



Aibek Aldabergenov

SWE573 Software Development Practice, Spring 2024

May 20, 2024

Project name: CBIS – Community-based information system

Deployment URL: <https://cbis.duckdns.org/>

Git Repo URL: <https://github.com/aaldaber/SWE573-Spring-2024/>

Git Tag URL: <https://github.com/aaldaber/SWE573-Spring-2024/releases/tag/v0.9/>

HONOR CODE

Related to the submission of all the project deliverables for the Swe573 2024 Spring semester project reported in this report, I Aibek Aldabergenov declare that:

- I am a student in the Software Engineering MS program at Bogazici University and am registered for Swe573 course during the Spring 2024 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.

Aibek Aldabergenov

OVERVIEW	3
SOFTWARE REQUIREMENTS SPECIFICATION	3
User Login and Registration	3
Communities	3
Moderation	4
Templates and posts	4
DESIGN DOCUMENTS	5
Logo	5
Use Case Diagram	5
System Interface Mockups	6
STATUS OF THE PROJECT	7
STATUS OF DEPLOYMENT	9
INSTALLATION INSTRUCTIONS	10
UNIT TESTS	10
DEMO VIDEO	10

Overview

This report discusses the development of a community-based information system developed during the SWE573 course. The name of the project is CBIS, and the project is deployed at <https://cbis.duckdns.org/>.

If you want to quickly test the deployed system, you can use the following credentials to login as the test user:

email: testuser@local.local

password: hAphisII

CBIS is a community-based information system. The aim of this project is to assist people in creating various communities and making it easier for users to discover information within the interested communities.

Software Requirements Specification

Below is a list of Software requirements that were formed before the development of the system started.

User Login and Registration

1. A user shall be able to register on the platform using email and password.
2. After registration, a user shall be able to login to the system using email and password.
3. After logging in, a user shall be able to go to his/her profile page and edit profile details (display name, avatar).

Communities

4. The platform's frontpage shall display the mix of the following communities: popular, recently active and newly created.
5. A user shall be able to create a new community.
6. A user shall be able to choose if the community is public or private during creation.
7. A user that created the community is the owner of that community.
8. The community's owners shall be able to invite other users to become the owners.

9. The community's owners shall be able to invite other users to become the moderators.

Moderation

10. Community moderators shall be able to publish community rules when creating communities or when editing them.
11. Community moderators shall be able to delete posts that break the community rules.
12. Community moderators shall be able to ban users that break the community rules.
13. Community moderators shall be able to remove comments that break the community rules.

Templates and posts

14. In a community, the most basic type of data that can be posted is a textual message.
15. Community moderators shall be able to create templates for posts.
16. Post templates shall be customizable to include such types of data as: image, video, url, document/file, date/time, gps coordinate, label/tag.
17. Post templates shall be customizable to include various fields such as: input text, checkbox, radio button, dropdown menu.
18. Each community shall have its own set of templates.
19. After a community moderator creates a template, that template shall be available to regular users for creating new posts.
20. In a community, a user shall be able to sort the posts by most viewed, most commented, and most starred.
21. By default, the posts are sorted by date of publication.
22. A user shall be able to star a post he/she likes.
23. A user shall be able to write a comment under a post.
24. Users shall be able to see all their starred and commented posts in the My Profile section.
25. A user shall be able to filter and sort the starred and commented posts in the My Profile section by date.
26. A user shall be able to see own post history sorted by date.

Design documents

Logo

The logo for the CBIS system was generated using Microsoft's AI image generator at: <https://designer.microsoft.com/image-creator/>.



Figure 1 CBIS logo

Use Case Diagram

The following use case diagram represents the separation of roles and actions available for each role.

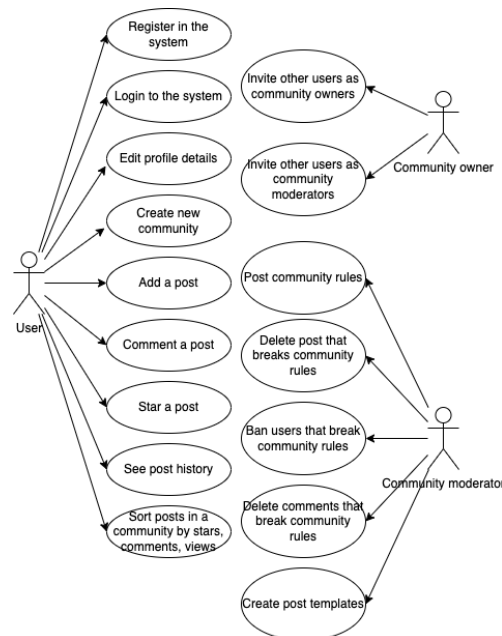


Figure 2 Use case diagram

System Interface Mockups

The mockup shows a window titled "CBIS - Registration". Inside, the word "Registration" is centered. Below it are two input fields: "Email" and "Password". At the bottom center is a "Submit" button.

Figure 4 Registration form

The mockup shows a window titled "CBIS - Login". Inside, the word "Login" is centered. Below it are two input fields: "Email" and "Password". At the bottom center is a "Submit" button.

Figure 3 Login form

The mockup shows a window titled "CBIS - Feed". On the left is a search bar "Search communities". Below it are three community links: "Discover communities", "Gadgets - a community about news and discussion of gadgets", "Alnews - everything about AI and ML", and "Watches - a community of watch enthusiasts". On the right is a user profile for "Aibek Aldabergenov" with "Edit Profile" and "New community" buttons. At the bottom right is a "Logout" button and a sidebar with "About CBIS", "FAQ", and "Careers".

Figure 5 CBIS main page

The mockup shows a window titled "CBIS - New community". On the left is a "Create a new community" section with a "Set avatar" button and three input fields: "Enter community name", "Enter community description", and "Enter community rules". Below these is a "Private" toggle and a "Submit" button. On the right is a user profile for "Aibek Aldabergenov" with "Edit Profile" and "New community" buttons. At the bottom right is a "Logout" button and a sidebar with "About CBIS", "FAQ", and "Careers".

Figure 6 Create a community

The mockup shows a window titled "CBIS - Manage community RetroCars". On the left is a sidebar with a car icon, "RetroCars" title, description, rules, and buttons for "New post" and "Post types". The main area has three tabs: "Most viewed", "Most commented", and "Most starred". Below "Most viewed" is a car image and a "Toyota Imaginarium" title with "info", "toyota", and "60s" tags. Below this are several horizontal lines representing text. On the right is a user profile for "Aibek Aldabergenov" with "Edit Profile" and "New community" buttons. At the bottom right is a "Logout" button and a sidebar with "About CBIS", "FAQ", and "Careers".

Figure 8 Community page

The mockup shows a window titled "CBIS - Manage community RetroCars". On the left is a sidebar with a car icon, "RetroCars" title, description, rules, and buttons for "New post" and "Post types". The main area has a "Post Types" section with "Basic", "RetroCarInfo", and "CarHistory" sub-sections. Below these are several horizontal lines representing text. At the bottom is an "Add new post template" button. On the right is a user profile for "Aibek Aldabergenov" with "Edit Profile" and "New community" buttons. At the bottom right is a "Logout" button and a sidebar with "About CBIS", "FAQ", and "Careers".

Figure 7 Template management in a community

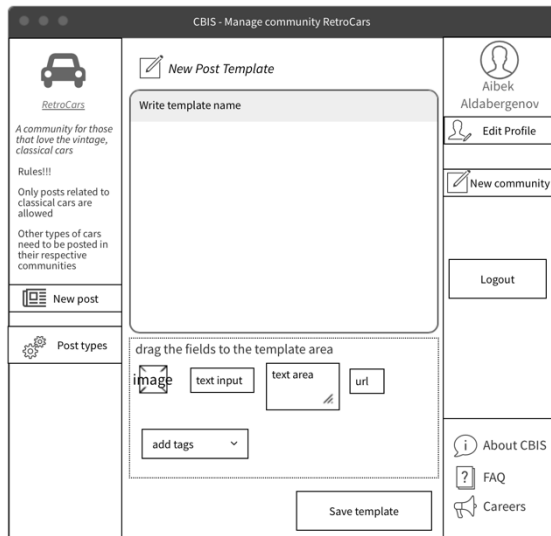


Figure 9 Create a new template

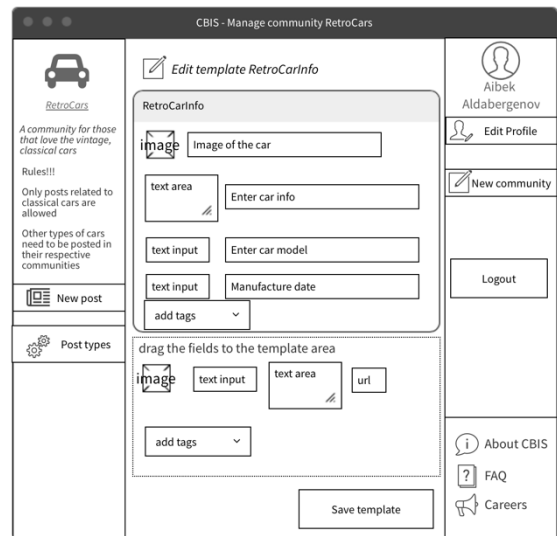


Figure 10 Template creation by dragging fields

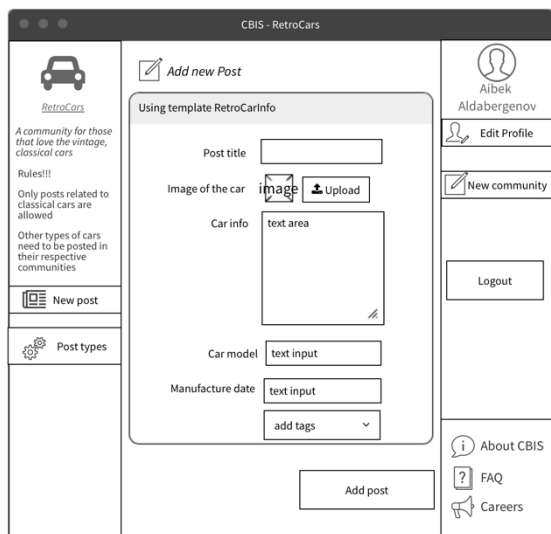


Figure 11 Create a post using new template

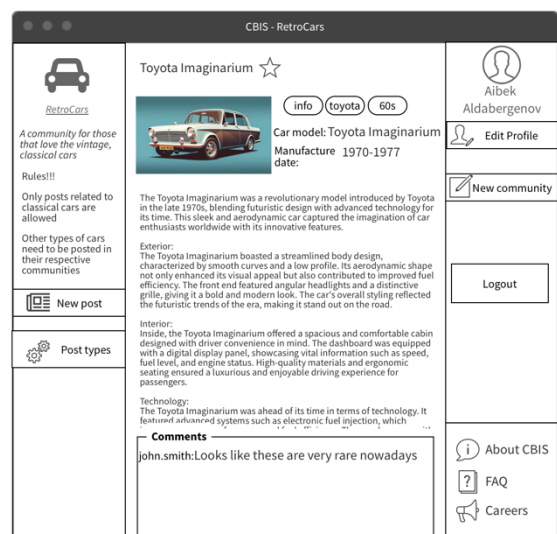


Figure 12 Post detail

Status of the project

- User Login and Registration – the functionality of user registration and login is fully functioning. If a user does not fill the username field, the system will generate a random UUID string as the username. After login, a user can edit username, about me and avatar fields of the profile.
- Communities – users can create new communities with name, description and community picture fields. A user that created the community is automatically assigned as its owner. Also, community owners can edit community details. The function of adding new community owners, or community moderators is not yet developed.
- Moderation – none of the functionality related to community moderation is developed at the moment.

- Templates and posts – when a community is created a default template with title and description text fields is added to a community. Community owners are also able to create custom templates with the following fields: text field, text area, integer, float, true/false checkbox, date and datetime selection field, file and image upload field, and a map interface for selecting geolocation coordinates. The template creation page has a drag and drop interface for convenient field reordering. After a template is created, community users can use the template to add new posts. On the upper right side of the system interface, users can search the communities by name and description fields. For searching posts, community users can use advanced search functionality where they can search for posts using posts' template fields. The system's index page shows the latest 10 posts ordered by creation date, most viewed posts in the last day, and the most viewed posts of all time. Starring posts, commenting posts, and post history in user's profile are not developed at the moment.

Home Communities

Search (min. 3 chars) Go

Welcome, aibek!

Edit profile

Create a community

Logout

New template

Template name

Custom template

Availabe fields

- Text
- Text Area
- Integer
- Float
- True / False
- Date
- Datetime
- File
- Image
- Geolocation

Place the needed fields to this space

Remove

Label: What

Required: ☒

Text

Text Area

Remove

Label: When

Required: ☒

Datetime

Remove

Label: Where

Required: ☒

Geolocation

Submit

Figure 13 Custom template creation

Home Communities

Search (min. 3 chars) Go

Welcome, aibek!

Edit profile

Create a community

Logout

Add post

Post title

To all Lexus owners

What

Gathering to show our Lexus cars

When

05/31/2024, 15:00

Where

Submit

Figure 15 Post creation using a custom template

Home Communities

Search (min. 3 chars) Go

Welcome, aibek!

Edit profile

Create a community

Logout

Search posts using template

What Where When

Post title

What

When from

05/01/2024, 15:23

When to

06/08/2024, 15:23

Search

Your search results will appear here

To all Lexus owners

May 20, 2024, 3:22 p.m.

by aibek

1

Figure 14 Advanced search using template fields

Status of Deployment

The system is deployed at <https://cbis.duckdns.org/>. Deployment consists of three containers:

- web – developed in Django, contains the backend logic and forms
- nginx – is used to serve static files and acts as the proxy server to backend
- db – PostgreSQL database is used to hold the data of the system

Installation instructions

You will need Docker (<https://docs.docker.com/get-docker/>) and Docker Compose (<https://docs.docker.com/compose/install/>) installed on your system.

Edit `.db.env.sample` and `.env.sample` files according to your needs, and rename the files to `.db.env` and `.env`

From the project root directory, run *`docker compose up -d`*

After the images are built and containers are created, the system will be available at `http://localhost:8080/`

Build search indexes using *`docker compose exec web python manage.py buildwatson`*

You can create a superuser using *`docker compose exec web python manage.py createsuperuser`*

Unit Tests

Tests for the following functions are available in the `tests.py` file:

- community creation
- edit community
- default template creation
- custom template creation
- post creation using custom template

Demo Video

https://drive.google.com/file/d/1g9Jm_AVmqtVxIHP5CsGBoY_PNu7RIuIC/view?usp=share_link