



News Summary

16.03.2023

Seida Basha, Lily Djami,

Eric J. Herschbach and Samuel Khodak

Outline



1. Requirements
2. Use Case
3. Technical Information
4. System Architecture
5. Working in Teams
6. Specification and Realization
 - a. Frontend & Backend
 - b. Implementation, Testing
7. Live Demo
8. Compare to Requirement

Requirements



Functional Requirements:

- Summary generation
- Fetching news from the past 24 hours
- Fetching news only from a specific category (i.e. sports news, business news)
- Summary generation using manual input
 - This summary can be downloaded

Requirements



Non-functional Requirements:

- Availability
 - The web app should be available for users 24/7, on-demand
 - Internet connection must be available to the user during operation
- Usability
 - The application should be easy to understand and not require training to use
 - Four dimensions: usefulness, satisfaction, ease of use, and reliability

Use Case

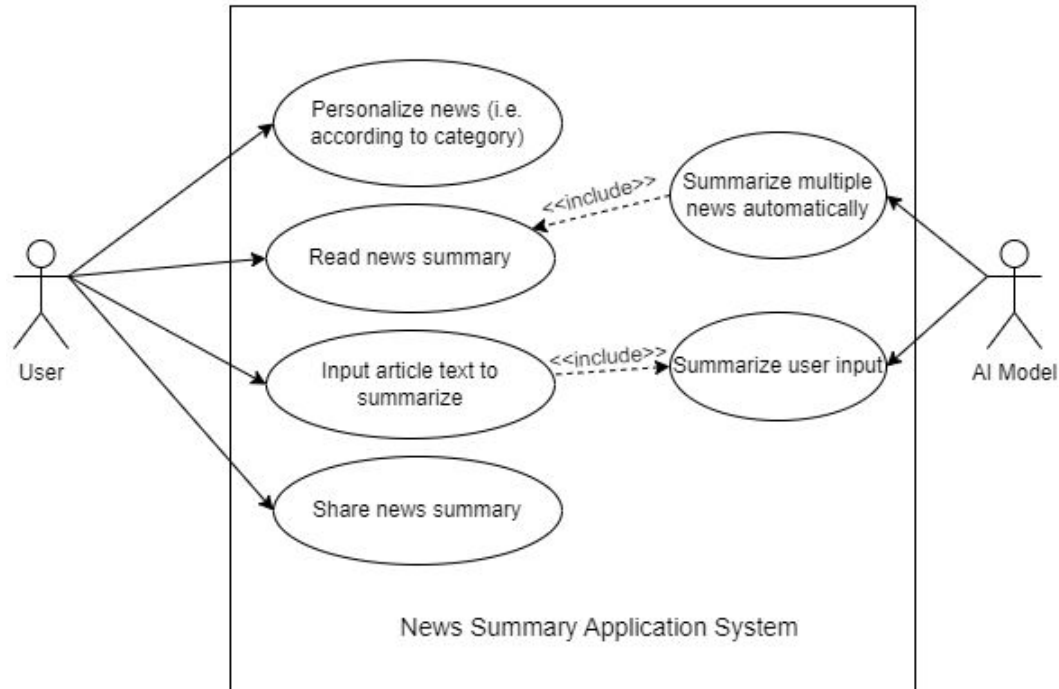


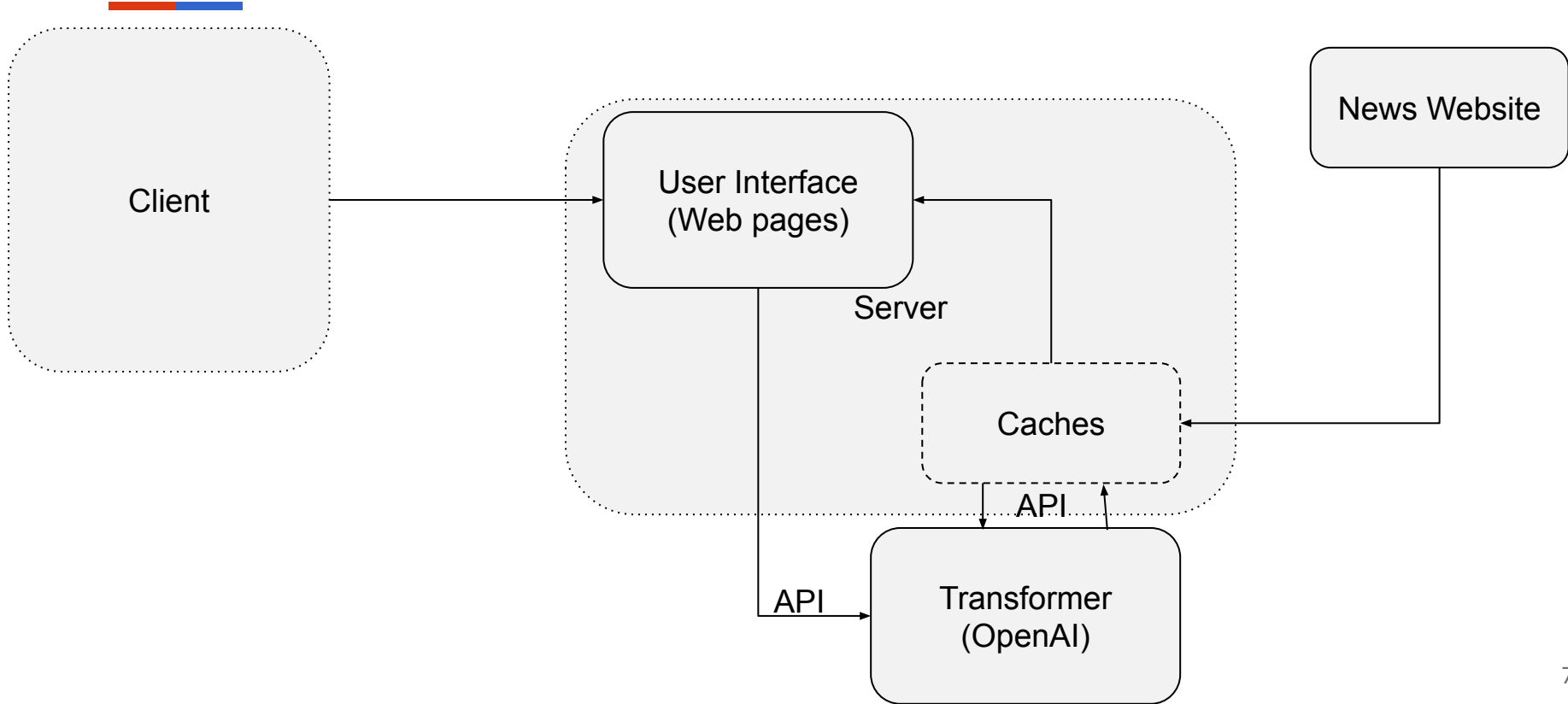
Figure 1: Use case diagram

Technical Information



- Used language: python (Streamlit)
 - Html was also considered an option
 - Streamlit was easier to use
- OpenAI API (Davinci model)
 - Custom transformer model was considered
 - Davinci has the required performance and accuracy
- Git for version control
- Discord for meetings

System-Architecture (Client-Server-Architecture)



Working in Teams

Frontend

Eric

Foundation, base
features, link user
Input to summarizer

Samuel

Contact form,
Help-page,
Page-structure

User Interface, Multipage, Design

Backend

Seida

Usability test

Lily

OpenAI API
connection

News fetching, Proofreading,
Usability test, Unit tests

Organization, Planning, Content

Specification & Realization



- Frontend:
 - Based on Streamlit (Python)
 - Web pages
 - Navigation via sidebar
 - Buttons and text area for interaction
 - Pictures for categories

Specification & Realization



- Backend:
 - Fetching news using RSS feeds
 - Scrape article text using BeautifulSoup
 - Connection via OpenAI API
 - Use OpenAI API for summary generation

Specification & Realization

- Unit test - Backend
 - Test scraping for text articles for the 3 implemented sources (The Guardian, BBC, BBC Sports)
 - Test fetching news
 - Test refresh

Coverage report: 83%

coverage.py v7.2.1, created at 2023-03-15 21:39 +0100

Module	statements	missing	excluded	coverage
unit_test.py	57	6	0	89%
utilities\bbc.py	19	0	0	100%
utilities\bbcsports.py	21	2	0	90%
utilities\newssite.py	49	6	0	88%
utilities\summarizer.py	29	19	0	34%
utilities\theguardian.py	19	0	0	100%
Total	194	33	0	83%

```
PS C:\Users\cons_\newssummarizer\News-Summarizer\NewsSummarizer\frontend> pytest
===== test session starts =====
platform win32 -- Python 3.11.2, pytest-7.2.2, pluggy-1.0.0
rootdir: C:\Users\cons_\newssummarizer\News-Summarizer\NewsSummarizer\frontend
collected 7 items

unit_test.py .....

===== warnings summary =====
..\..\..\AppData\Local\Programs\Python\Python311\Lib\site-packages\feedparser\encodings
lated for removal in Python 3.13
  import cgi

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
===== 7 passed, 1 warning in 9.44s =====
```

Figure 2: Results of the test coverage and the unit tests

Specification & Realization



- Testing Frontend
 - Unit Tests frontend:
 - Check if buttons activate a function
 - Check if the sidebar operational is
 - Check if news fetching works
 - Integration Test:
 - The text can be pasted in the text field and be summarized
 - The refresh button reloads the site and show new articles

Usability Test



The USERindex measures User Experience based on the 4 dimensions of: Usefulness, Satisfaction, Ease of Use and Reliability

Month/Year of Study	
No of participants	4
Industry	
Interface type	
SUS score (optional)	
NPS score (optional)	
Target country of study	

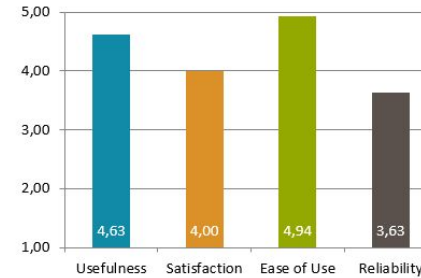
Please enter the **number of participants**

**USERindex
Score***

4,30

*value range from 1 to 5;
a value of between 4.2 and
4.7 is good and a value
greater than 4.7 is excellent

**USERindex
per Dimension**



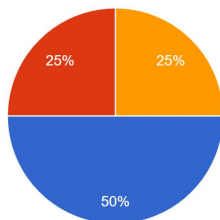
Dimension	Aspect	Statement	Rating					Score
			Strongly disagree	Somewhat disagree	Neutral	Somewhat agree	Strongly agree	
Reliability	Performance	I felt the site responded quickly.		1	2	1		3,00
	Accuracy	I found the information on the site to be accurate.				3	1	4,25
								3,63
Usefulness	Effectiveness	The site does what I need it to do.				1	3	4,75
	Usefulness	The information on the site was useful to me.			1		3	4,50
								4,63
Ease of Use	Ease of Use	I found the site easy to use.					4	5,00
	Navigation	It was easy to find what I was looking for on the site.					4	5,00
	Learnability	It didn't take me long to learn how to use the site.					4	5,00
	Efficiency	It didn't take me long to complete my task on the site.				1	3	4,75
								4,94
Satisfaction	Appeal	I found the design of the site appealing.			2	1	1	3,75
	Engagement	I liked using the site.			1	1	2	4,25
								4,00
USERindex score								4,30

Figure 3: Usability evaluation with USERindex

Summary Quality

Most keywords (or their synonyms) are included in the summary.

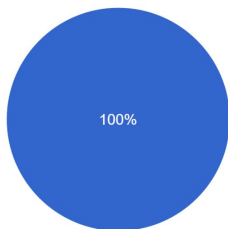
4 responses



- Strongly agree
- Somewhat agree
- Neutral
- Somewhat disagree
- Strongly disagree

No false statements have been made in the summary.

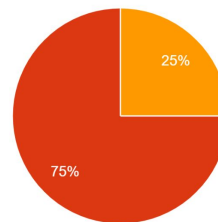
4 responses



- Strongly agree
- Somewhat agree
- Neutral
- Somewhat disagree
- Strongly disagree

The meaning of the article is not changed in the summary.

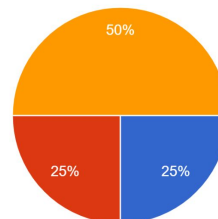
4 responses



- Strongly agree
- Somewhat agree
- Neutral
- Somewhat disagree
- Strongly disagree

The generated text is contextually sound.

4 responses



- Strongly agree
- Somewhat agree
- Neutral
- Somewhat disagree
- Strongly disagree

Figure 4: Requirement evaluation

Challenges and Risks



- Working with Machine Learning team
 - Communication problems
- Using OpenAI API
 - Limited free usage
 - Limited possibilities to customize results
- Ressources
 - Time limit
 - Training data

Live Demo

Comparison to Requirements



Planned	Realized
<ul style="list-style-type: none">- Summary Generation- Automatic news fetching from past 24 h ❌- Fetching news from selected categories- Summary Generation with manual input	<ul style="list-style-type: none">- Summary Generation using GPT-3 ✅- Fetching News using BeautifulSoup (Python library) ✅- Manual summary generation by using input textbox and connecting also to GPT-3 ✅

References and Links



Github repository: <https://github.com/Ansothso/News-Summeraizer>

Streamlit documentation: <https://docs.streamlit.io>

OpenAI API documentation: <https://platform.openai.com/docs/introduction/overview>

USERindex: <https://userindex.org/>

Thank you for listening