



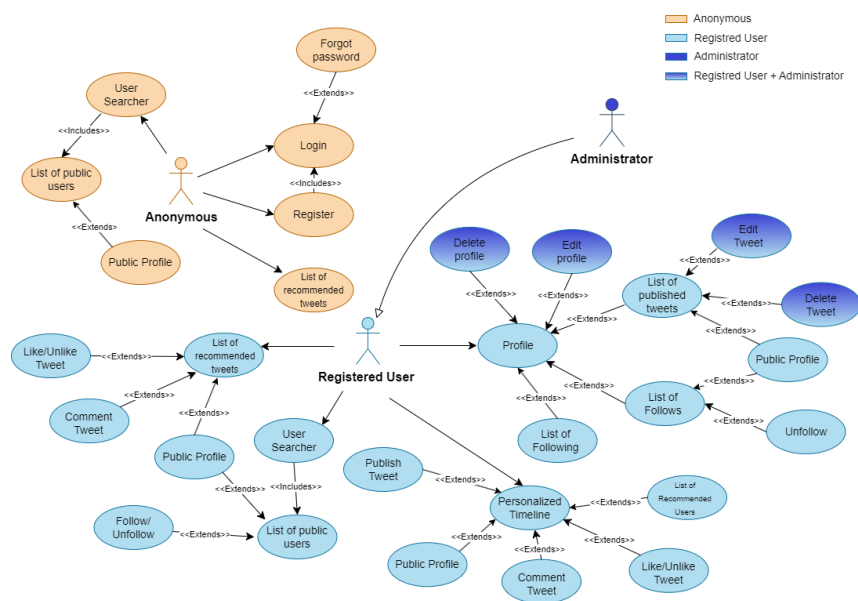
INTRODUCTION

We are going to develop a sports-oriented Twitter called **SLAM**, where we provide users with the latest updates, analysis, and discussions on the world of sports. Our aim is to provide them with a reliable source of information on their favorite teams and athletes, while also offering insights and commentary on major sporting events.

Our goal is to create a dynamic community of sports enthusiasts, where we can engage in lively discussions and share our passion for the games we love. We encourage people to join the conversation, share their thoughts and opinions, and connect with like-minded individuals from around the world.

TYPE OF USERS

Slam will have 3 different types of user (**registered, anonymous and administrator**), each of these will have access to different functionalities depending on their role. These functionalities are described in the following use case diagram.

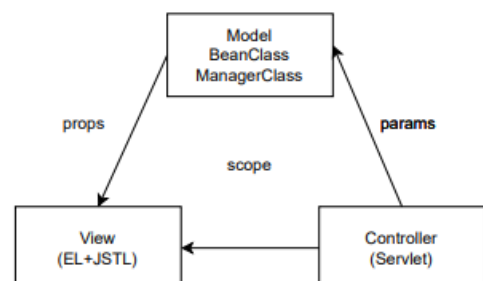


ARCHITECTURAL PATTERN

We used the **MVC (Model-View-Controller)** pattern in order to organize and structure our dynamic website effectively. This architectural design pattern separates the different components of our application, providing clarity and maintainability to our codebase.

At the heart of our MVC diagram lies the Model, which represents the underlying data and business logic of our website. The Model encapsulates the information and functionalities that are essential for the functioning of our application. It is responsible for tasks such as data retrieval, data manipulation, and validation.

The View component handles the presentation layer of our website. It represents the user interface and is responsible for displaying the data from the Model to the users in a visually appealing and intuitive manner. The View ensures that the information is presented in the most appropriate format, be it HTML, CSS, or any other technology required for rendering the user interface.





The Controller acts as the intermediary between the Model and the View. It receives user inputs and initiates the appropriate actions within the Model, which may involve fetching or updating data. The Controller also interacts with the View, updating it with the latest information from the Model or instructing it to display specific views based on user interactions.

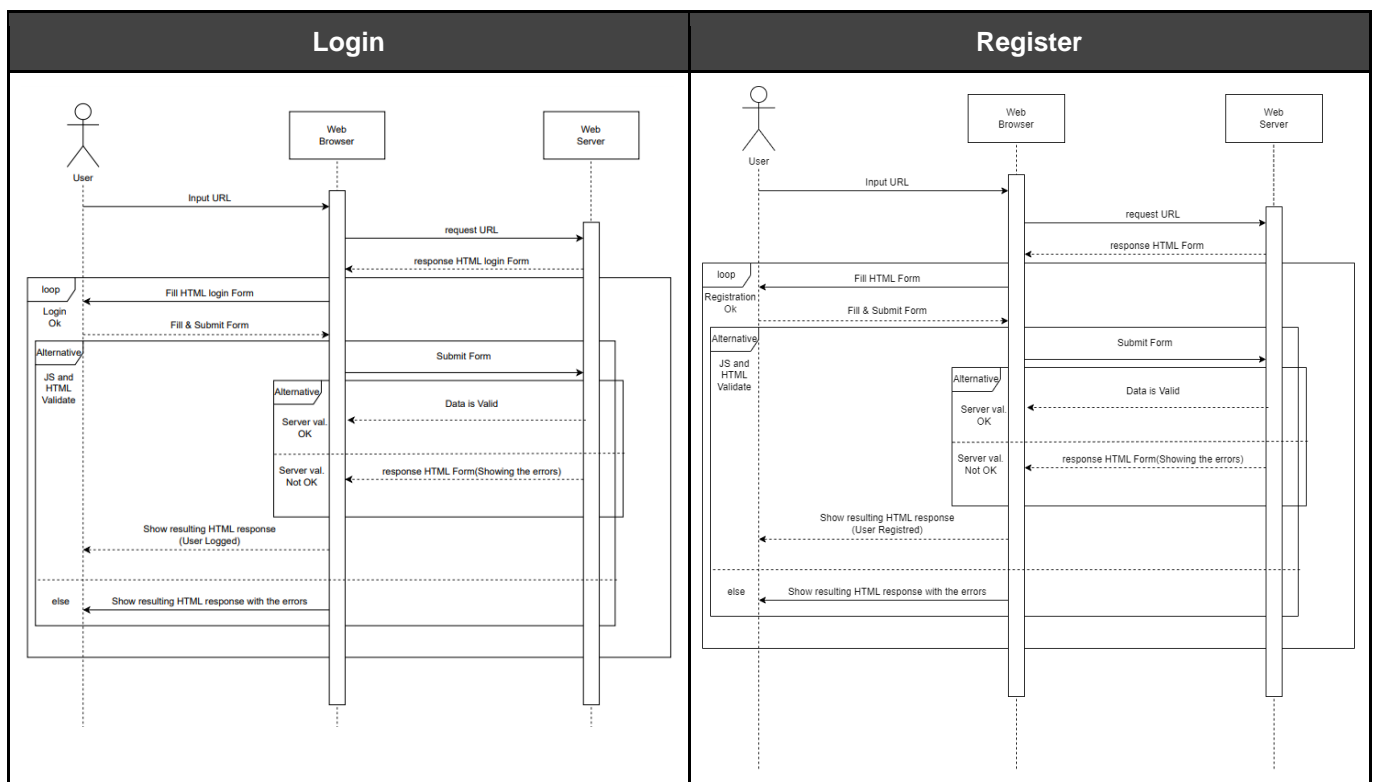
By implementing the MVC pattern, we achieve a clear separation of concerns within our dynamic website. This separation allows us to modify or enhance specific components independently without affecting the entire system. It also promotes code reusability, as different Views can utilize the same Model or multiple Controllers can interact with a single Model.

Overall, the MVC pattern improves the maintainability, scalability, and testability of our dynamic website, making it easier to develop, debug, and enhance over time.

LAYERS AND MVC DIAGRAMS

The user will start from a landing page where 3 options will be displayed: Login, register, or continue anonymously.

If the user chooses to access either of the first two options, it will be redirected to the login or registration form, respectively. As we will see, they have a similar design, consisting of a box containing a set of fields that the user will have to fill in to access their account or register. It is worth noting that if the user wants to access their account but does not remember the password, they will have a field that allows them to change it to a new one. Next, we have the **sequence diagrams** in each of these cases.

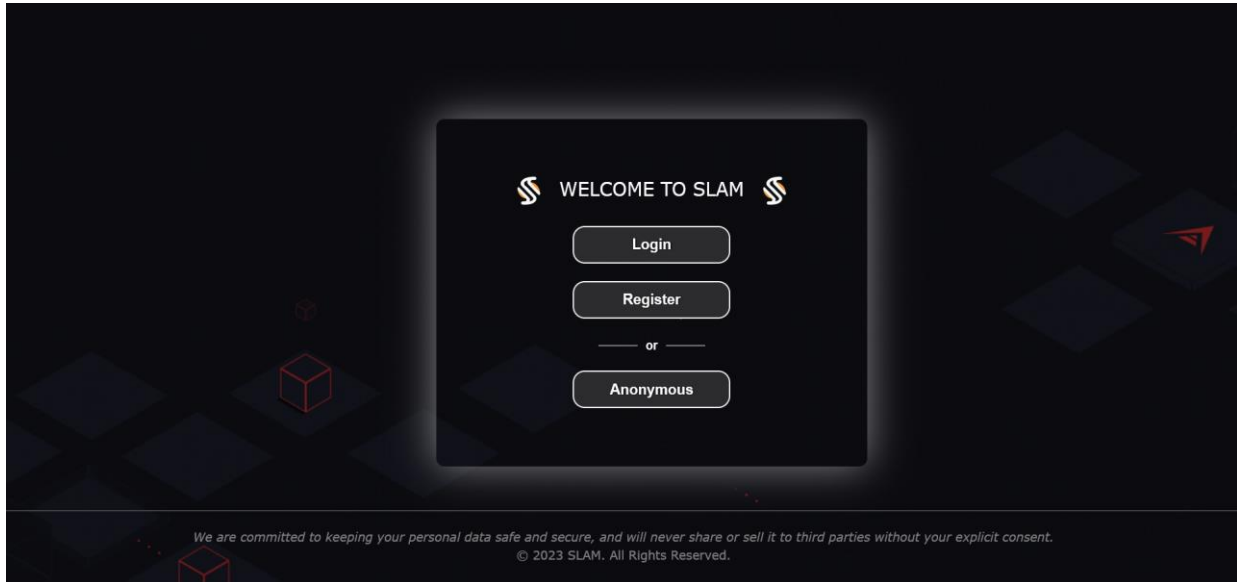


Once the user has logged in, chosen the option to continue anonymously, or is an administrator, it will have a navigation menu located on the left side in a fixed position, with all the functionalities explained in the use case diagram. On the other hand, the content of each functionality will be displayed to the right of the navigation bar, allowing scrolling through the page to view it in its entirety.

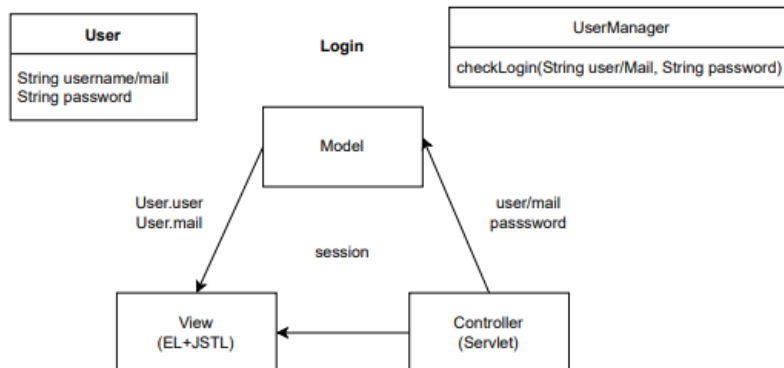


Next, we will see each of the screens (from logged user) together with their MVC diagram.

Landing Page



Login Form





Log in to Slam

Username:
Enter your username

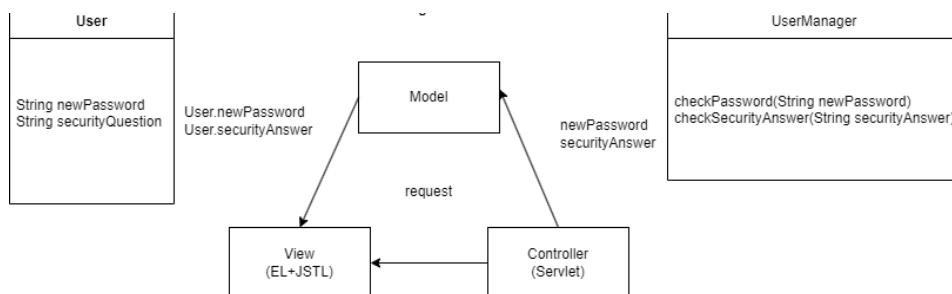
Password:
Enter your password

Submit

You don't have an account yet? [Register](#)

We are committed to keeping your personal data safe and secure, and will never share or sell it to third parties without your explicit consent.
© 2023 SLAM. All Rights Reserved.

Change Password Form



Changing password of rubeenVM

New Password:
..

Password Requirements

Password must contain:
Minimum 8 characters
At least 1 lowercase
At least 1 uppercase
At least 1 number
At least 1 special character

Confirm Password:
Password

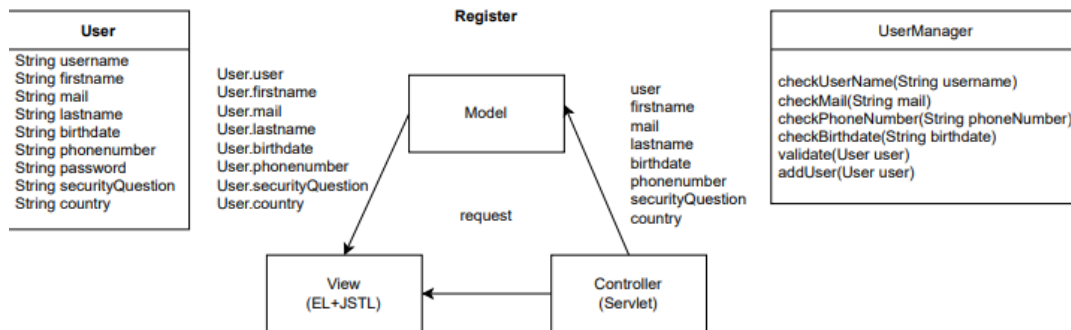
Answer to the security question: Name of your pet
Answer to the question

Change Password

We are committed to keeping your personal data safe and secure, and will never share or sell it to third parties without your explicit consent.
© 2023 SLAM. All Rights Reserved.



Register Form



Register to SLAM

Username*: First Name*:

Mail*: Last Name*:

Password*: Confirm Password*:

Password Requirements

Birth Date*: Phone Number:

Choose your security question*: Select your country:

Answer to the question*: I agree to the terms and conditions ☐

[You already have an account? Login](#)

We are committed to keeping your personal data safe and secure, and will never share or sell it to third parties without your explicit consent.
© 2021 SLAM. All Rights Reserved.

Timeline Usuaris

Home
Explore
Search User
Profile
Logout

SLAM

Publish a New Tweet

Status: Feeling Blue

WHAT'S GOING ON

ruben3 *2023-06-25 19:42:52.0

Vaya partidazo del FCB!!

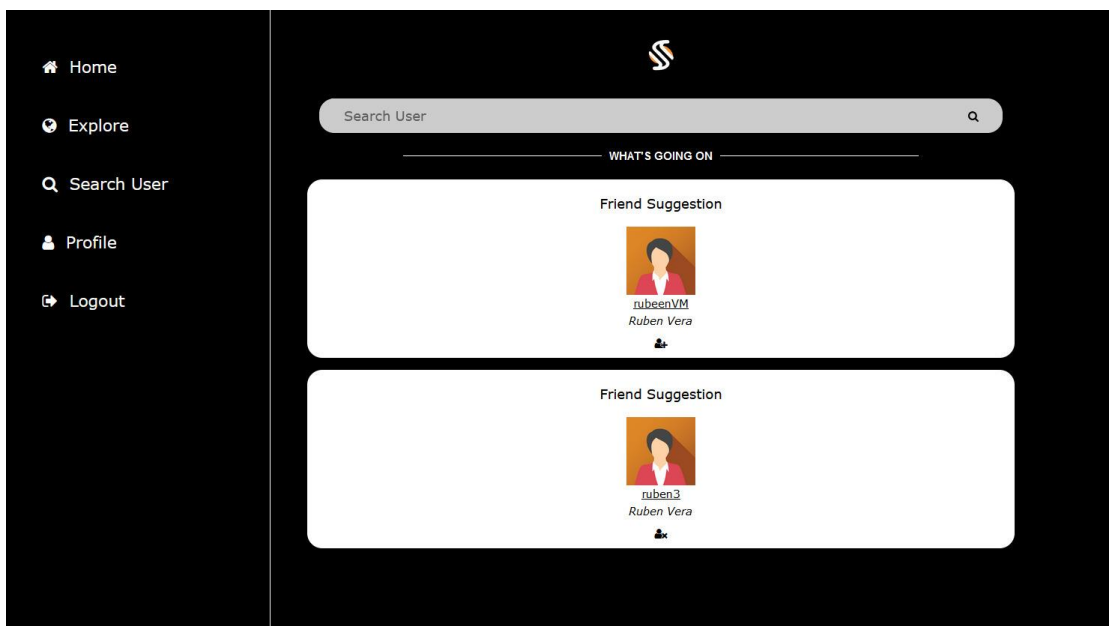
Friend Suggestion

rubenVM

Explore



Search User





Profile

The screenshot displays the user profile for @ruben2, Ruben Vera. The profile shows 0 followers, 1 following, and 1 tweet. The tweet content is "Magia de Antetokumbo !!". The interface includes a sidebar with navigation options: Home, Explore, Search User, Profile, and Logout. The profile card has an "Edit" link. The tweet card has "Like", "Comment", "Delete", and "Edit" buttons.

Profile Information:

- Username: @ruben2
- Full Name: Ruben Vera
- Followers: 0
- Following: 1
- Tweets: 1

Tweet Information:

- Content: Magia de Antetokumbo !!
- Timestamp: "2023-06-25 19:45:31.0"
- Actions: Like, Comment, Delete, Edit

Edit Info Modal:

The modal contains the following fields and options:

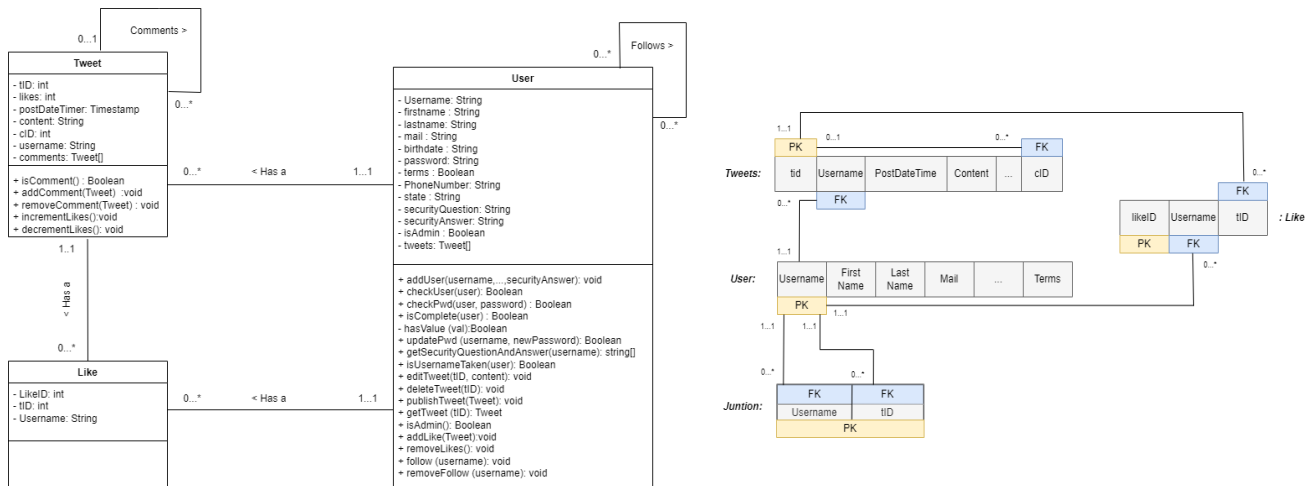
- Username: ruben2
- Email: r@v.com
- First Name: Ruben
- Phone Number: (empty)
- Last Name: Vera
- State: Select your country (dropdown menu)
- Security Question: Answer to the security question: Name of your pet
- Security Answer: security answer
- Submit Button

A red text prompt "Do you want to change password?" is visible below the state dropdown.



DATABASE DESIGN

In order to understand how the design of the database is implemented we designed the following UML and relational database design



FILE STRUCTURE AND DESIGN (CSS)

Our idea is to create a sports-oriented Twitter where the entire interface is minimalist and simple so that the user can enjoy the experience to the fullest. We have also thought that the name should be short and easy to remember, which is why we have opted for a short name, as well as a simple but creative and memorable logo.

Logotype	Color palette	Typography
	<div>#000000</div> <div>#F79D1E</div> <div>#CCCCCC</div> <div>#F3F3F2</div> <div>#FFFFFF</div>	Poppins (Google fonts)

To achieve the desired design, we utilized some designs provided by W3Schools (<https://www.w3schools.com/w3css/>) as a starting point . We then customized them to suit our specific requirements and incorporated our own unique design elements.



The main structure of our JavaServer Pages (JSP) files is as follows:

```
<!-- Page Container -->
<div class="w3-container w3-content w3-round-xlarge" style="max-width:80%;margin-top:3%">
  <!-- The Grid -->
  <div class="w3-row">

    <!-- Left Column -->
    <div class="w3-col m3 left-column">
      <div class="w3-left-align w3-large">
        // Home,Explore ... buttons
      </div>
    </div>
    <!-- End Left Column -->

    <!-- Vertical Line Separator-->
    <div class="verticalLine"></div>
    <!-- End Vertical Line Separator-->

    <!-- Right Column -->
    <div class="w3-col m7">
      // Content code...
    </div>
    <!-- End Right Column -->
  </div>
  <!-- End Grid -->
</div>
<!-- End Page Container -->
```