REPORT

WEB ENGINEERING - 206

PHOTO EDITING SYSTEM

Prepared By

Syed Muhammad Muneeb (SE-019)

Fatima Younus (SE-042)

Aasia Zahir (SE-044)

Link to Github:

https://github.com/syedmuhammadmuneeb/syedmuhammadmuneeb.github.io

Link to Azure:

https://php-project.azurewebsites.net/userlogin.php

REQUIREMENT SPECIFICATIONS:

FUNCTIONAL REQUIREMENTS:

Photo Editor:

This is a web based photo editing app that allows you to edit your .User can also use this editor as graphic editor as you can create canvas by inputting the desire size for it and using drawings tools and gradients.

Photo Upload:

User can upload picture to be edited from their computer on the application or you can edit using default images available.

Photo Format:

User can use PNG and JPEG format images. At a time more than one PNG format images can be added on a single JPEG image.

Editing Options:

Users can do basic editing like cropping, rotate resizing and changing the background color, for more than one photo you can also merge them into one. You can also simply drag your images over canvas. User can apply range of built in features as well as they can write text over it.When user wants to use it as graphic editor there are various drawing tools that allows you make different shapes and use brushes to create various stuff.

NON-FUNCTIONAL REQUIREMENTS:

Usability:

This application is user friendly as it allows everyone to use it's every feature.

Reliability:

Since the data i.e. your edited images are uploaded directly from your PC instead of accessing your email to use your photos it is safe and reliable because the image edited or drawing made is directly downloaded to your computer instead of saving in any database that can cause trouble in case database is compromised.

Portability:

This application is portable as it can work both for Windows and MAC OS.

IMPLEMENTATION:

For making this application we used HTML5 and JavaScript. As a result this application is very responsive. Since we have used HTML 5 so it's better to use it in modern browsers.

We have used MySQL from XAMPP and PHP for managing database of information provided by user at the time of logging in or registering themselves.

TESTING ACTIVITIES:

Title: Merging Images – Adding different images on main image to act as one.

Description: A user should be able to see all images added as "MAIN IMAGE" in layer option.

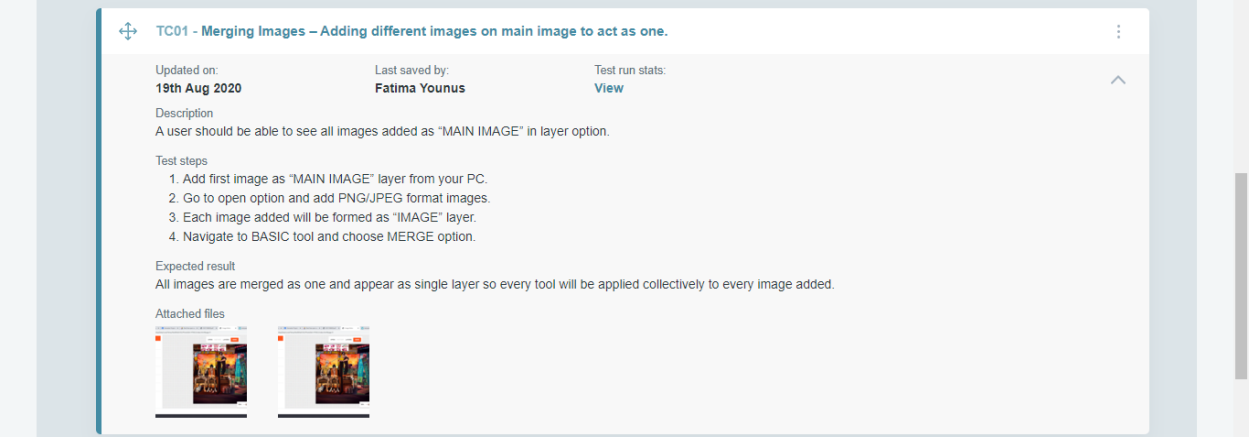
Pre-condition: The user must upload main image of larger resolution than images to be merged.

Assumption: All images will be edited as one now.

Test Steps:

1. Add first image as "MAIN IMAGE" layer from your PC.
2. Go to open option and add PNG/JPEG format images.
3. Each image added will be formed as "IMAGE" layer.
4. Navigate to BASIC tool and choose MERGE option.

Expected Result: All images are merged as one and appear as single layer so every tool will be applied collectively to every image added.



PLANNING:

The first and very most thing to be considering while planning our photo editor is to estimate the time and effort required. The estimated time for the project to be built is 2 months with each member equally participating in respective activities and tasks.

Increment planning:

The project is built keeping in mind the framework activities. 3 increments are to be done in total.

Risk analysis:

Unlike the editing apps which we can download on our phone, most of the online photo editing has a risk associated with it that even if you locally delete your image the image is still there on the server which might be modified or misused by any hacker.

But while constructing our site we've kept in our mind our users' privacy and hence the editor will not save the pic in the server once the image is delete locally it's deleted completely.

GUI:

After the risk analysis. GUI including color theme, navigation, menu bar and the features to be added in our editor were discussed and negotiated.

Database:

To store user's info a database would be required.

Scheduling:

Group meeting was scheduled for 2 times a week.in which every member will propose their success they made that particular week.

DEPLOYMENT:

Database Deployment:

The database is deployed on azure. Following steps were taken:

- Grant public access to it for our frontend IP.
- connect to the database from MySQL Workbench using the connection string, port, user and password.
- Use this configuration in our application to be able to fetch data from it from the code.

Backend Deployment:

The backend composes of serverless azure functions. Following steps were taken to deploy:

- Create a project and install the dependencies.
- Create a profile on azure console for the project.
- In the terminal, run azure configure –profile <<profile_name>>, and enter the details.
- Write some code and define the routes in the serverless.yml file.

Frontend Deployment:

The frontend is deployed on azure with auto deploy on push to master branch. This was achieved using GitHub Actions. Following steps are required:

- Create a new project on the azure console.
- Create a GitHub Actions file and use the token in that file.
- The file runs when ever changes are pushed to master branch and the code is deployed to azure.

Evaluation:

code is uploaded on Github and exposed for the users to evaluate the editor in terms of verification and validation. Prior to this it was used by our group members to identify any possible errors lying in the front or backend.

Users after evaluating it will provide their feedback on it whether this editor is helpful or not. Based on the users' feedback project will be modified and maintained.

FRAMEWORK ACTIVITIES:

ITERATION 1:

Communication:

Platform for uploading pictures.

- We asked users what source to they want the Photo Editing Web App to get their pictures from..
- Most of the users said they want to upload from Google Photos as it's convenient and easy while other users also preferred to use Google Drive etc.
- We suggested uploading pictures directly from their devices.
- We negotiated users that it's better to use their Device directly because their personal information in case of attack on Photo Editor Data can be compromised.

User agreed as this was more secure for editing their personal pictures.

ITERATION 2:

Communication:

Color scheme and login form:

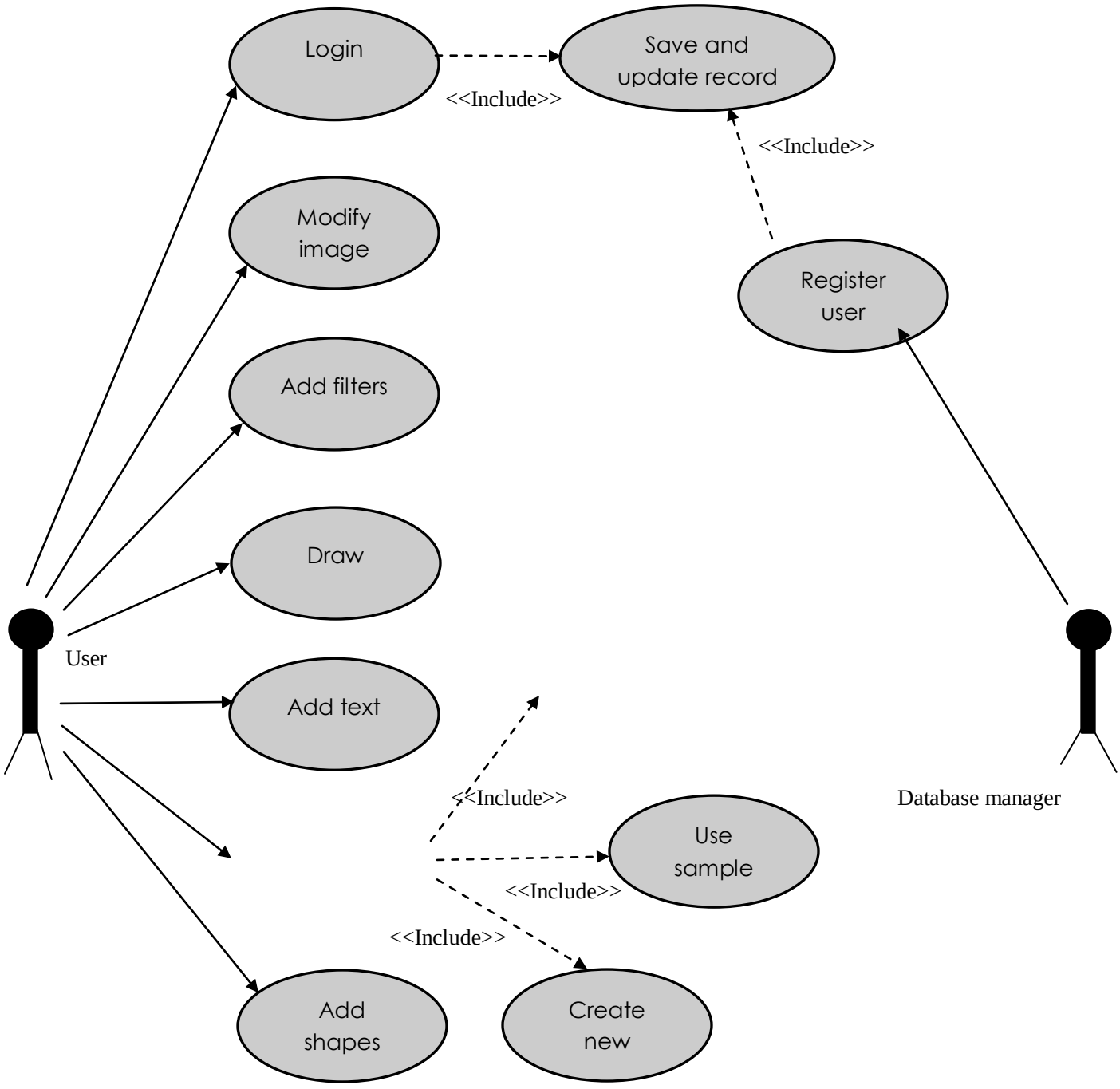
In second meeting with stakeholders we discussed the following points:

- Photo editor would comprise of a login form which would allow user to enter their email and password before they could start with the editing app. The color scheme should be neutral and sober.

Planning:

- Dark gray and light gray are the main colors for the app while red color is used for some of the buttons to provide a contrast.
- Login form would be made using HTML and CSS for styling. A database would be made to store the emails and passwords of the users.

USE-CASE DIAGRAM:


Login
Save and update record
<<Include>>
<<Include>>
Modify image
Register user
Add filters
Draw
User
Add text
Database manager
<<Include>>
Use sample
<<Include>>
<<Include>>
Add shapes
Create new