.

There are many news websites around the world, but ours takes a more important place in the user's needs. We know that the whole process of finding suitable information can be too demanding and exhausting for many people, so our goal is to make it as simple as possible. Because you deserve to be well informed and not to spend time on scrolling the pages with improper and completely useless information.

We want to make your life easier and we are willing to do the worst part of finding the news from different sites and prepare the feedback with the results, just for you.

Personas

Julia is a 27 years old Polish emigrant. After finishing medical studies in the capital city of Poland, she decided to leave her country in order to find a better paid job. She moved to Portugal and found employment as an assistant of a main surgeon, in a private hospital located in Porto. She is not married and she doesn't have time to go on dates, her only company is an old cat.

Motivation: Julia feels a bit lonely sometimes, because she lives far away from her family and friends. But she still would like to be up-to-date with everything that's happening in her

mother country. She doesn't have time to search on her own for the hottest information from her homeland.

Kylie is a 22 year old beauty influencer that likes to read beauty news everyday during breakfast. She makes stories on her instagram account about the hottest beauty trends and cosmetics. Her account has more than 100,000 followers and each of her stories is viewed by the average of 250,000 users.

Motivation: She lives from being an influencer so she needs to be well-informed of the latest beauty products and trends, because she cannot fail her followers.

Albert is a 35 year old football fan who works as a storekeeper. He doesn't have children nor a wife, he shares a flat with two other of his friends, who also share the same interest.

Motivation: After his work Albert loves to sit on the couch and watch football matches (especially from Champions League). Unfortunately sometimes he has to work in the evenings and a few times he missed an important match because of that. When he is not able to watch Champions League matches he would like to at least know everything about the missed ones. Besides sports Albert is also very interested in politics so in the same site he would also like to just change the news topic and search for politics.

Steve is a retired 75 years old teacher. He lives alone and spends a few hours per day on making discussions on forums. He doesn't have any other hobbies and conversations with other Internet users are his entertainment. Many websites blocked him because some other users didn't like his arguments and out of envy they reported him to the moderators.

Motivation: Steve is looking for a new news forum where he would be able to write comments to specific news and have discussions with other users in the comment section on the topic related to the specific news.

Main scenarios

- ➔ **Julia** called her mother and heard that there were big strikes on the streets. She wants to know more about the current situation and about the striker's demands. She was trying to search for that information on her own but she failed and only found fake news on some suspicious websites. Finally she opens the What's New? application and after log in she **filters the news by country** which results in displaying the newest information about the strikes.
- ➔ **Kylie** heard gossip from her friend about the new eyeshadow palette created by the Gucci brand. She knows that this topic can bring her thousands of new followers so she is very determined to make a reel about this topic before other influencers. She enters the What's New? application and **chooses the beauty category to see only news related to that topic**. In a short time she finds three articles about this eyeshadow palette on the list of beauty news. She makes a reel before the majority of influencers hear about that topic.
- ➔ **Albert** had a shift on Wednesday evening, exactly when there was a match of his favorite club Arsenal in the Champions League. He missed the match but when he came back home he opened the What's New? app and **wrote "Arsenal" in the search field**. After clicking the search button he got a list of news regarding the missed football match.
- ➔ **Steve** felt very lonely on Sunday after he saw happy families in the park. He felt like he had to speak with some other people but he was too afraid to ask some random people on the streets. He logged into the What's New? app and chose the first news that he saw on the website. He **clicked the button "join discussion" and wrote a comment** which became the start point of the conversation in the comment section. Steve felt less lonely after that online discussion with other users.

3 Architecture notebook

Key requirements and constraints

To easily scale the application we need to have decentralized services, meaning that each component will need to have a distinct and independent role in the final outcome.

This decentralized architecture does not need to conform to the same speed and responsiveness and, as such, the requirements dictate that the public facing components need to be as responsive as possible, while the non-public ones need to be as maintainable as possible, even if it means that it will not be as responsive.

Since this is a system that will use current and updated API's it does not need to conform to legacy systems. However, as it will be discussed below, the system needs to be easily modified if some API changes and/or the requirements change.

Architectural View

As shown in the ***Architectural View***, we decided to separate the Data Generation from Processing, because if something changes in the scrapping requirements we just need to change the Processing and if the external API's changes only the adaptation of the Data Generator will be needed. Meanwhile, as the processing can be quite resource intensive, it will make more sense to scale out just that portion and not the whole Data Generation and Processing part.

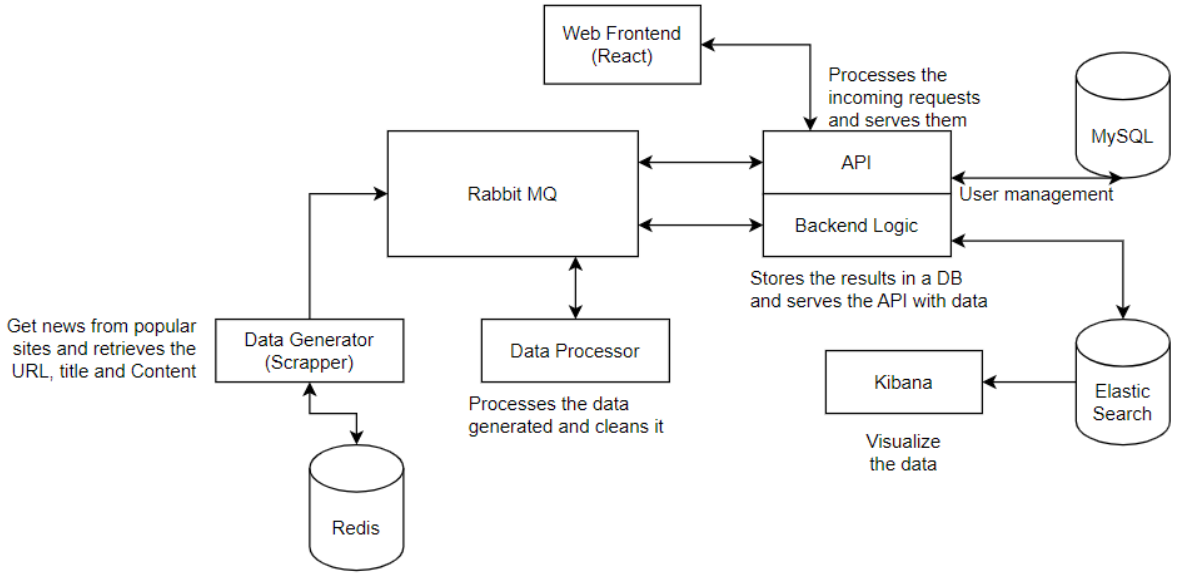
With that said, the application will run in a controlled environment and no private code needs to be exchanged with the public besides the generated *HTML*. As such, there will be no unusual conditions.

While our application depends on external API's, we will implement caching as a way to prevent that a failure in an API compromises the whole system. And as an added bonus, we can monitor which API's are up and which are down and inform the public of that.

While there is no Mobile Application planned, it can also be included if we see the need for it, since our API will already be working.

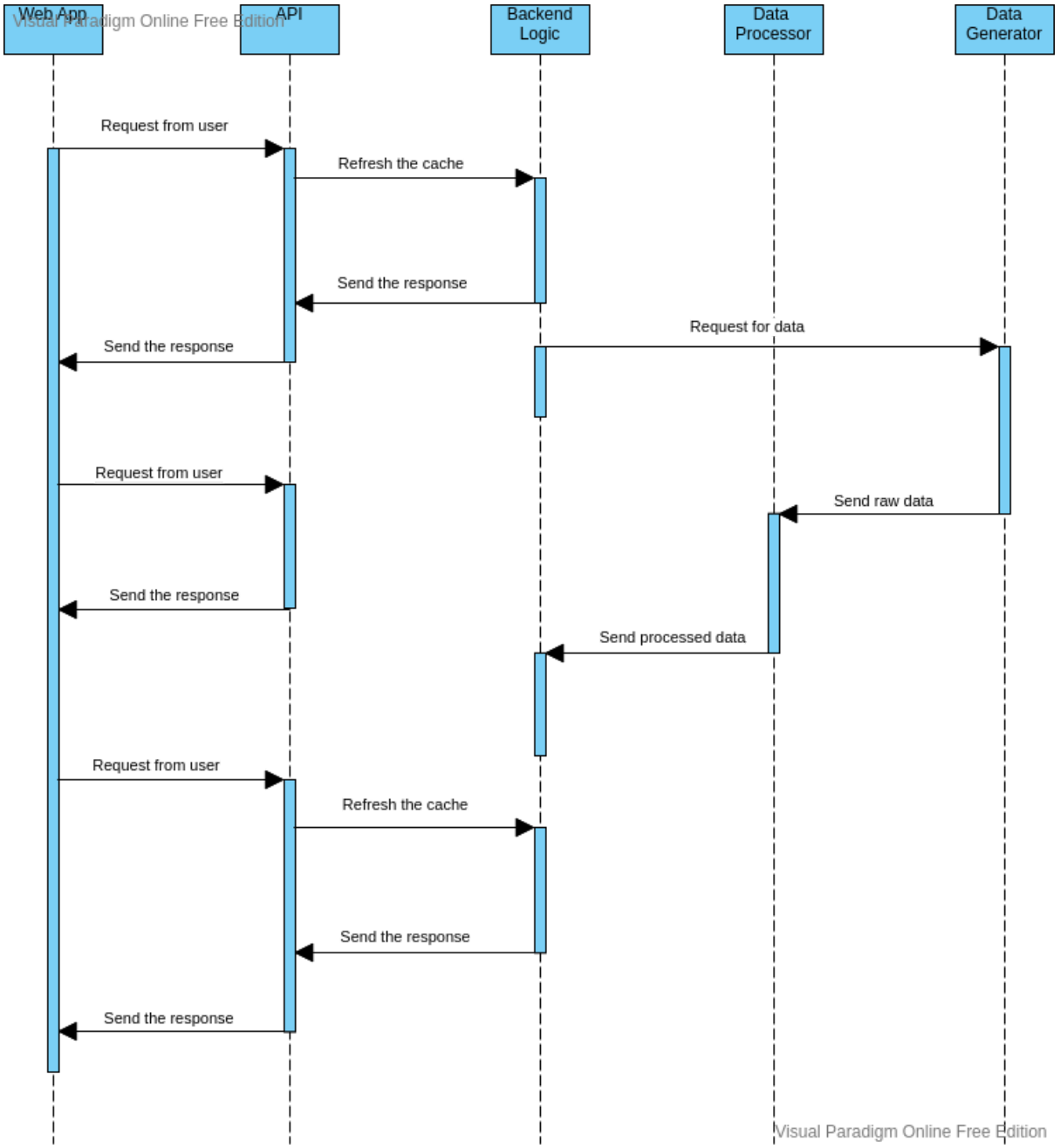
Regarding the retrieval of news, they will need to be searchable and easily available so it was made the decision to use *ElasticSearch*. To store the references to the already scrapped news we will be using Redis.

Architectural view



- **Data Generator** - This component has the simple objective of consuming and monitoring the external API's and notify **Backend Logic** in case of a failure or send the API response to the **Data Processor** in case of success.
- **Data Processor** - Processes, cleans and prepares data to be consumed in Backend Logic. Since the Data Processing takes much longer than fetching it and can be quite extensive, we separated it from the Data Generation. This also allows us to gain insights and personalize the processing of the data down the line by making use of several *RabbitMQ* queues.
- **Backend Logic** - Orchestrate the **Data Generator** requesting more or less news from a specified service and responding to requests in the **API**. This component also receives information in case of an external API malfunction, presenting that information to the *User* via our **API**.
- **API** - This component handles the *User Management* as well as querying **Backend Logic** for the most up-to-date news.
- **Web Frontend** - Handles the interaction with the *User*, forwarding the requests made by the *User* to the correct endpoints in our **API**.

Module interactions



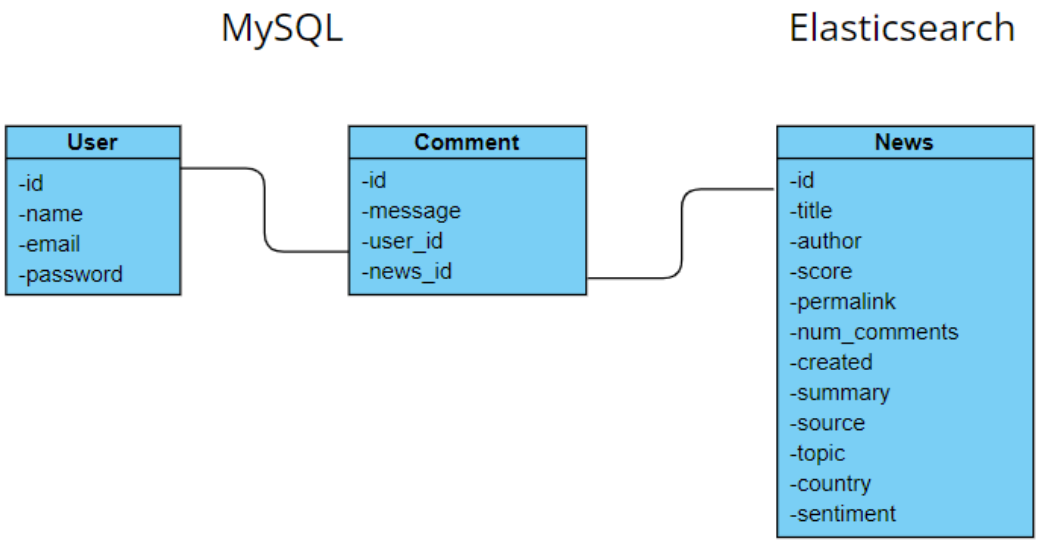
This diagram represents a normal workflow of the application in which 2 things can happen, either the **API**'s cache is empty or invalid (for example: X seconds passed since last refresh) and in that case a request is made to the **Backend Logic** to refresh the latest news or the **API**'s cache is valid and in that case no further propagation is needed.

Backend logic will monitor the requests made from **API** and decide if there is more news needed.

If there is the need for more news of a particular service, **Backend Logic** will request it to the **Data Generation** and receive either an error message or the processed news. In case of an error it will notify **API** that that particular service is down and in case of success, it will store in the *ElasticSearch* database the received news.

All these operations will generate logs that will be stored to be queried later by the product owners so they can make informed decisions about the system.

4 Information perspective



5 References and resources

https://www.elastic.co/community/

https://www.elastic.co/kibana/

https://spring.io/guides

https://developer.nytimes.com/docs/top-stories-product/1/routes/%7Bsection%7D.json/get

https://springdoc.org/v2/#migrating-from-springfox

http://deti-ies-11:8000/swagger-ui/index.html