LABAP Project - MyPics

Authors:
Mauro Ficorella 1941639
Martina Turbessi 1944497
Valentina Sisti 1952657

Sapienza
DATA

Contents

1 Initial idea

Interactive dashboard to look for images uploaded by other users.

1.1 System objectives

- Show the dashboard with most popular images and images uploaded by followed users
- Manage user authentication
- Show a page related to users' profiles containing all the images uploaded by them
- Allow users to upload, search and see images
- Allow users to follow other users to easily access their user profile and images

1.2 Distributed Software Application with Containerization

- **Front-end layer**:
  - Login/registration page
  - Homepage
  - User profile page
  - User settings page
  - Image visualization page with description and comments
  - Image upload page
- **API gateway**: to take an application user's request, route it to one or more backend services, gather the appropriate data and deliver it to the user in a single, combined package
- **Logic layer**:
  - Microservice for user management: handles registration/authentication/access to the user profile
  - Microservice for notifications: allows the user to be notified about new likes or new comments on his images and new followers
  - Microservice for images management: handles the upload and deletion of images
  - Microservice for social part: allows the user to like/comment/save an image and follow another user
  - Microservice for search: handles the search of an image or an user
- **Persistence layer**: NoSQL database

Each element of this list represents a different Docker container of the system and all the containers are orchestrated using Docker Compose.

1.3 Potential Users

- People interested in discovering images from people that they follow
- People interested in uploading and sharing images

1.4 Use Cases

- User can register to the app
- User can login into the app
- User can search for an image
- User can search for an user
- User can visualize the home page containing most popular images and the images published by followed users
- User can upload an image
- User can save an image
- User can remove an uploaded image
- User can like an image
- User can comment an image
- User can get notified if another user likes one of its images
- User can get notified if another user leave a comment on one of its images
- User can get notified for a new follower
- User can follow/unfollow another user
- User can access its own profile to visualize his images and the ones that he saved from other users
- User can access its own profile to manage it
- User can access another user profile to view his details and his published images
- User can logout
- User can delete its account

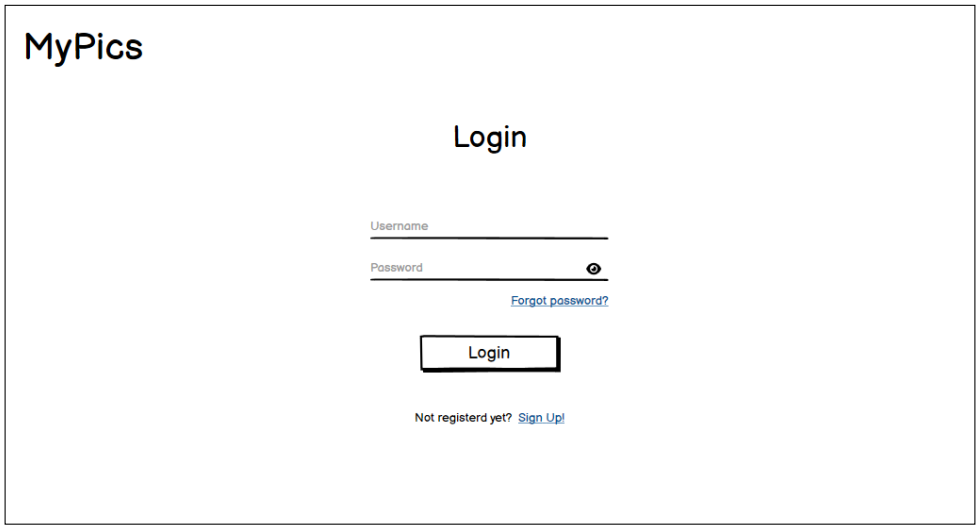
2 User stories and mockups

2.1 Sign-up

AS A	I WANT TO	SO THAT I CAN	ADDED BY
Guest	Register to the system	Create my profile	Everyone

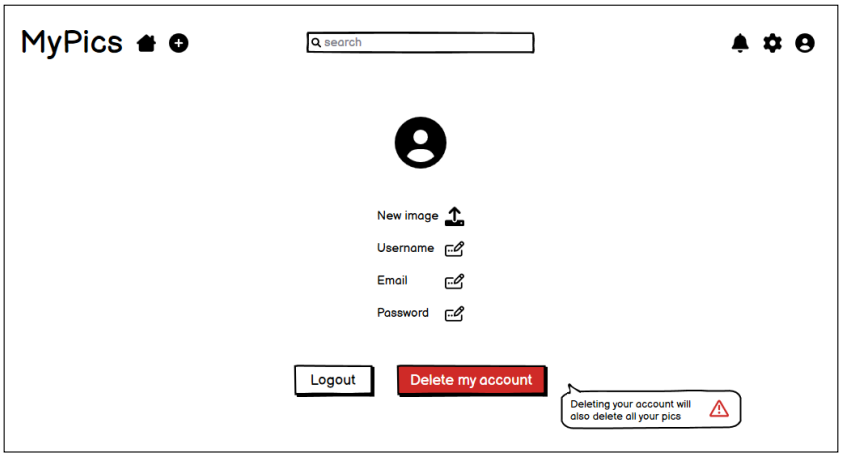
2.2 Login

AS A	I WANT TO	SO THAT I CAN	ADDED BY
Non-logged registered user	Login into the system	Use system's services	Everyone



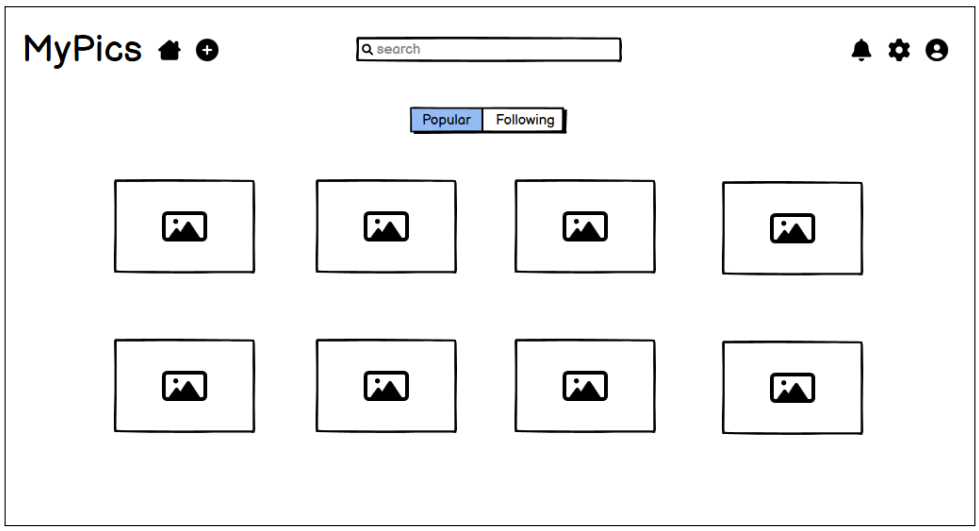
2.3 User settings

AS A	I WANT TO	SO THAT I CAN	ADDED BY
Logged registered user	Logout from the system	Login as another user	Everyone
Logged user	Delete my profile	No longer access the system	Everyone
Logged user	Access my profile settings	Manage it	Everyone



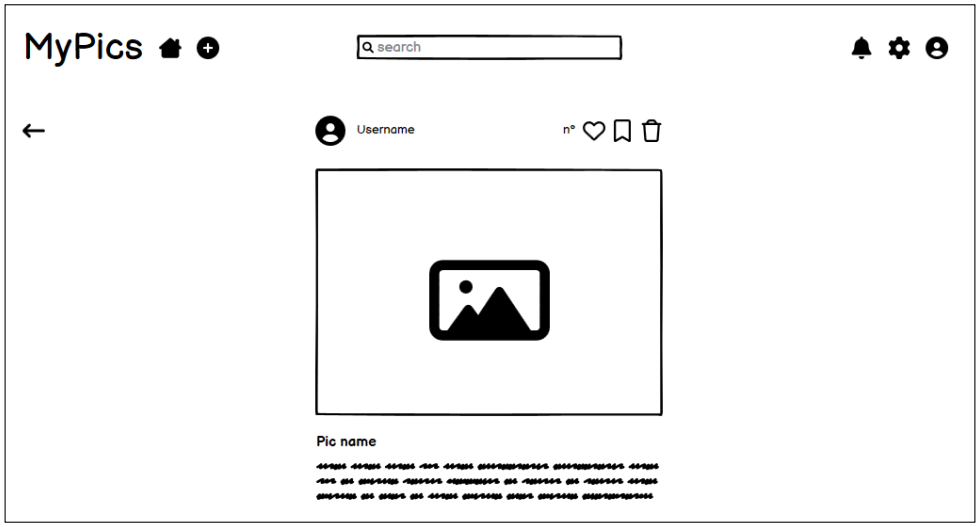
2.4 Main

AS A	I WANT TO	SO THAT I CAN	ADDED BY
Logged user	Access the homepage	Discover the most popular images	Everyone
Logged user	Access the homepage	Discover images published by followed users	Everyone
Logged user	Search for images	Visualize them	Everyone
Logged user	Search for other users	Visualize their profile and their images	Everyone
Logged user	Get notified	Know if another user liked or commented one of my image or followed me	Everyone
Logged user	Access other user's profile page	Visualize his details and published images	Everyone
Logged user	Visualize image	Visualize its details and comments	Everyone



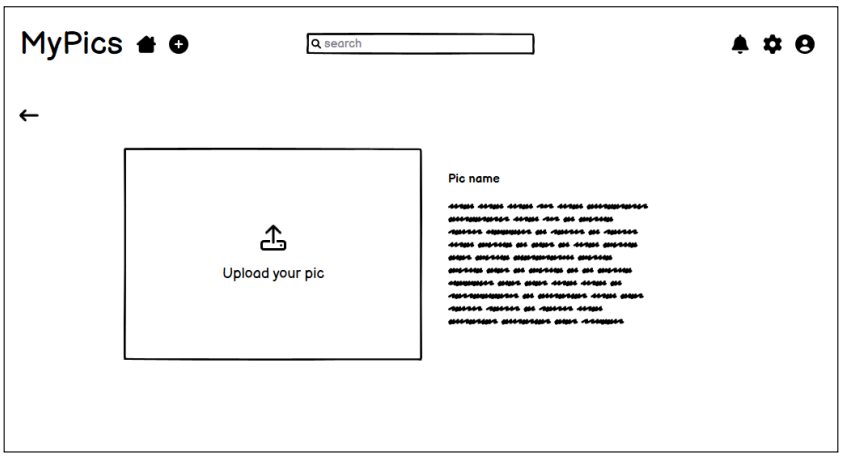
2.5 Pic

AS A	I WANT TO	SO THAT I CAN	ADDED BY
Logged user	Like an image	Express my appreciation about it	Everyone
Logged user	Save an image	Discover the most popular images	Everyone
Logged user	Remove an uploaded image	Deny to other users to visualize it	Everyone



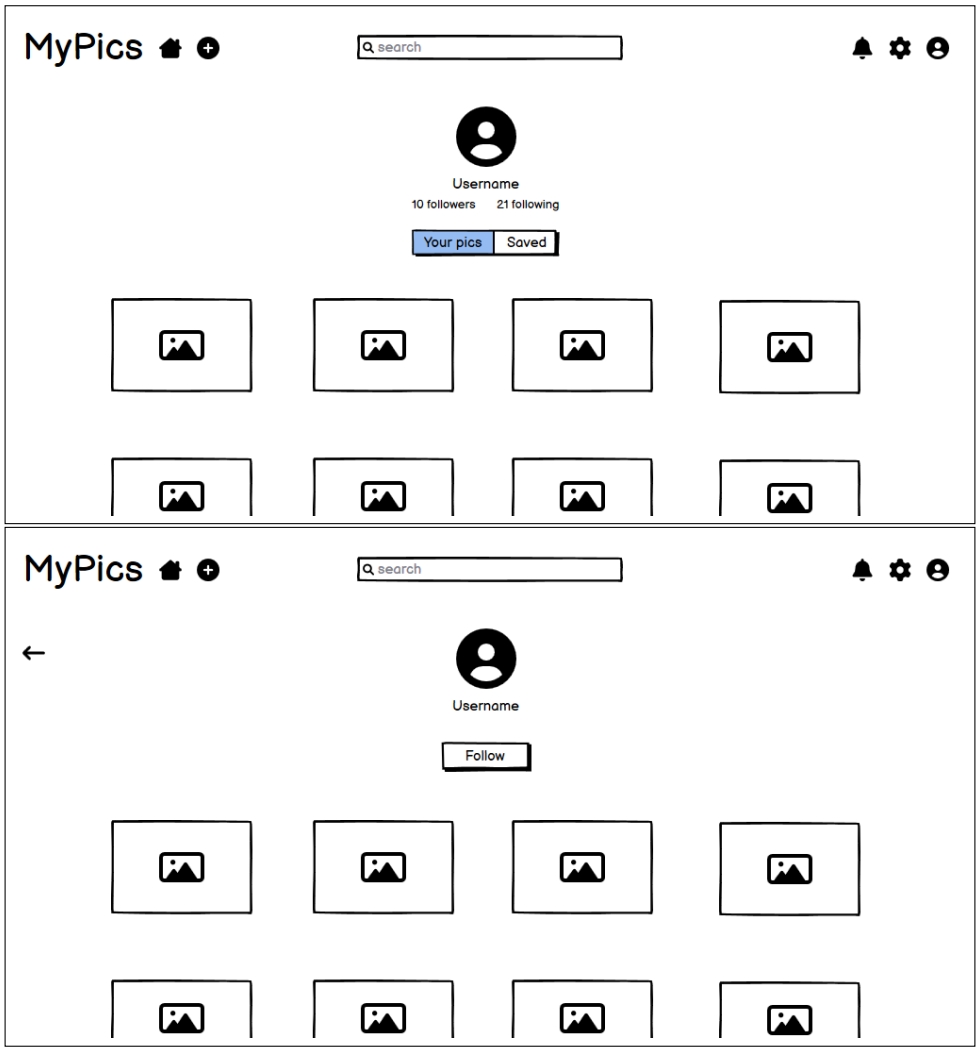
2.6 Upload pic

AS A	I WANT TO	SO THAT I CAN	ADDED BY
Logged user	Upload an image	Share it with other users	Everyone



2.7 User profile

AS A	I WANT TO	SO THAT I CAN	ADDED BY
Logged user	Access my profile	Visualize all my published images	Everyone
Logged user	Access my profile	Visualize all my saved images	Everyone
Logged user	Access user profile	Visualize followers/followed users	Everyone
Logged user	Follow another user	Stay updated about his new images	Everyone



3 Effort estimation

3.1 Function Points

No.	Module	Function Name	Description	Type	DET	RET / FTR	Complexity	FP	Remarks	Adjust %	FP adjusted
1	User	User	Model stored to represent the users	ILF	5	1	Low	7			**7**
2	Post	Post	Model stored to represent the posts	ILF	5	1	Low	7			**7**
3	Comment	Comment	Model stored to represent the comments	ILF	3	1	Low	7			**7**
4	Notification	Notification	Model stored to represent the notifications	ILF	3	1	Low	7			**7**
5	Login	Login	Login to the application	EI	2	1	Low	3			**3**
6	Sign up	Sign up	Sign up to the application	EI	3	1	Low	3			**3**
7	Main Page	Show popular images	Shows popular images published on MyPics	EQ	5	2	Low	3			**3**
8	Main Page	Show followed users' images	Shows images published by followed users	EQ	5	2	Low	3			**3**
9	Search	Search	Seach for images published on MyPics or users	EQ	6	2	Average	4			**4**
10	Pic	Visualize pic	Show image with its details	EO	15	3	Average	5			**5**
11	Pic	Add pic	Publish image on MyPics	EI	6	2	Average	4			**4**
12	Pic	Delete pic	Delete published image from MyPics	EI	1	1	Low	3			**3**
13	Pic	Add/Remove like	User add/remove like on post	EI	2	2	Low	3			**3**
14	Pic	Add comment	User add comment on post	EI	5	3	High	6			**6**
15	Pic	Delete comment	User delete previously published on post	EI	1	1	Low	3			**3**
16	Pic	Save pic	User save an image	EI	2	2	Low	3			3
17	Profile	Show published images	Show images on user profile	EQ	5	2	Low	3			**3**
18	Profile	Show saved images	Show saved images on user profile	EQ	5	2	Low	3			**3**
19	Profile	Follow/Unfollow user	User follow/unfollow another user	EI	2	1	Low	3			**3**
20	Profile	Visualize followers	Show list of followers	EO	4	1	Low	4			**4**
21	Profile	Visualize followed users	Show list of followed users	EO	4	1	Low	4			**4**
22	Profile settings	Update profile pic	Update profile pic	EI	2	1	Low	3			**3**
23	Profile settings	Update username	Update username	EI	2	1	Low	3			**3**
24	Profile settings	Update email	Update email	EI	2	1	Low	3			**3**
25	Profile settings	Update password	Update password	EI	2	1	Low	3			**3**
26	Profile settings	Delete account	Delete account registered on MyPics	EI	1	1	Low	3			**3**
27	Notification	Show notifications	Show notifications of received likes, comments and follows	EQ	6	3	Average	4			**4**
28	Notification	Update notification	Update not read notification to read notification	EI	2	1	Low	3			**3**
29	Notification	Add notification	Add notification on database	EI	5	3	High	6			**6**

Unadjusted FP	**116**

Considering Java as main language, this is equivalent to 6148 SLOC.

3.2 CoCoMo II

COCOMO II - Constructive Cost Model

Monte Carlo Risk: Off
Auto Calculate: Off

Software Size Sizing Method: Function Points

Unadjusted Function Points: 116 Language: Java

Software Scale Drivers

Precedentedness	Nominal	Architecture / Risk Resolution	Nominal	Process Maturity	Low
Development Flexibility	Very High	Team Cohesion	Very High		

Software Cost Drivers

Product		Personnel		Platform	
Required Software Reliability	Low	Analyst Capability	High	Time Constraint	Nominal
Data Base Size	Nominal	Programmer Capability	High	Storage Constraint	Nominal
Product Complexity	Nominal	Personnel Continuity	Very High	Platform Volatility	Nominal
Developed for Reusability	Nominal	Application Experience	Nominal	**Project**	
Documentation Match to Lifecycle Needs	Nominal	Platform Experience	Nominal	Use of Software Tools	Very High
		Language and Toolset Experience	Nominal	Multisite Development	Extra High
				Required Development Schedule	Nominal

Maintenance Off

Software Labor Rates

Cost per Person-Month (Dollars): 2000

Calculate

Results

Software Development (Elaboration and Construction)

Effort = 7.2 Person-months
Schedule = 6.8 Months
Cost = $14359

Total Equivalent Size = 6148 SLOC
Effort Adjustment Factor (EAF) = 0.35

Staffing Profile

Your project is too small to display a staffing profile due to truncation.

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	0.4	0.8	0.5	$862
Elaboration	1.7	2.5	0.7	$3446
Construction	5.5	4.2	1.3	$10913
Transition	0.9	0.8	1.0	$1723

Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.1	0.2	0.5	0.1
Environment/CM	0.0	0.1	0.3	0.0
Requirements	0.2	0.3	0.4	0.0
Design	0.1	0.6	0.9	0.0
Implementation	0.0	0.2	1.9	0.2
Assessment	0.0	0.2	1.3	0.2
Deployment	0.0	0.1	0.2	0.3

4 System architecture

The system architecture is based on microservices, each running on its own Docker container and each accessing its own data. These containers are orchestrated through Docker Compose.

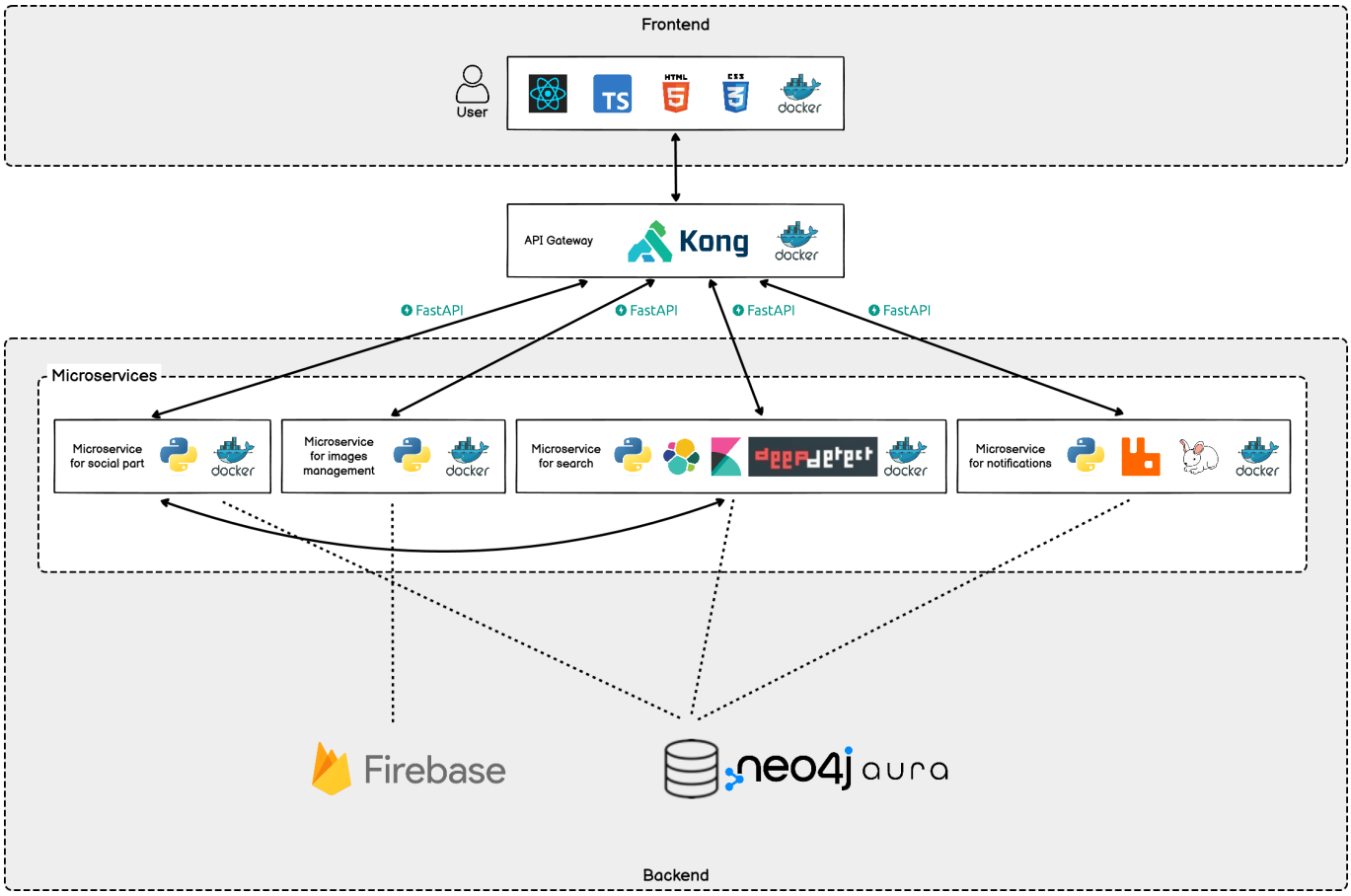


Figure 1: Overview of system architecture

- **Frontend:** developed in React (based on TypeScript, HTML and CSS). Users interact with the system through this layer.
- **API Gateway:** we use Kong Gateway which is a lightweight, fast, and flexible cloud-native API gateway that serves as a reverse proxy that lets us manage, configure, and route requests to our APIs exposed by the microservices.
- **Microservice for social part:** exposes APIs to manage all social aspects of the system (i.e. user registration, likes, comments, new posts).
- **Microservice for images management:** exposes APIs to upload images on the Firebase cloud storage.
- **Microservice for search:** we use Elasticsearch as search engine and Kibana, which provides search and data visualization capabilities for data indexed in Elasticsearch. Moreover we use Deepdetect deep learning platform which offers pre-trained models for classifing images' based on represented subjects and to be able to search for images based on them through Elasticsearch.
- **Microservice for notifications:** we use RabbitMQ message oriented middleware through CloudAMQP, which provides managed RabbitMQ servers in the cloud, in order to be able to access all the different queues in the cloud from any device.

- **Database:** we use Neo4j NoSQL graph database in order to store all the system data. In particular we use Neo4j AuraDB that offers a database instance in the cloud with which all the above mentioned microservices can communicate.

We developed the backend microservices using Python and all of them expose a REST interface that leverages on FastAPI framework.

5 Sprint analytics