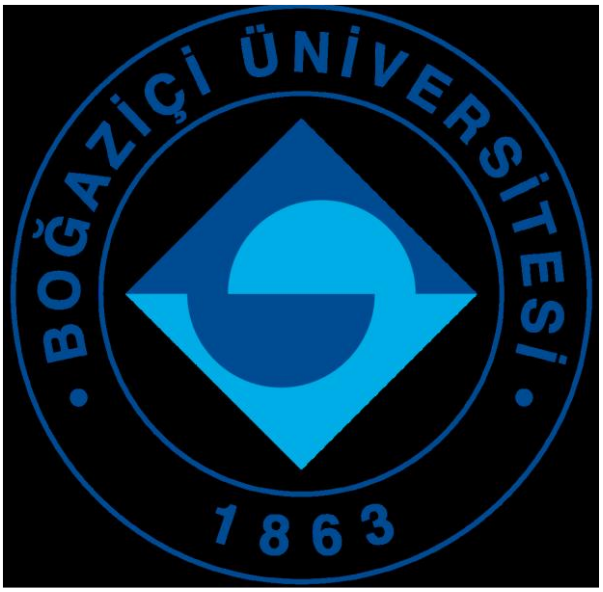


SWE573 SOFTWARE DEVELOPMENT PRACTICE

M. Görkem KUYUCU

29.05.2023

Project Name: OnceUponATime

Git Repository: https://github.com/gorkemkuyucu/SWE573-Spring23

Git Tag Version: v0.9

Deployment URIs: GCP= http://34.141.119.221:8000/

Video: https://drive.google.com/drive/folders/1Jt2cYWULnLuMRFosbVc1-oqE87UzxQpF?usp=sharing

HONOR CODE

Related to the submission of all the project deliverables for the Swe573 2022 Fall semester project reported in this report, I <Enter Your Name> declare that:

- I am a student in the Software Engineering MS program at Bogazici University and am registered for Swe573 course during the Spring 2023 semester.
- All the material that I am submitting related to my project (including but not limited to the project repository, the final project report, and supplementary documents) have been exclusively prepared by myself.
- I have prepared this material individually without the assistance of anyone else with the exception of permitted peer assistance which I have explicitly disclosed in this report.

M. Görkem Kuyucu

TABLE OF CONTENTS

1. Overview

This project was prepared for the Sprin 2023 Swe 573 Software Development Practice class. Our aim for developing the project is to gain experience of a full stack project cycle from start to finish, including dockerization and deployment. Through the development period, we interviewed potential members to extract stories and essential needs. With those the requirements are formed and their specification included mockup design. Following that, a web application is designed using Django. Lastly our process is finished by dockerization and deployment of the application using a cloud service provider. Second cloud service provider is also experienced. Git was used to follow up development process and to upload the code to the cloud.

Purpose of the application is to give users a web platform to share memories with their geolocations. Members can follow others, read, like, comment on stories and search geolocations using circular area additional to searching by their content.

2. Software Requirements Specification

2.1 Interview Extraction

First story is from Gülfer Kuyucu Meriç about Koç University and area around it. Her experience started in 2004 as a student and continued as an teaching assistant until 2015. After this date, she visited the area when passing by or to meet some friends. First she mentioned the family visits to Sarıyer for region-specific pastries. The road was so long from her residence at that time and traffic jam starts from coastal road (with exact words “from Aşk-I Memnu Yalısı”) because of the traffic inside Sarıyer. During this period only road reaching to Koç University passed from there. The traffic jam was usually continued till the graveyard. After that, the climbing road on which cows were grazing or crossing, starts. The gendarme was responsible from the area. There was no lighting, no market or whatsoever on this country like road. This is why traffic accidents were normal in this road. She remembered only two sellers; Dominos and other one who sells frozen pizzas from Superfresh or homemade salads. Inside the university, there were three restaurants called Mios, Yedi and Suzy’s. After graduation one of her friend’s opened a café called SOS. Tankut Cenkter, dean of law faculy was hired her as an assistant. There was an area they called area of sites with only 3 sites. When they wanted to go out for a restaurant, they were going to Anzer Sofrası near Hacıosman Metro. After starting as an assistant, she mentioned from opening of Café Nero as a milestone event. Another milestone was the tunnel from Hacıosman which bypasses the traffic in Sarıyer. Day by day the road from tunnel to KU is filled with buildings and sites. During this period, university had some lodgments inside the campus but since they were insufficient university rented some houses in these sites. This is why “Batı Yurdu” was constructed in the site area mentioned before. She worked until January 2015. After that even Sushico and Starbucks is opened. During her visit in December 2021, she mentioned she did not recognize the area and roads she was passing because of this changes. She also mentioned selections are increased unbelievably. When she was first enrolled, it was mentioned that even hammering nails is forbidden in the campus. But after some time, a new faculty and a new dormitory were constructed. She also mentioned about dorms. During her stay, there were no restrictions other than alcohol prohibition. The entrance and leaving is not controlled, there was no gender discrimination and even her friends who did not stay in dorms can visit and stay. And there was no inspection about that. As she followed from social media, coed dorm was changed to female

dormitory and some students eating mixed in communal kitchens got disciplinary punishment after detecting them on camera. She also mentioned that these changes were made because of YÖK regulations. Ümran Aşkar were the rector while she was assistant.

2.2 Glossary

A guest is an unregistered user of the platform.

A member is a registered user of the platform.

A user is a guest or a member using the platform.

An admin is a privileged member who can audit and edit other user's accounts and stories.

A story or memory is a text shared by a member.

2.3 Requirements

2.3.1. System Requirements

2.3.1.1. The platform shall be available only on the web. COMPLETED

2.3.1.2. The platform's response time shall not exceed 3 seconds. COMPLETED

2.3.1.3. The platform shall show a maximum of 10 stories on a page. IN-COMPLETED

2.3.2. Register, Login, and Logout

2.3.2.1. A user shall be able to register on the platform. COMPLETED

2.3.2.2. A user shall be able to use unique nickname and e-mail for registering. COMPLETED

2.3.2.3. A user's nickname shall contain only alphanumeric characters. COMPLETED

2.3.2.4. The platform shall send a verification code when a user is signed up. IN-COMPLETED

2.3.2.5. If the verification code is not matched from the database, an error message shall be sent to the user. IN-COMPLETED

2.3.2.6. The user shall access the platform as a member after verification code is matched with the send one. IN-COMPLETED

2.3.3. Search

2.3.3.1. A user shall be able to search and filter published stories by using story name or publisher's nickname or relevant date or radius of a geolocation. COMPLETED

2.3.3.2. A user shall be able to select sorting of the published stories. IN-COMPLETED

2.3.3.3. If the search result is more than 10 stories, the platform shall divide it to pages. IN-COMPLETED

2.3. 4. Story

2.3.4.1. A story shall contain title, marked geolocation, share's nickname, mentioned date interval and created date. COMPLETED

2.3.4.2. A story shall contain a maximum of 5000 characters. COMPLETED

2.3.4.3. A member shall be able to publish their own stories. COMPLETED

2.3.4.4. A member shall be able to save a draft of a story. SEMI-COMPLETED

2.3.4.5. The member who published a story shall be able to edit it. COMPLETED

2.3.4.6. The user who published a story shall be able to assign tags to it. COMPLETED

2.3.4.7. A member shall be able to comment and like a story. COMPLETED

2.3.4.8. Every member shall be able to like a story for one time. COMPLETED

2.3.4.9. A member shall be able to unlike a liked story. COMPLETED

2.3.4.10. A user shall be able to observe published stories. COMPLETED

2.3.4.11. When a story is reported by a member, it is signed with "reported" flag. IN-COMPLETED

2.3.4.12. When a story is flagged with "reported" tag, admin is notified. IN-COMPLETED

2.3.4.13. A member shall be able to add pictures to a story via website link. COMPLETED

2.3.4.14. An admin shall be able to delete a story. COMPLETED

2.3.4.15. An admin shall be able to delete a comment. COMPLETED

2.3.4.16. A member shall be able to select date resolution for their stories. COMPLETED

2.3.4.17. Date resolution of a story shall be single date, date interval, single month, single year, season or decade. COMPLETED

2.3.5. Member Profile

2.3.5.1. A member shall be able to see how many likes are given to stories. COMPLETED

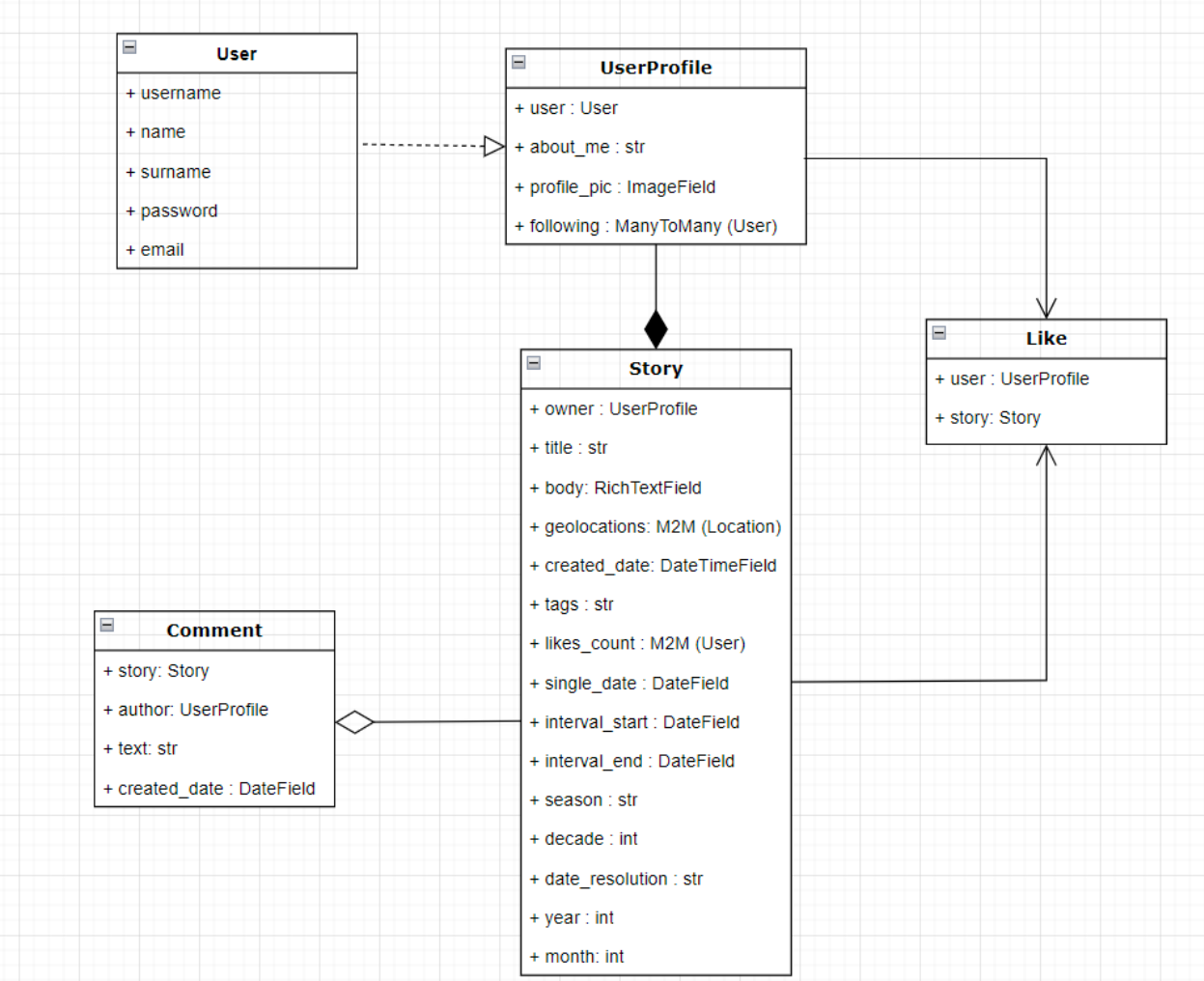
2.3.5.2. A member shall be able to add a profile photo to his/her profile. COMPLETED

2.3.5.3. A member shall be able to follow other members. COMPLETED

3. Mock-ups and UML Diagrams

Mock-ups designed using Figma. They can be observer using the link: https://www.figma.com/file/Fbpxz8dDKd1nZGmJSMYJpU/Once-Upon-A-Time?node-id=2%3A409&t=DaZzhn1PYag42zjO-1

UML Diagram is drawn as follows.



4. Dockerization and Deployment

3 files are needed for dockerization:

requirements.txt:

```
Django>=4.1.7,<4.2
psycopg2>=2.9.6,<2.10
pillow
django-taggit
django-ckeditor
geopy
psycopg2-binary
django-environ
python-dotenv
```

dockerfile:

```
FROM python:3.9.16-slim-buster

LABEL maintainer="onceuponatime.com"

ENV PYTHONUNBEFFERED 1

# Install dependencies (like GDAL and GEOS)
RUN apt-get update && apt-get install -y \
    binutils libproj-dev gdal-bin python-gdal python3-gdal \
    libgeos-dev python3-dev netcat postgresql gcc libpq-dev

COPY ./requirements.txt /requirements.txt
COPY ./ ./OnceUponATime

WORKDIR /OnceUponATime
EXPOSE 8000

RUN python -m venv /py \
    && /py/bin/pip install --upgrade pip \
    && /py/bin/pip install -r /requirements.txt \

ENV PATH="/py/bin:$PATH"
ENV CPLUS_INCLUDE_PATH=/usr/include/gdal
ENV C_INCLUDE_PATH=/usr/include/gdal
```

and docker-compose.yml:

```
version: '3.9'

services:
  app:
    build:
      context: .
      dockerfile: Dockerfile
    command: >
      sh -c "python manage.py wait_for_db &&
             python manage.py migrate &&
             python manage.py runserver 0.0.0.0:8000"
    ports:
      - 8000:8000
    volumes:
      - ./app:/app
```

```
    env_file:
      - .env
    depends_on:
      - db
    networks:
      - onceuponatime



  db:
    image: kartoza/postgis:15
    ports:
      - 5432:5432
    volumes:
      - postgres_data:/var/lib/postgresql
    environment:
      - POSTGRES_DB=${POSTGRES_DBNAME}
      - POSTGRES_USER=${POSTGRES_USER}
      - POSTGRES_PASSWORD=${POSTGRES_PASS}
    networks:
      - onceuponatime
    restart: "on-failure"

networks:
  onceuponatime:
    driver: bridge

volumes:
  postgres_data:
```

Since there is no private information on .env file, it is pushed to the repo also. After assigning Puclic IP of the server by adding it to the .env file the application containerized using termina command:

docker-compose up --build -d

PS: Since gdal is used, building this container for the first time could took higher than 1800 seconds on the local computer. GCP was a lot faster comparing to it.

Using Docker Desktop, container statuses are observed. The problem was the application was starting before the database. To solve this problem, an algorithm is used to check whether database is up or not. A makefile is written in order to make writing codes easy.

Makefile:

```
ifneq (,$(wildcard ./.env))
    include .env
```

```
    export
    ENV_FILE_PARAM = --env-file .env

endif

build:
    docker-compose up --build -d --remove-orphans
up:
    docker-compose up -d
down:
    docker-compose down
logs:
    docker-compose logs
migrate:
    docker-compose exec app python3 manage.py migrate --noinput
makemigrations:
    docker-compose exec app python3 manage.py makemigrations
superuser:
    docker-compose exec app python3 manage.py createsuperuser
volume:
    docker volume inspect OnceUponATime_db
run:
    docker-compose exec app python3 manage.py runserver
```

To deploy the project, a VM on cloud service provider is created. Port used in the project which is 8000, is added to the firewall. Then public ip address is written in hosts of the .env file. Following to that, required problems are installed on the VM. Code is cloned to the VM from git and docker container is composed in the machine. It is tested using VM's public ip addres and allowed port(8000)

Sample stories are uploaded to the GCP. (http://34.141.119.221:8000/story/list/)

5. Manual and User Tests

5.1 User Manual (Q&A)

How to Signup? In order to share memories, like and comment user should sign up.

1. Go to the sign up page: http://34.141.119.221:8000/sign_up/

2. Fill relevant areas: username, first name, last name, email, verify email, password, password confirmation. Check the terms and conditions checkbox.

3. Click to the sign up button.

How to Login? After having an account allows user to log in.

1. Go to the login page: http://34.141.119.221:8000/sign_in/

2. Fill username and password area.

3. Click sign in button.

How to See Posts? To see other users' posts click to Story List in the footer. (http://34.141.119.221:8000/story/list/)

How to use basic search on stories? Basic search allow one to filter stories .

1. Type what you wanted to search to the search box in the navbar.

2. The stories containing written charactes in their author or body or title or tags will be showed.

How to use advanced search on stories? Advanced search will allow you to perform more thrilled search.

1. Click advanced search button near the search box.

2. Fill the specific areas you want to find and click "Advanced Search" button to get results.

How to use location search on stories? Search by location will give you the results of stories with the circular area you selected.

1.Click "I want to search by location" in the advanced search area.

2. In the left side, you will see a circle icon. Clicking it will allow you to draw a circle starting from the center.

3. After clicking second time to select the radius, obtained data will be shown in boxes below.

4. Click "Find stories in this area" button to get the results.

How to like and comment on stories?

1. Select a story by clicking on the title.

2. On the top, you can see the like button if you are not the author. Clicking it will help you to like or unlike a story.

3. On the bottom, there is a comment box. Writing the text field and clicking "Add comment" will allow you to send your comment under the story.

4. If you wanted to delete your own comment, you can use the "Delete comment" button appearing if you have comment.

How to See Member's Page & Follow them & Edit my own profile?

1. Click to the member list link on the footer or member nicknames near the titles of stories. (Member List: http://34.141.119.221:8000/profiles/)

2. Profile page opens. A member can follow or unfollow a member via this page.

3. If the member goes to own profile, Edit profile button on the top and Edit story button near relevant stories. Edit profile will allow a member to change About me part and upload a different profile photo. User can see, edit his/her own stories, followers and followings in this page.

How to delete a story?

1. Click a story title to read it.

2. If you are the author, delete story button will be appeared on top.

How to write a story?

1. Sign in the the platform.

2. After you signed in, a write story button on right side of the navbar will appear.

3. You can use the link on the footer also but an error will be occurred if you are not signed in!

4. Fill all the areas except tag. If you are not chosen at least one location or a date, form won't be accepted as valid and your story won't be created.

That's all for now. Have fun!

5.2 Test Credentials

Site address: http://34.141.119.221:8000/

Superuser

Nickname: gorkem

Password: asdasd

Regular user 1

Nickname: gulferm

Password: Asd.Qwe1

Regular user 2

Nickname: suzkudarli

Password: Asd.Qwe1

Test case 1: Sign up

Sign up is tested using different user information. Both test accounts are created using this form.

Test case 2: Sign in

Sign in is tested with correct info and fault password. Correct one allowed us to sign in and faulty one gave error message.

Test case 3: Navigating

Navigating through the site using links on the navbar and footer is tested. For authenticated user links performed successful. For unauthenticated user, it is tried to reach edit story page using pk and edit story page which do not allow and show an error message was also successful.

Test case 4: Search

Basic search gave the seeked information on the query. For the advance search, while testing location search, a bug (showing same story with multiple location in selected area) is occurred. The issue was created but this could not be fixed due to lack of time.

Test case 5: Writing a story

Sending writing a story form without filling necessary areas resulted showing error messages. Filling all the essential spaces allowed us to post the story form successfully show us a success message.

Test case 6: Deleting a story

On the read page with other authors, delete story button is not seen. When the signed in profile is the author, delete story button is showed and performed succesfuly.

Test case 7: Liking and Commenting on a story

On the read page with other authors' stories, like button is seen and performed succesfuly. Like and unlike is changed with the number of likes accordingly. Comment section allowed a member to post a comment and delete button only shown and performed successful when the member observing the story is alt author of the comment.

Test case 8: Following another profile

Member profiles are only visible to the authenticated users. When a member visits another member's profile follow button is appears. It is changed to unfollow if clicked and the member clicking the follow button appears on the list in member profile. An error message is shown when signed in member is tried to follow their own profile which is the expected result therefore successful.