PetSeeker

universidade de aveiro

- **David Raposo** - 93395
- **Diogo Torrinhas** - 98440
- **João Torrinhas** - 98435
- **Miguel Tavares** - 98448
- **Tiago Bastos** - 97590

- Software Engineering 2023/2024
- Group 5
- 18/12/2023

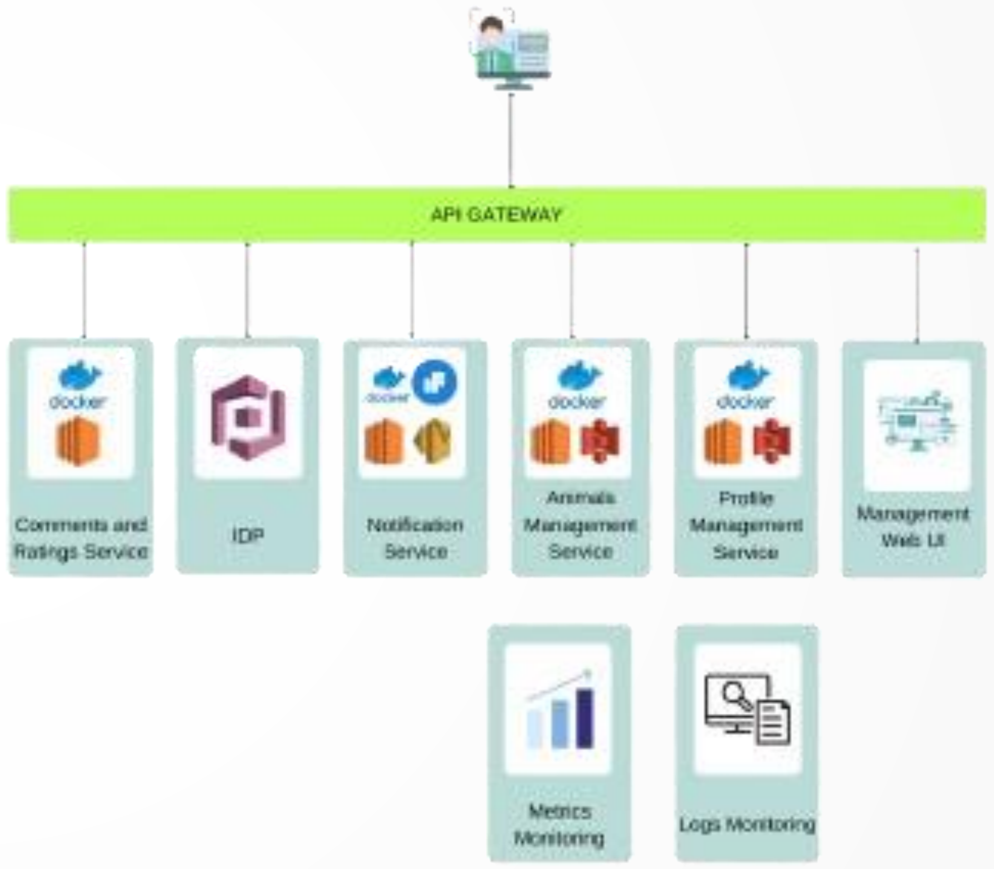
Value Of Our App

- **Adopt/Buy your next friend:**
  - Browse through a diverse selection of adorable animals waiting for their forever homes.
  - Easy sign-up and secure adoption/purchase process.
- **Build your profile:**
  - Create a personalized profile with your information (name, location, interests...).
- **Notifications:**
  - The user receives notifications if an animal of its interest is published.
  - The user can turn off the notifications to avoid the spam of them.
  - The user receives a notification if their post is commented on.
- **User ratings and comments:**
  - Establish credibility with a user rating system.
  - Engage in meaningful conversations by commenting on other user's posts.

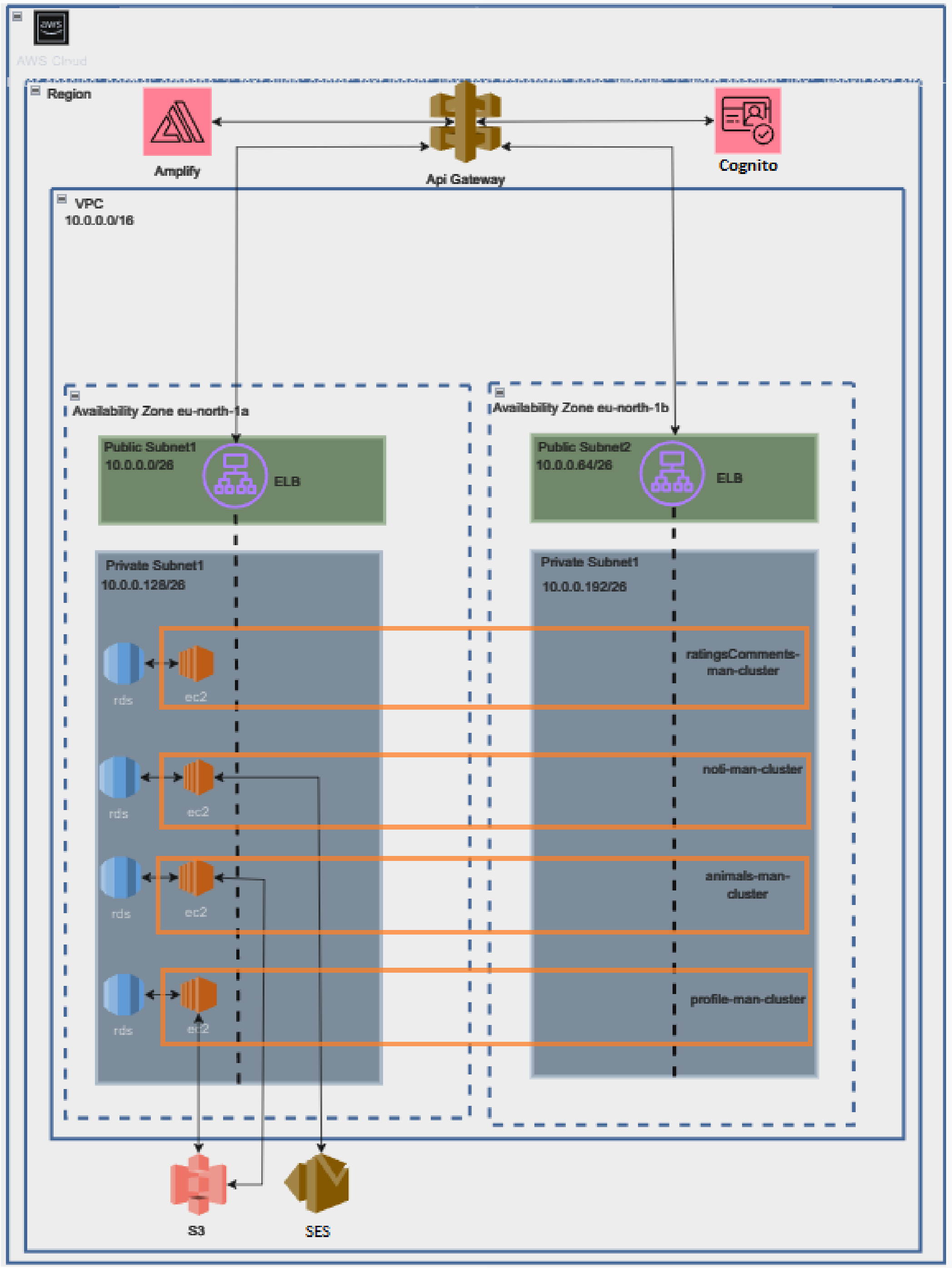
Value Of Our App

- **Admin Oversight:**
  - The admin ensures the quality of content.
  - Publications undergo validation to maintain a safe and reliable environment for the users.

Architecture Solution



AWS Architecture Solution



Micro-Services

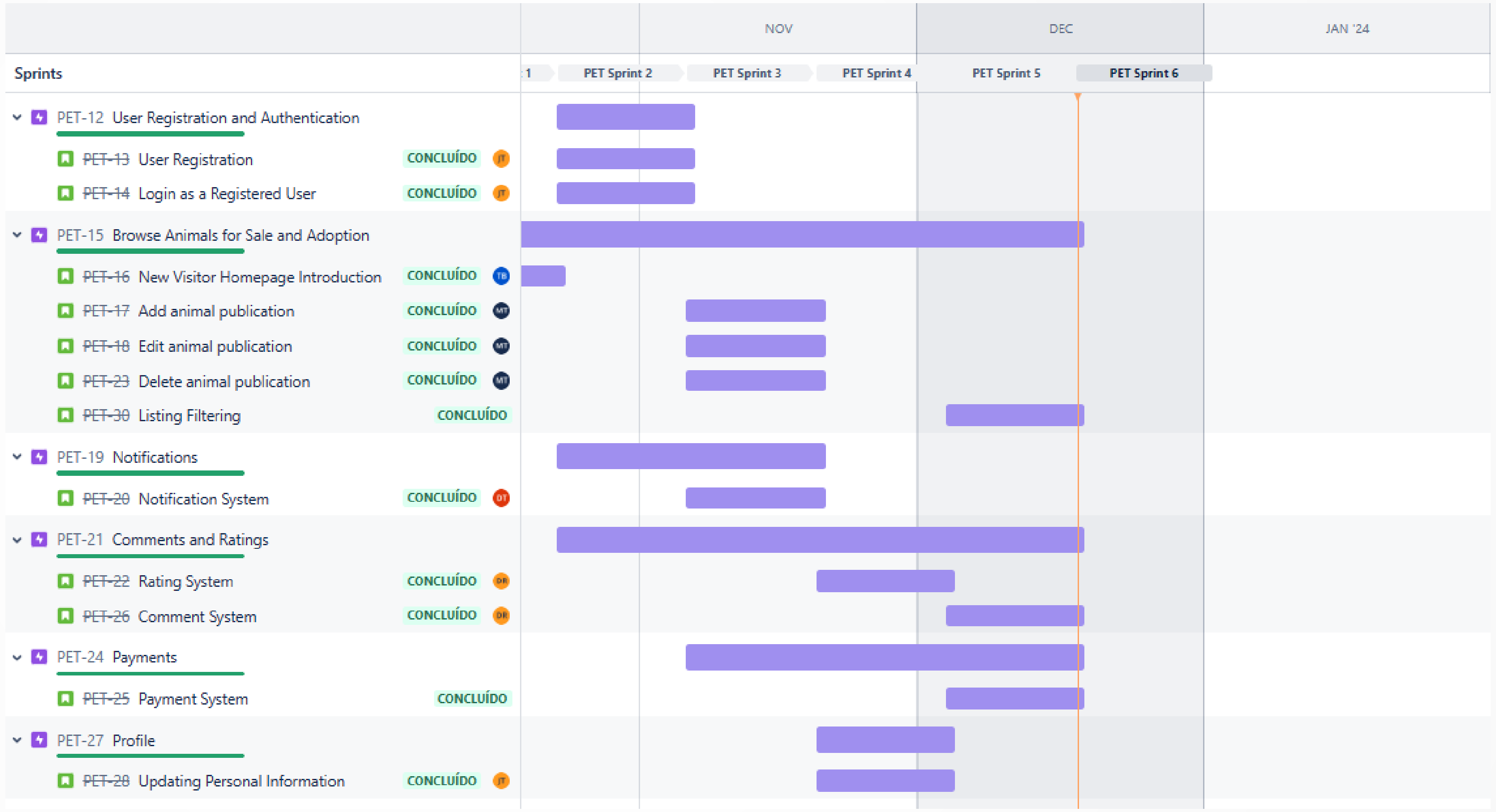
- **Animals-Management-Service:** service to manage the publications of the animals (add, delete, and edit publications).
- **Notifications-Service:** service to send notifications to the email of the users. The users have the possibility to turn off the notifications.
- **Profile-Management-Service:** service to manage the user profile(create and edit profiles).
- **Ratings&Comments-Service:** service to manage the rating systems of the users and the comments and replies of the publications.
- **WebApp frontend** – Website of our aplication.

Cloud Services Adopted

- **AWS Amplify** - for hosting front-end Webapp.
- **AWS Cognito** – acts as an IDP, to create and authenticate users.
- **AWS Api Gateway** – acts as a scalable and secure intermediary for front-end Amplify application, facilitating seamless API calls to services hosted on EC2 instances and Cognito.
- **AWS ECS** – cluster that run a EC2 instance through task definitions and ECS services.
- **AWS ECR** - to save the docker images in the repository.
- **AWS ELB** - distributes incoming traffic across EC2 instances in your cluster, enhancing availability and fault tolerance.
- **AWS RDS** - hosts dedicated databases for each microservice, offering a fully managed and scalable relational database solution.
- **AWS S3** - stores images from the microservice, providing a scalable and durable object storage solution.
- **AWS SES** - facilitates reliable email notifications for your microservice.

Backlog Overview

Project Timeline



NOV | DEC | JAN '24

Sprints: 1 · PET Sprint 2 · PET Sprint 3 · PET Sprint 4 · PET Sprint 5 · PET Sprint 6

PET-12 User Registration and Authentication
- ~~PET-13~~ User Registration — CONCLUÍDO
- ~~PET-14~~ Login as a Registered User — CONCLUÍDO

PET-15 Browse Animals for Sale and Adoption
- ~~PET-16~~ New Visitor Homepage Introduction — CONCLUÍDO
- ~~PET-17~~ Add animal publication — CONCLUÍDO
- ~~PET-18~~ Edit animal publication — CONCLUÍDO
- ~~PET-23~~ Delete animal publication — CONCLUÍDO
- ~~PET-30~~ Listing Filtering — CONCLUÍDO

PET-19 Notifications
- ~~PET-20~~ Notification System — CONCLUÍDO

PET-21 Comments and Ratings
- ~~PET-22~~ Rating System — CONCLUÍDO
- ~~PET-26~~ Comment System — CONCLUÍDO

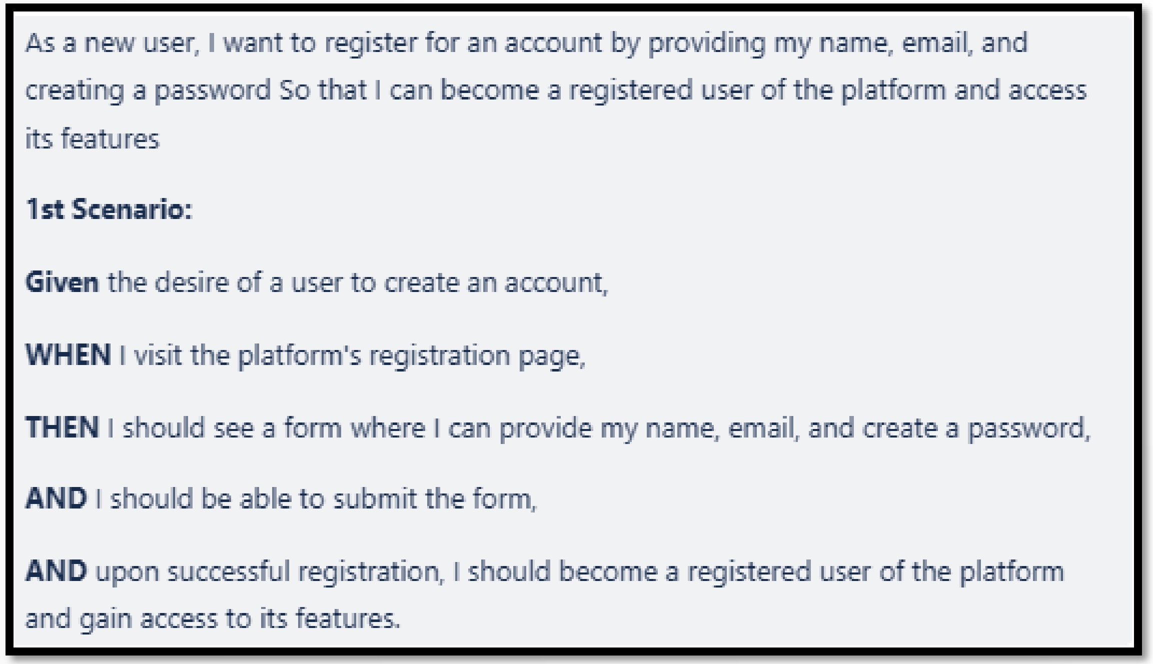
PET-24 Payments
- ~~PET-25~~ Payment System — CONCLUÍDO

PET-27 Profile
- ~~PET-28~~ Updating Personal Information — CONCLUÍDO

Backlog Overview

User story examples

User Registration



As a new user, I want to register for an account by providing my name, email, and creating a password So that I can become a registered user of the platform and access its features

1st Scenario:

Given the desire of a user to create an account,

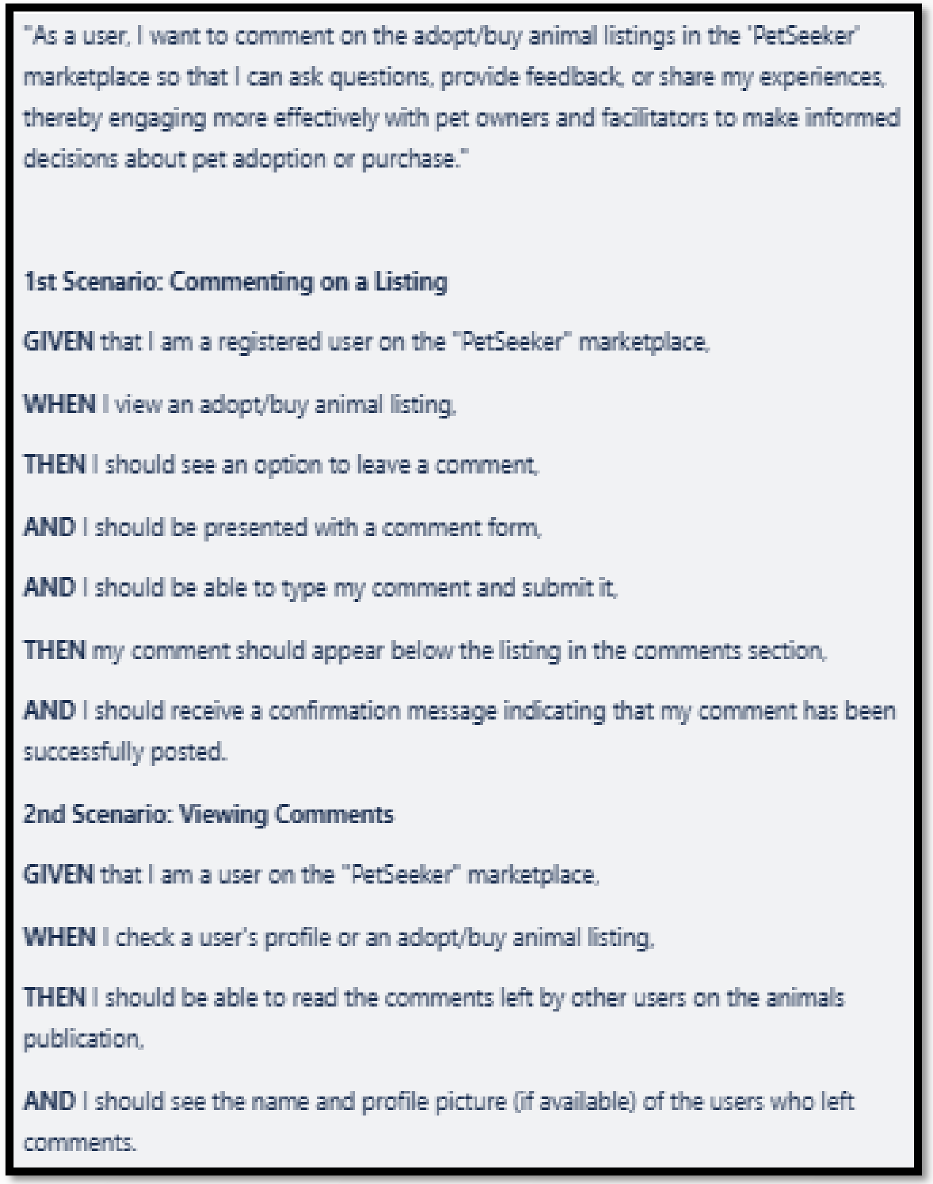
WHEN I visit the platform's registration page,

THEN I should see a form where I can provide my name, email, and create a password,

AND I should be able to submit the form,

AND upon successful registration, I should become a registered user of the platform and gain access to its features.

Comment System



"As a user, I want to comment on the adopt/buy animal listings in the 'PetSeeker' marketplace so that I can ask questions, provide feedback, or share my experiences, thereby engaging more effectively with pet owners and facilitators to make informed decisions about pet adoption or purchase."

1st Scenario: Commenting on a Listing

GIVEN that I am a registered user on the "PetSeeker" marketplace,

WHEN I view an adopt/buy animal listing,

THEN I should see an option to leave a comment,

AND I should be presented with a comment form,

AND I should be able to type my comment and submit it,

THEN my comment should appear below the listing in the comments section,

AND I should receive a confirmation message indicating that my comment has been successfully posted.

2nd Scenario: Viewing Comments

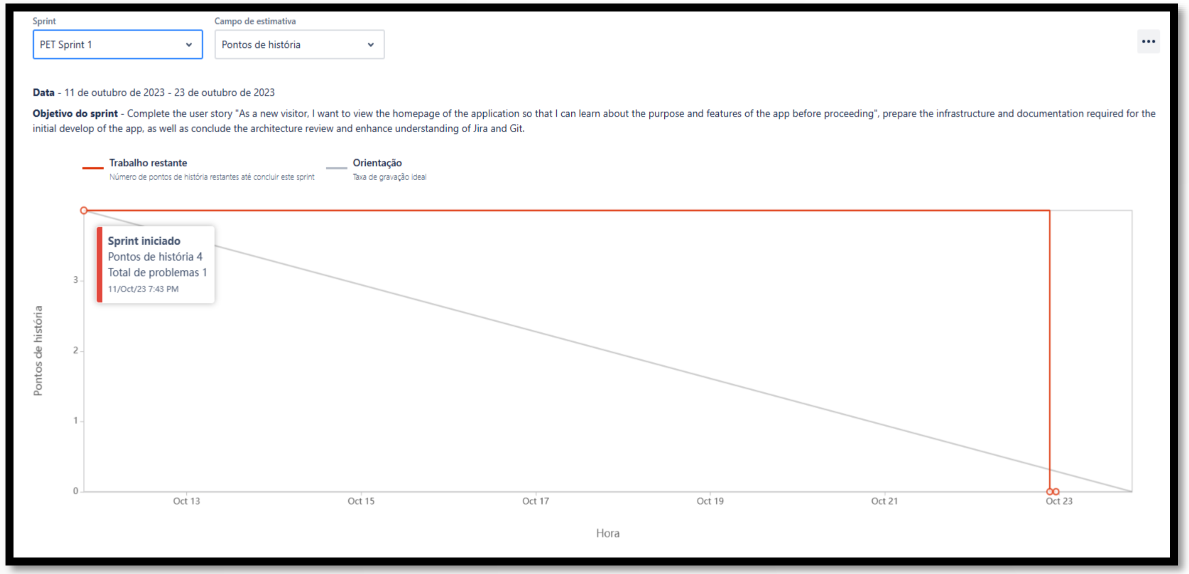
GIVEN that I am a user on the "PetSeeker" marketplace,

WHEN I check a user's profile or an adopt/buy animal listing,

THEN I should be able to read the comments left by other users on the animals publication,

AND I should see the name and profile picture (if available) of the users who left comments.

Backlog Overview

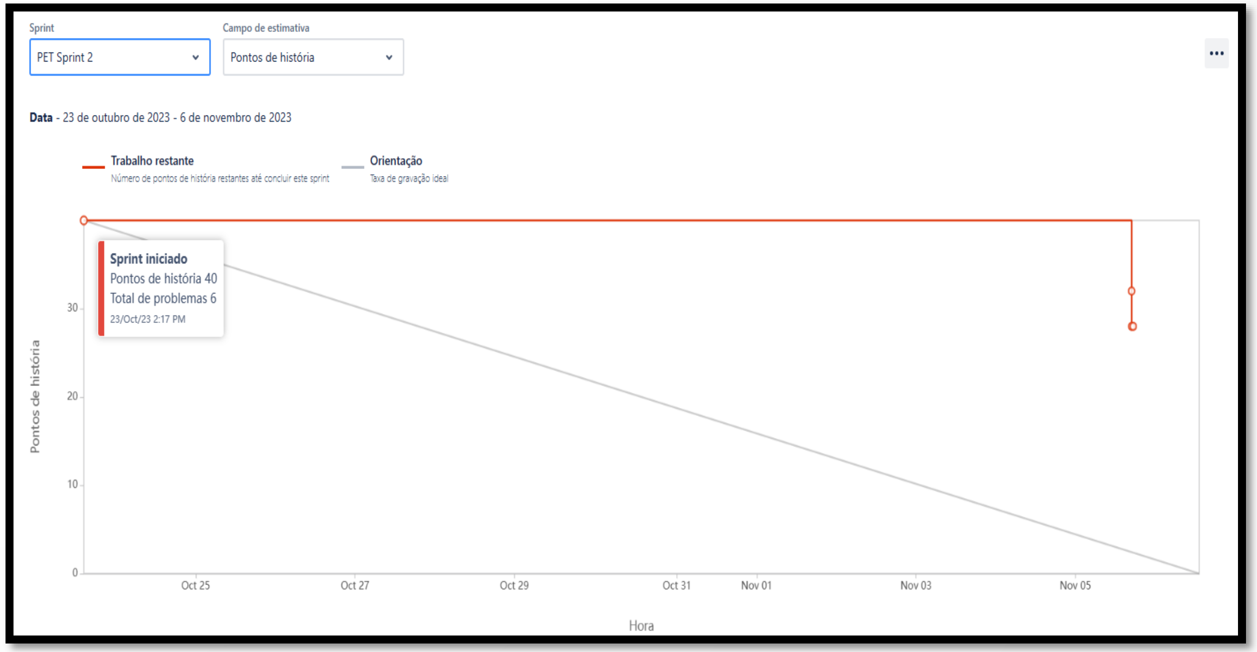


Sprint: PET Sprint 1

Campo de estimativa: Pontos de história

Data - 11 de outubro de 2023 - 23 de outubro de 2023

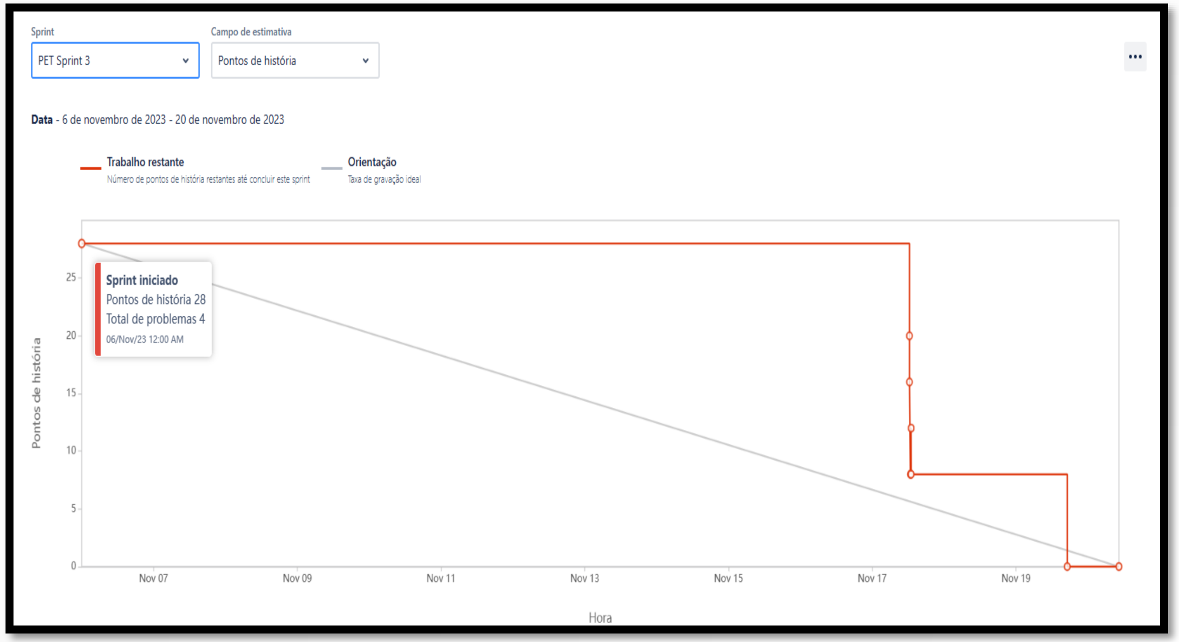
Objetivo do sprint - Complete the user story "As a new visitor, I want to view the homepage of the application so that I can learn about the purpose and features of the app before proceeding", prepare the infrastructure and documentation required for the initial develop of the app, as well as conclude the architecture review and enhance understanding of Jira and Git.



Sprint: PET Sprint 2

Campo de estimativa: Pontos de história

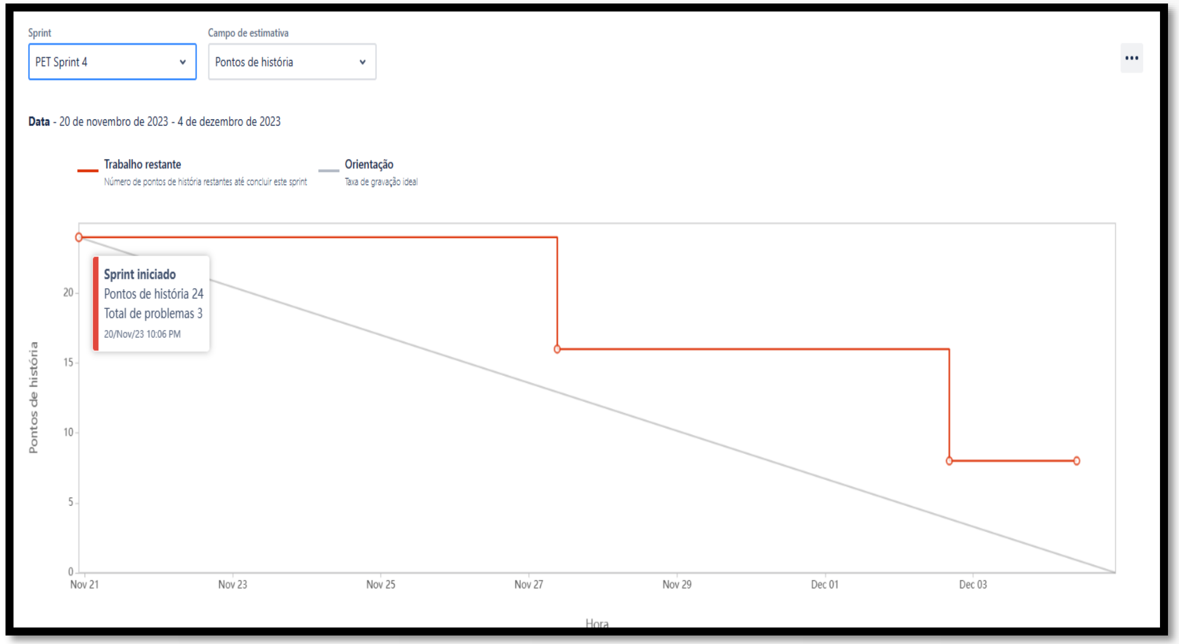
Data - 23 de outubro de 2023 - 6 de novembro de 2023



Sprint: PET Sprint 3

Campo de estimativa: Pontos de história

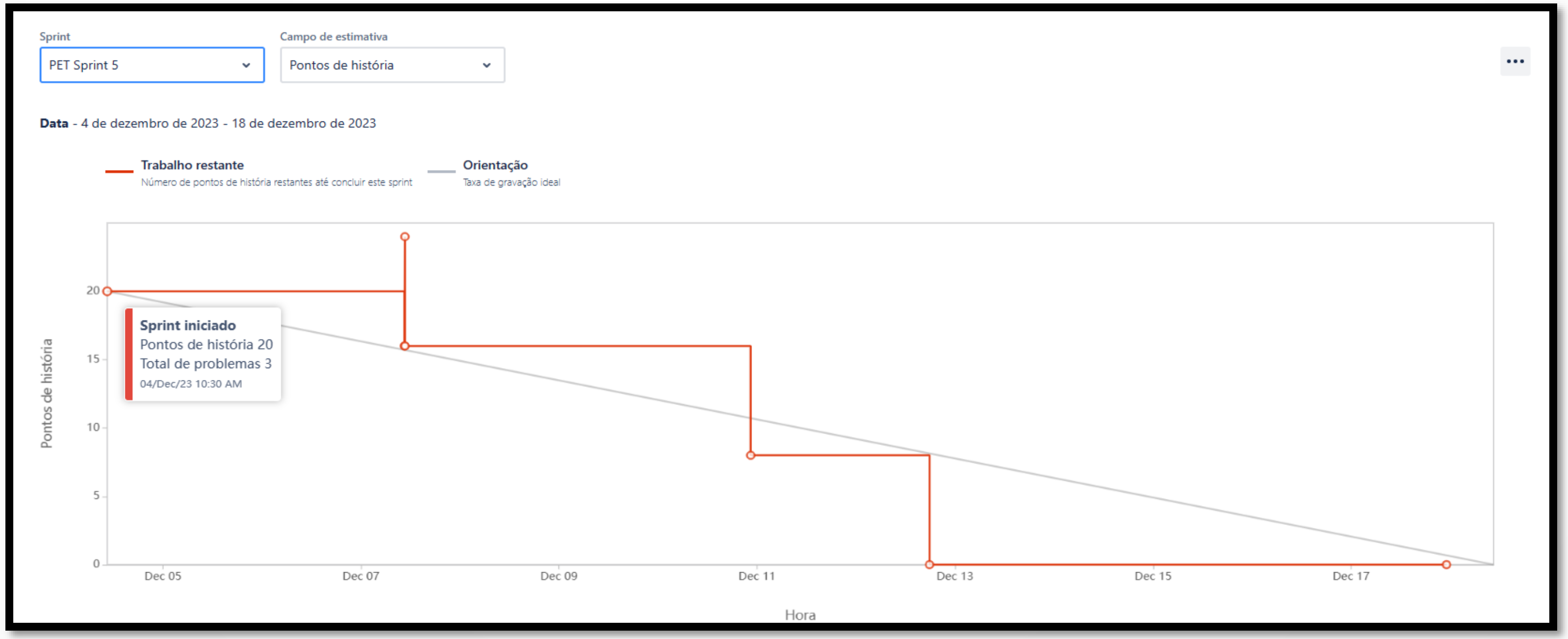
Data - 6 de novembro de 2023 - 20 de novembro de 2023



Sprint: PET Sprint 4

Campo de estimativa: Pontos de história

Data - 20 de novembro de 2023 - 4 de dezembro de 2023



Sprint: PET Sprint 5

Campo de estimativa: Pontos de história

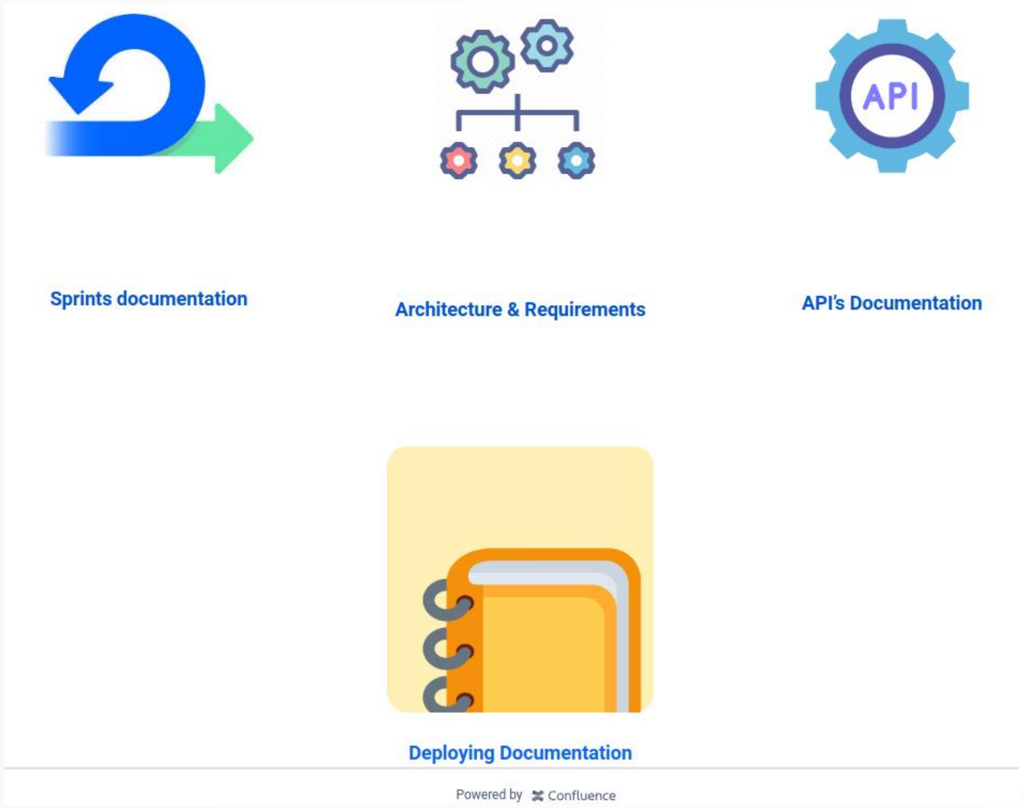
Data - 4 de dezembro de 2023 - 18 de dezembro de 2023

Lessons learned

- **Improved Sprint Organization:**
  - We learned the importance of organization and split tasks in each sprint.
- **User Story Refinement:**
  - We discovered the importance of breaking down user stories into smaller, manageable components.
- **Increased Meeting Frequency:**
  - We recognized the need for more regular team meetings to promote communication and collaboration.
- **Prioritizing Value Delivery:**
  - We understanded the significance of prioritizing the delivery of value to the user in our projects.

Documentation Website Showcase

- https://software-enginner-ua.atlassian.net/wiki/spaces/PST/overview



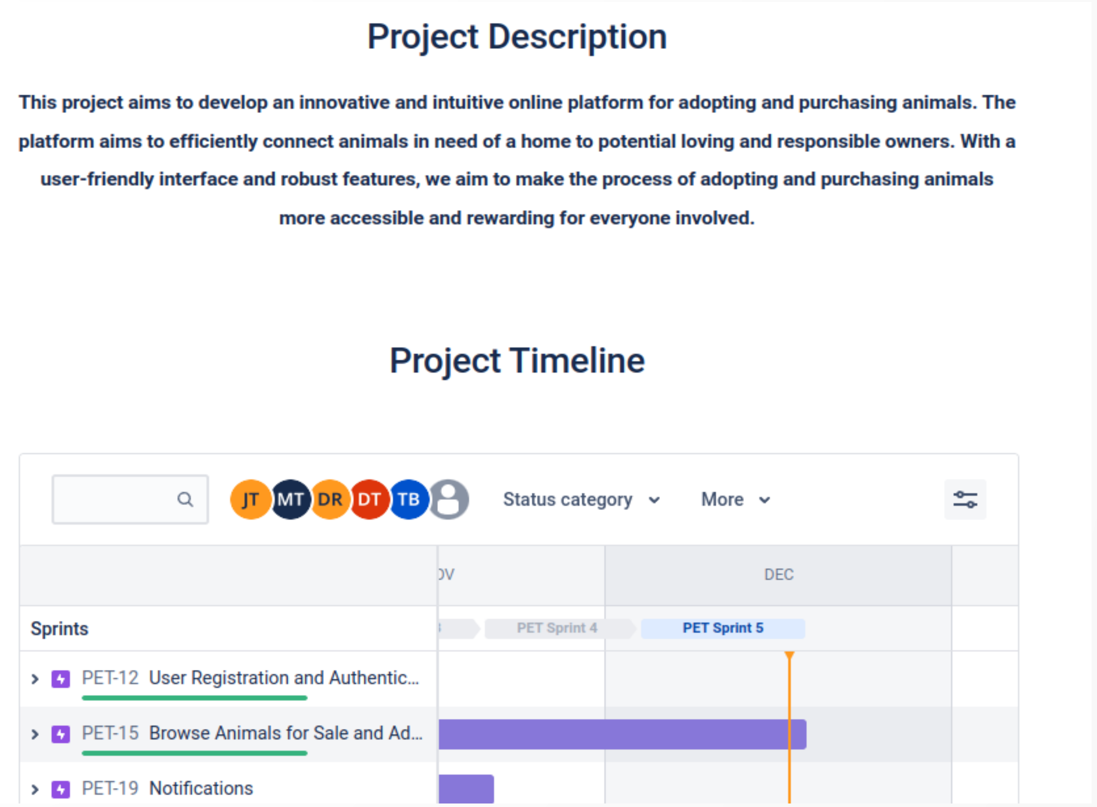
Sprints documentation

Architecture & Requirements

API's Documentation

Deploying Documentation

Powered by Confluence

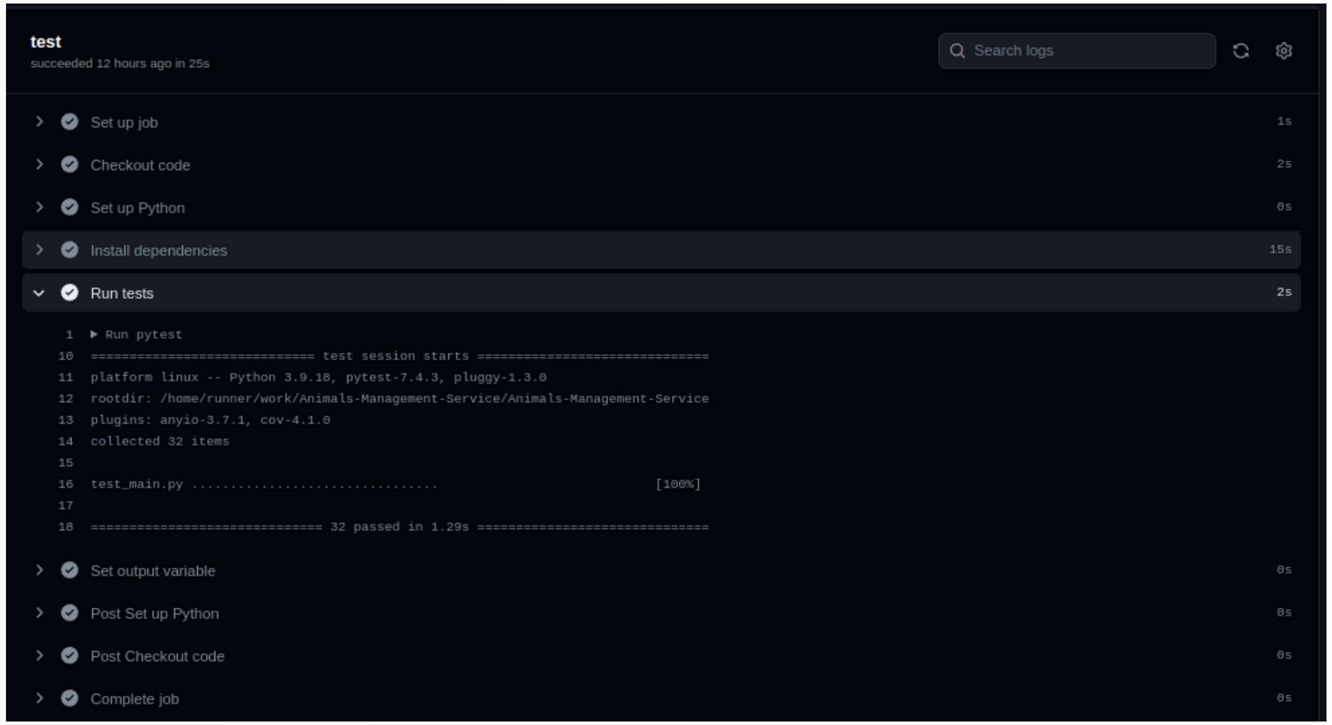


Project Description

This project aims to develop an innovative and intuitive online platform for adopting and purchasing animals. The platform aims to efficiently connect animals in need of a home to potential loving and responsible owners. With a user-friendly interface and robust features, we aim to make the process of adopting and purchasing animals more accessible and rewarding for everyone involved.

Project Timeline

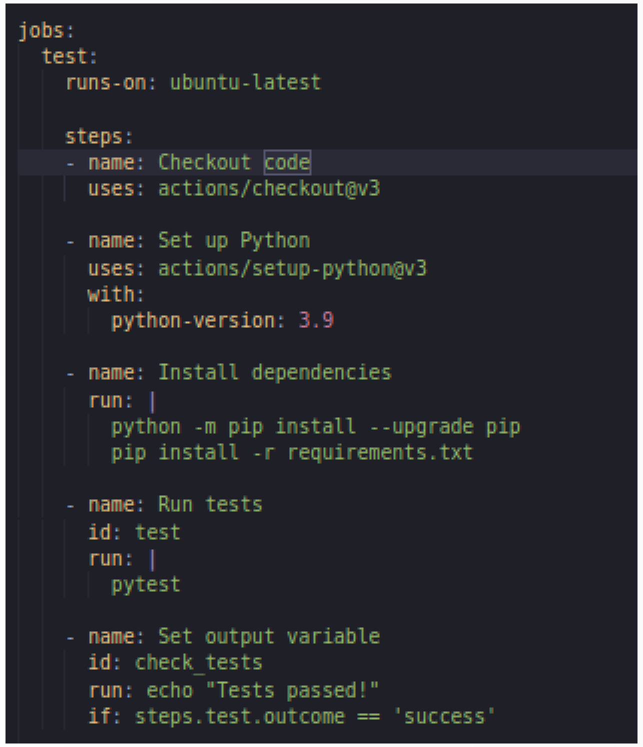
CI - Testing



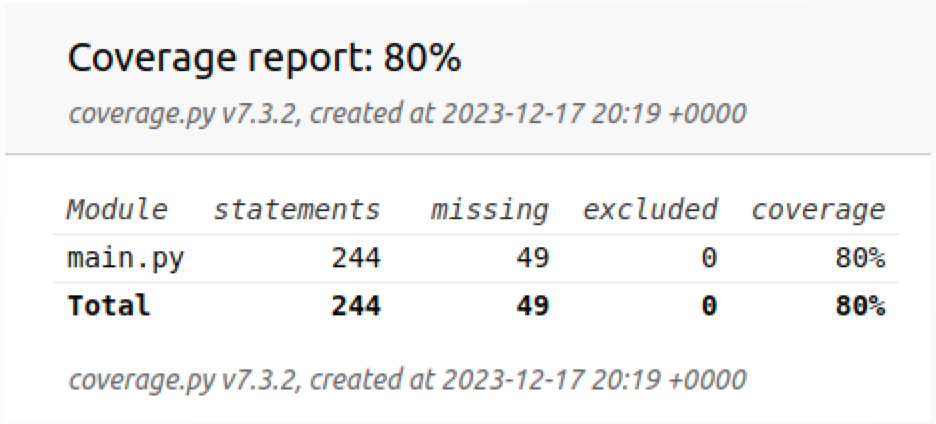
test
succeeded 12 hours ago in 25s

Search logs

```
> Set up job                1s
> Checkout code             2s
> Set up Python             0s
> Install dependencies      15s
v Run tests                 2s
  ▶ Run pytest
  ============================ test session starts =============================
  platform linux -- Python 3.9.18, pytest-7.4.3, pluggy-1.3.0
  rootdir: /home/runner/work/Animals-Management-Service/Animals-Management-Service
  plugins: anyio-3.7.1, cov-4.1.0
  collected 32 items

  test_main.py ................................                          [100%]

  ============================= 32 passed in 1.29s =============================
> Set output variable       0s
> Post Set up Python        0s
> Post Checkout code        0s
> Complete job              0s
```



```yaml
jobs:
  test:
    runs-on: ubuntu-latest

    steps:
    - name: Checkout code
      uses: actions/checkout@v3

    - name: Set up Python
      uses: actions/setup-python@v3
      with:
        python-version: 3.9

    - name: Install dependencies
      run: |
        python -m pip install --upgrade pip
        pip install -r requirements.txt

    - name: Run tests
      id: test
      run: |
        pytest

    - name: Set output variable
      id: check_tests
      run: echo "Tests passed!"
      if: steps.test.outcome == 'success'
```



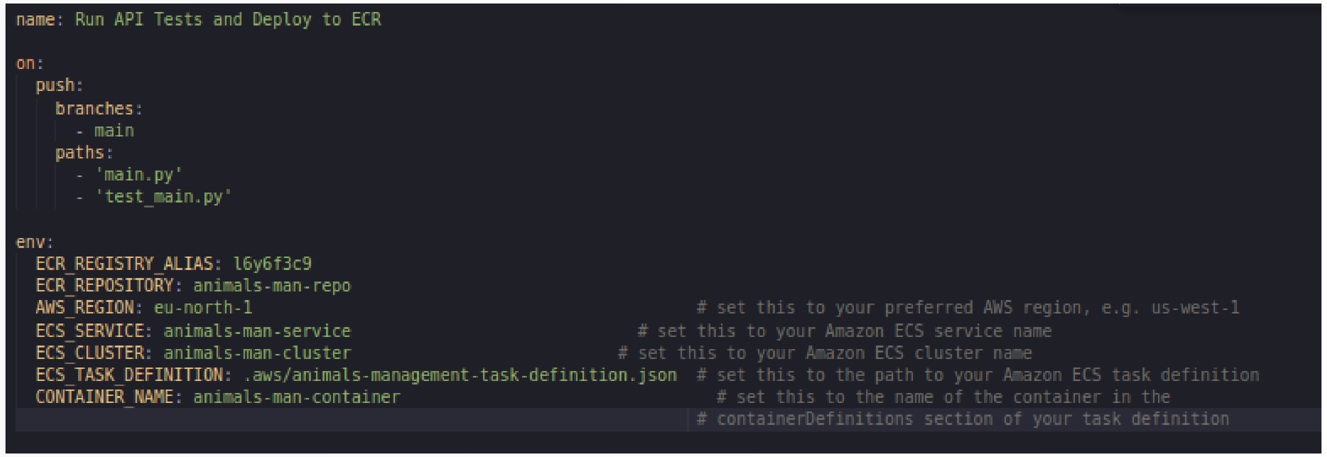
Coverage report: 80%

coverage.py v7.3.2, created at 2023-12-17 20:19 +0000

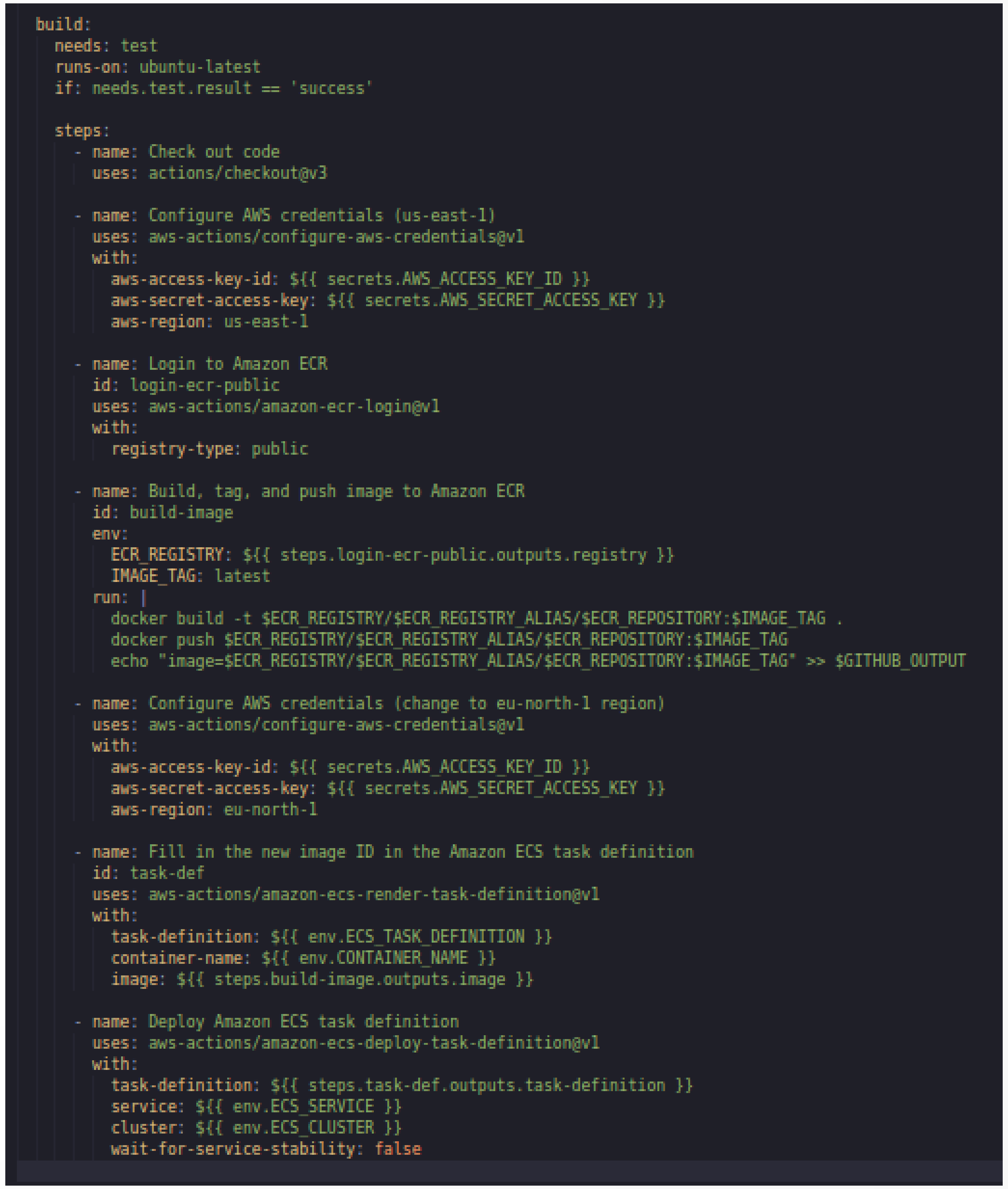
Module	statements	missing	excluded	coverage
main.py	244	49	0	80%
Total	**244**	**49**	**0**	**80%**

coverage.py v7.3.2, created at 2023-12-17 20:19 +0000

CD - Deployment



```yaml
name: Run API Tests and Deploy to ECR

on:
  push:
    branches:
      - main
    paths:
      - 'main.py'
      - 'test_main.py'

env:
  ECR_REGISTRY_ALIAS: l6y6f3c9
  ECR_REPOSITORY: animals-man-repo
  AWS_REGION: eu-north-1                                    # set this to your preferred AWS region, e.g. us-west-1
  ECS_SERVICE: animals-man-service                       # set this to your Amazon ECS service name
  ECS_CLUSTER: animals-man-cluster                     # set this to your Amazon ECS cluster name
  ECS_TASK_DEFINITION: .aws/animals-management-task-definition.json  # set this to the path to your Amazon ECS task definition
  CONTAINER_NAME: animals-man-container                       # set this to the name of the container in the
                                                                    # containerDefinitions section of your task definition
```



```yaml
build:
  needs: test
  runs-on: ubuntu-latest
  if: needs.test.result == 'success'

  steps:
    - name: Check out code
      uses: actions/checkout@v3

    - name: Configure AWS credentials (us-east-1)
      uses: aws-actions/configure-aws-credentials@v1
      with:
        aws-access-key-id: ${{ secrets.AWS_ACCESS_KEY_ID }}
        aws-secret-access-key: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
        aws-region: us-east-1

    - name: Login to Amazon ECR
      id: login-ecr-public
      uses: aws-actions/amazon-ecr-login@v1
      with:
        registry-type: public

    - name: Build, tag, and push image to Amazon ECR
      id: build-image
      env:
        ECR_REGISTRY: ${{ steps.login-ecr-public.outputs.registry }}
        IMAGE_TAG: latest
      run: |
        docker build -t $ECR_REGISTRY/$ECR_REGISTRY_ALIAS/$ECR_REPOSITORY:$IMAGE_TAG .
        docker push $ECR_REGISTRY/$ECR_REGISTRY_ALIAS/$ECR_REPOSITORY:$IMAGE_TAG
        echo "image=$ECR_REGISTRY/$ECR_REGISTRY_ALIAS/$ECR_REPOSITORY:$IMAGE_TAG" >> $GITHUB_OUTPUT

    - name: Configure AWS credentials (change to eu-north-1 region)
      uses: aws-actions/configure-aws-credentials@v1
      with:
        aws-access-key-id: ${{ secrets.AWS_ACCESS_KEY_ID }}
        aws-secret-access-key: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
        aws-region: eu-north-1

    - name: Fill in the new image ID in the Amazon ECS task definition
      id: task-def
      uses: aws-actions/amazon-ecs-render-task-definition@v1
      with:
        task-definition: ${{ env.ECS_TASK_DEFINITION }}
        container-name: ${{ env.CONTAINER_NAME }}
        image: ${{ steps.build-image.outputs.image }}

    - name: Deploy Amazon ECS task definition
      uses: aws-actions/amazon-ecs-deploy-task-definition@v1
      with:
        task-definition: ${{ steps.task-def.outputs.task-definition }}
        service: ${{ env.ECS_SERVICE }}
        cluster: ${{ env.ECS_CLUSTER }}
        wait-for-service-stability: false
```

Future Work

- Online chat
- A better payment system
- Increase the visibility of our application
- Improve CI/CD pipeline and terraforms scripts.

Demonstration

- https://main.dzgh2fc7t2w9u.amplifyapp.com/