

Software Architectures

Assignment 2: MVC using Scala Play

Ahmed Zerouali, Camilo Velazquez
Emails: {azeroual, cavelazq}@vub.be
Office: {10F725}

Assignment

The goal of this assignment is to implement a Model-View-Controller (MVC) application using Scala Play.

Deadline: August 15th 2021 by 23:59. The deadline is fixed and will not be extended for any reason.

Deliverables: A report and source code. The report (in English) explains your solution to the problem and the main components used in your implementation. When possible, use simple diagrams to illustrate.

The report should be handed in as a single PDF file of no more than 3 pages (excluding diagrams/screenshots/code). The file should follow the naming schema `FirstName-LastName-SA2.pdf`, for example: `Camilo-Velazquez-SA2.pdf`.

The source code is your implementation of the project zipped or exported from IntelliJ.

Submit the *report* and *source code* as a single zip file on the Software Architectures course page in Canvas, by clicking on *Assignments > Assignment 2*.

Plagiarism: Note that copying – whether from previous years, from other students, or from the internet – will not be tolerated, and will be reported to the dean as plagiarism, who will decide appropriate disciplinary action (e.g., expulsion or exclusion from the examination session). If you use any other resources besides those provided in the lectures and in this document, remember to cite them in your report.

Grading: Your solution will be graded and can become subject of an additional defense upon your or the assistants' request.

Problem Description

For this assignment you will implement a social networking website application where users can share their pictures so that other users can see them and like or comment on them. The implementation of the application should follow the Model-View-Controller pattern. The mockup in Figure 1 could be a motivational example for the front page of your website.

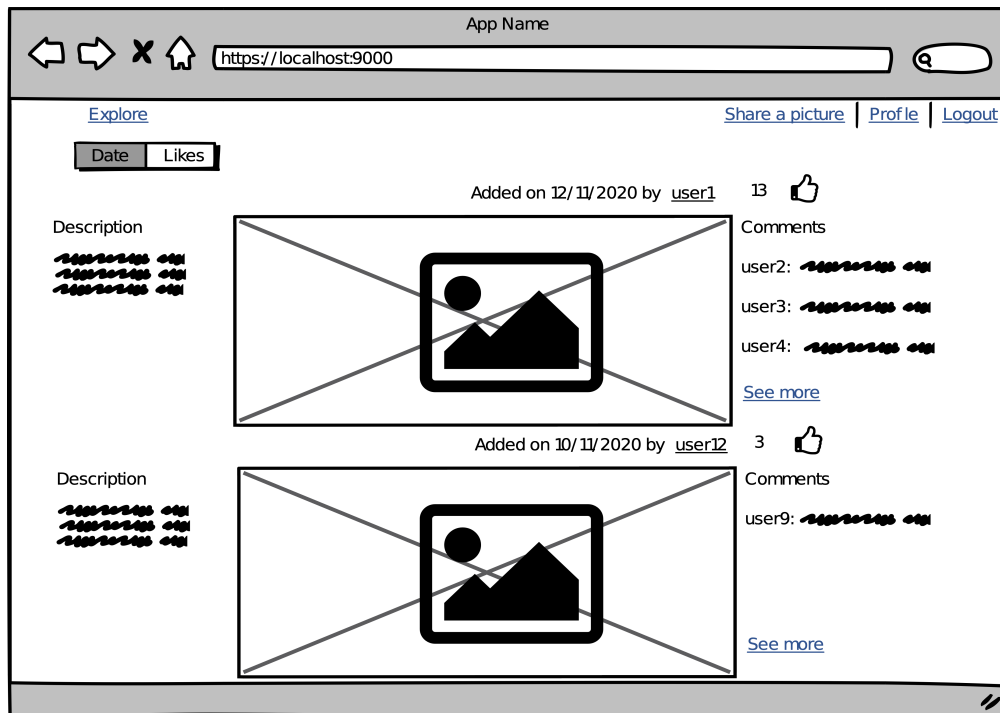


Figure 1: Front page of the website

As shown, you will have to design the layout of the website in a way that pictures are displayed in the middle of the screen, while description, likes and comments are at the left and right sides of the pictures. The pictures can be shown by their date of upload or their number of likes. In the comments section, the front page should only show a limited number of comments, to see more comments and see the figure in bigger size, you should create a “See more” link that will redirect the users to the page of the corresponding picture. A possible page could look like the mockup in Figure 2.

Only logged in users can see shared pictures. Additionally, they can make comments or like the pictures, as well as share new pictures.

To share new pictures on the website, you should implement a form that looks like the mockup in Figure 3. This form contains the mandatory fields such as the file upload input and text area, as well as the “Share” option. After selecting the picture to share, the user can select which audience can see it (“Share”). Once the picture is submitted, it can be seen by everyone or by a specific list of users that should be able to comment

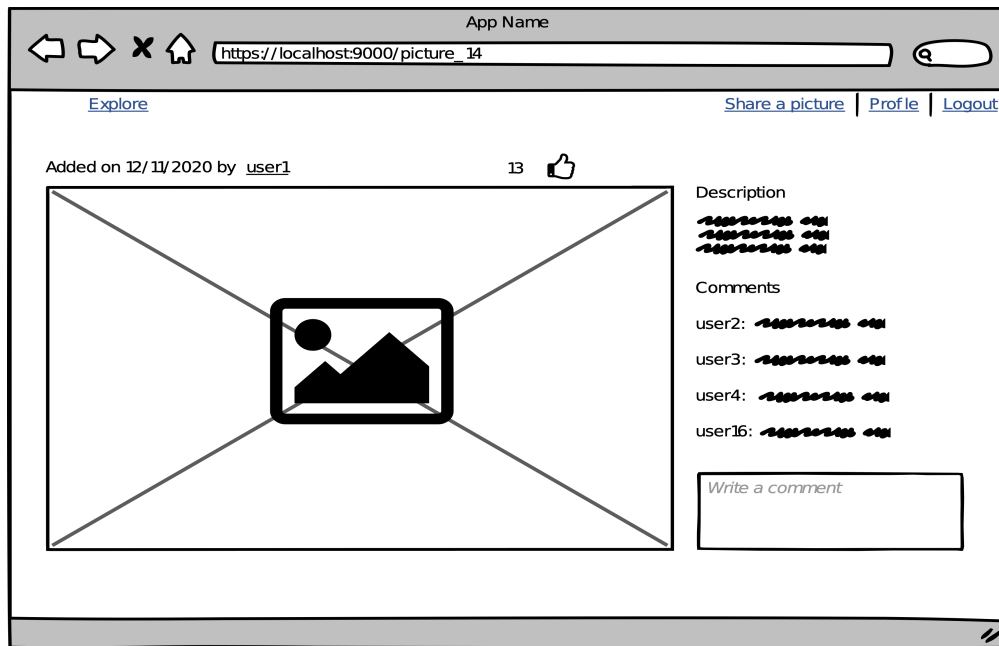


Figure 2: Picture in full size with all comments.

or like it. This choice should be made at the upload time. If a picture is uploaded, its poster can modify who can see it from their own profile only.

All users should be able to visit each other's profile. Moreover, logged-in users should be able to remove their own shared pictures when they access their profiles.

The elements "Explore", "Share a picture" and "Profile" should be hidden from the navigation bar at the top when users are logged out. You are also required to implement a page where unregistered users can sign up.

As a final recommendation, you can use any public template for your web page design (e.g. Twitter's Bootstrap). Do not use a database for your project, just keep the data in memory or store them in a json file. Pictures should be stored in a public folder within the project.

The above functionalities should be implemented using Model-View-Controller (MVC) pattern with Scala Play. In addition to the general approach, you will be evaluated according to how you implemented the different functionalities, views and controllers. You should take into account issues such as validations, error handling and any other abstractions that you deem fit for the scenario. Motivate your design decisions in the report as per the problem description and illustrate your approach using concepts in Scala Play.

More information on Scala Play

- <https://www.playframework.com/>

The image shows a web browser window with the title "App Name". The address bar displays "https://localhost:9000/add_picture". The page content includes a navigation bar with links: "Explore", "Share a picture", "Profile", and "Logout". The main form area contains a "Select picture" button, a text input field labeled "picture location", a "Description" text area, a "Share" button with a menu icon, and "Submit" and "Cancel" buttons.

Figure 3: Share a new picture.