

**TÀI LIỆU THIẾT KẾ BAN ĐẦU**

**Thiết kế kiến trúc của dự án**

**Được soạn bởi: Nguyễn Thị Lan Hương**

**Nguyễn Minh Dũng**

**Chu Quang Cần**

**Lại Hoàng Hiệp**

**Trần Huy Hoàng**

**Trường Đại học Công nghệ, ĐHQGHN**

**Hà Nội, 26-04-2024**

**Mục lục**

[**Lịch sử sửa đổi 4**](#_gjdgxs)

[**Danh mục hình ảnh và biểu đồ 4**](#_1fob9te)

[**1. Kiến trúc hệ thống 6**](#_3znysh7)

[1.1. Vấn đề đặt ra 6](#_2et92p0)

[1.2. Kiến trúc MVC 7](#_tyjcwt)

[1.3. Biểu đồ thành phần hệ thống 8](#_3dy6vkm)

[**2. Biểu đồ tuần tự các ca sử dụng chính 8**](#_4d34og8)

[2.1. Đăng nhập 9](#_2s8eyo1)

[2.2. Đăng ký 10](#_3rdcrjn)

[2.3. Tạo flashcard 11](#_35nkun2)

[2.4. Đăng bài viết 12](#_44sinio)

[2.5. Tương tác với bài viết 13](#_z337ya)

[2.6. Tạo phòng 14](#_1y810tw)

[2.7. Tham gia phòng 15](#_2xcytpi)

[2.8. Chơi game 16](#_3whwml4)

[**3. Biểu đồ lớp các ca sử dụng 16**](#_qsh70q)

[3.1. Đăng nhập 17](#_3as4poj)

[3.2. Đăng ký 18](#_49x2ik5)

[3.3. Tạo flashcard 19](#_147n2zr)

[3.4. Tạo bài đăng 20](#_23ckvvd)

[3.5. Tương tác với bài đăng 21](#_32hioqz)

[3.6. Tạo phòng 22](#_41mghml)

[3.7. Tham gia phòng 23](#_vx1227)

[3.8. Chơi game 24](#_1v1yuxt)

[**4. Mô phỏng giao diện các chức năng 24**](#_2u6wntf)

[4.1. Đăng ký / Đăng nhập: 24](#_19c6y18)

[4.2. Flashcard: 26](#_nmf14n)

[4.3. Forum: 32](#_sqyw64)

[4.4. Game 35](#_2r0uhxc)

[**5. Thiết kế cơ sở dữ liệu 38**](#_34g0dwd)

[5.1. Giới thiệu 38](#_1jlao46)

[5.2. Thiết kế 38](#_43ky6rz)

[5.3. Tổng kết 42](#_2iq8gzs)

[**6. Giao diện ngoài 43**](#_3hv69ve)

[6.1. Tổng quan về Gemini 43](#_1x0gk37)

[6.2. Các bước thực hiện 43](#_4h042r0)

# Lịch sử sửa đổi

|  |  |  |  |
| --- | --- | --- | --- |
| **Họ và tên** | **Thời gian** | **Lý do sửa đổi** | **Phiên bản** |
| Nguyễn Minh Dũng | 26/04/2024 | Khởi tạo mẫu tài liệu | 1.0 |
| Chu Quang Cần | 29/04/2024 | Bổ sung phụ lục | 1.1 |
| Chu Quang Cần | 29/04/2024 | Viết kiến trúc hệ thống | 1.2 |
| Chu Quang Cần | 30/04/2024 | Viết thiết kế cơ sở dữ liệu | 1.3 |
| Trần Huy Hoàng | 1/5/2024 | Viết thiết kế biểu đồ tuần tự các ca sử dụng chính | 1.4 |
| Trần Huy Hoàng | 2/5/2024 | Viết thiết kế biểu đồ tuần tự các ca sử dụng chính | 1.5 |
| Trần Huy Hoàng | 2/5/2024 | Viết mô phỏng giao diện các chức năng | 1.6 |

# Danh mục hình ảnh và biểu đồ

[1 - Biểu đồ thành phần hệ thống 8](#_1t3h5sf)

[2 - Biểu đồ đăng nhập 9](#_17dp8vu)

[3 - Biểu đồ đăng ký 10](#_lnxbz9)

[4 - Biểu đồ tạo flashcard 11](#_1ksv4uv)

[5 - Biểu đồ đăng bài viết 12](#_2jxsxqh)

[6 - Biểu đồ tương tác bài viết 13](#_3j2qqm3)

[7 - Biểu đồ tạo phòng 14](#_4i7ojhp)

[8 - Biểu đồ tham gia phòng 15](#_1ci93xb)

[9 - Biểu đồ chơi game 16](#_2bn6wsx)

[10 - Biểu đồ lớp - Đăng nhập 17](#_1pxezwc)

[11 - Biểu đồ lớp - Đăng ký 18](#_2p2csry)

[12 - Biểu đồ lớp - Tạo flashcard 19](#_3o7alnk)

[13 - Biểu đồ lớp - Tạo bài đăng 20](#_ihv636)

[14 - Biểu đồ lớp - Tương tác bài đăng 21](#_1hmsyys)

[15 - Biểu đồ lớp - Tạo phòng 22](#_2grqrue)

[16 - Biểu đồ lớp - Tham gia phòng 23](#_3fwokq0)

[17 - Biểu đồ lớp - Chơi game 24](#_4f1mdlm)

[18 - Giao diện đăng ký tài khoản 25](#_3tbugp1)

[19 - Giao diện đăng nhập 25](#_28h4qwu)

[20 - Trang chủ flashcards 26](#_37m2jsg)

[21 - Chọn một chủ đề flashcards 27](#_1mrcu09)

[22 - Giao diện một màn hình học 1 flashcard 27](#_46r0co2)

[23 - Giao diện một màn hình học 1 flashcards 28](#_2lwamvv)

[24 - Thêm flashcards vào một bộ có sẵn 28](#_111kx3o)

[25 - Thêm một flashcard mới vào một bộ có sẵn 29](#_3l18frh)

[26 - Thêm một bộ flashcards mới 29](#_206ipza)

[27 - Thêm một bộ flashcards mới 30](#_4k668n3)

[28 - Xóa một bộ flashcard 30](#_2zbgiuw)

[29 - Xóa một bộ flashcards 31](#_1egqt2p)

[30 - Xóa một flashcard trong một bộ 32](#_3ygebqi)

[31 - Xóa một flashcard trong một bộ 32](#_2dlolyb)

[32 - Diễn đàn 33](#_3cqmetx)

[33 - Tạo bài viết 34](#_1rvwp1q)

[34 - Xem/tương tác với bài viết 35](#_4bvk7pj)

[35 - Trò chơi 36](#_1664s55)

[36 - Tạo phòng 36](#_3q5sasy)

[37 - Phòng 37](#_25b2l0r)

[38 - Tham gia phòng 37](#_kgcv8k)

[39 - Biểu đồ cơ sở dữ liệu 42](#_xvir7l)

# Kiến trúc hệ thống

## Vấn đề đặt ra

Trong quá trình lựa chọn kiến trúc cho hệ thống của mình, chúng tôi đã xác định các vấn đề chính mà chúng tôi cần giải quyết để đảm bảo hiệu suất, bảo trì và mở rộng của hệ thống đó là:

* Vấn đề phân tách trách nhiệm và quản lý logic: Hệ thống của chúng tôi đang phát triển với nhiều tính năng phức tạp và các tính năng lại có phần trách nhiệm và quản lý logic khác nhau. Vấn đề đặt ra là làm thế nào để phân tách rõ ràng các trách nhiệm và quản lý logic của các thành phần khác nhau. Chúng tôi cần một cách để tổ chức mã nguồn sao cho dễ dàng bảo trì, tái sử dụng và kiểm thử.
* Vấn đề tách biệt giao diện và xử lý logic: Giao diện người dùng của chúng tôi cần phản ánh trạng thái dữ liệu hiện tại và phản hồi tương tác người dùng, Tuy nhiên, chúng tôi muốn tách biệt giao diện và xử lý logic để dễ dàng thay đổi giao diện mà không ảnh hưởng đến logic và ngược lại. Điều này giúp chúng tôi phát triển và bảo trì ứng dụng một cách linh hoạt hơn.
* Vấn đề tương tác dữ liệu đồng bộ và bất đồng bộ: Ứng dụng của chúng tôi tương tác với cơ sở dữ liệu để lấy và cập nhật dữ liệu. Việc quản lý tương tác này một cách hiệu quả và đảm bảo tính nhất quán của dữ liệu trở thành một thách thức. Chúng tôi cần một cơ chế để xử lý tương tác dữ liệu đồng bộ và bất đồng bộ, đồng thời đảm bảo tính toàn vẹn và khả năng phục hồi khi có lỗi xảy ra.

Và kiến trúc MVC (Model - View - Controller) mà chúng tôi chọn cho hệ thống của mình đã giải quyết được hết các vấn đề trên.

## Kiến trúc MVC

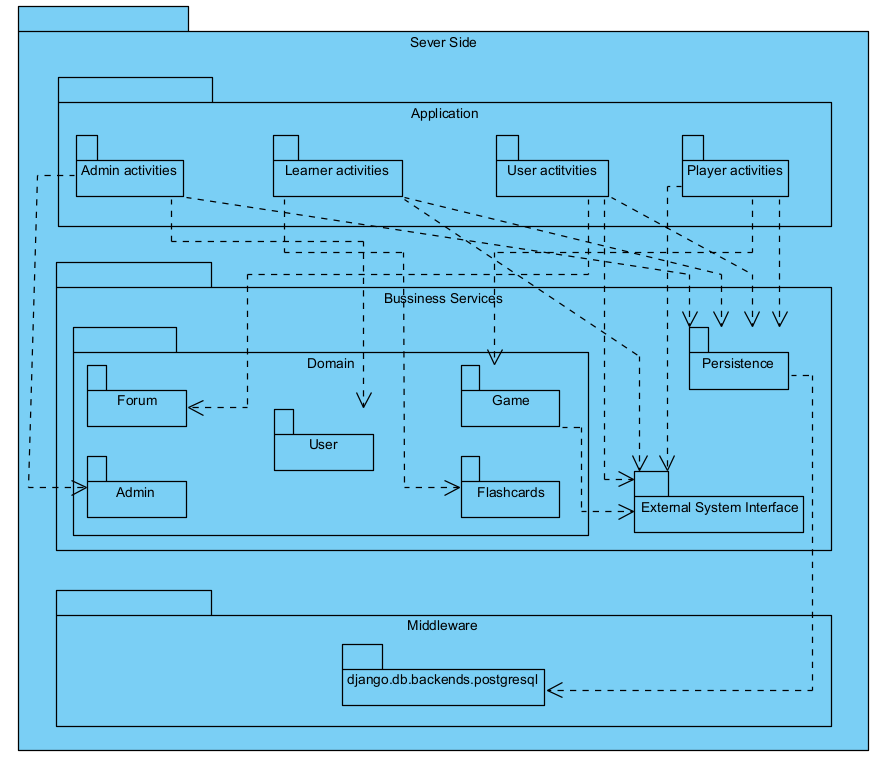
Mô hình kiến trúc MVC được chia làm 3 phần:

* Model (Mô hình): Model đại diện cho việc quản lý và xử lý logic liên quan đến dữ liệu trong hệ thống. Chúng tôi sử dụng PostgreSQL để lưu trữ và thực hiện xử lý dữ liệu. Các dữ liệu của hệ thống chúng tôi bao gồm: thông tin đăng nhập người dùng, kiến thức có trong flashcards, bài đăng và comment trên forum, và kiến thức phục vụ các trò chơi ôn tập khác. Model cung cấp các phương thức và thao tác để truy xuất, thêm, sửa, xóa dữ liệu từ cơ sở dữ liệu.
* View (Giao diện): View là mô-đun phụ trách phần hiển thị giao diện người dùng của hệ thống. Chúng tôi sử dụng công cụ hỗ trợ là Bootstrap để giúp đỡ thiết kế giao diện hiển thị. Giao diện được thiết kế để hiển thị dữ liệu từ Model và cung cấp các tương tác cho người dùng, bao gồm đăng nhập, đăng ký, tạo và chỉnh sửa flashcards, tạo và chỉnh sửa bài viết, tương tác với các người dùng khác trên diễn đàn, tạo và chơi các trò chơi học tập.
* Controller (Bộ điều khiển): Controller là phần xử lý logic và điều khiển sự tương tác giữa mô-đun Model và mô-đun View. Chúng tôi sử dụng thành phần Django Form để thực hiện thu thập yêu cầu người dùng và Django View để xử lý yêu cầu HTTP từ người dùng và tạo phản hồi tương ứng. Controller nhận các yêu cầu từ người dùng thông qua giao diện, tương tác với Model thông qua API để truy xuất và cập nhật dữ liệu, sau đó cập nhật lại giao diện để hiển thị kết quả cho người dùng.

Bằng việc sử dụng kiến trúc MVC với 3 thành phần tách biệt, và phối hợp cùng nhau, từ đó cho phép chúng tôi dễ dàng bảo trì, sửa đổi và mở rộng hệ thống trong tương lai.

## Biểu đồ thành phần hệ thống

Biểu đồ các thành phần hệ thống của chúng tôi:



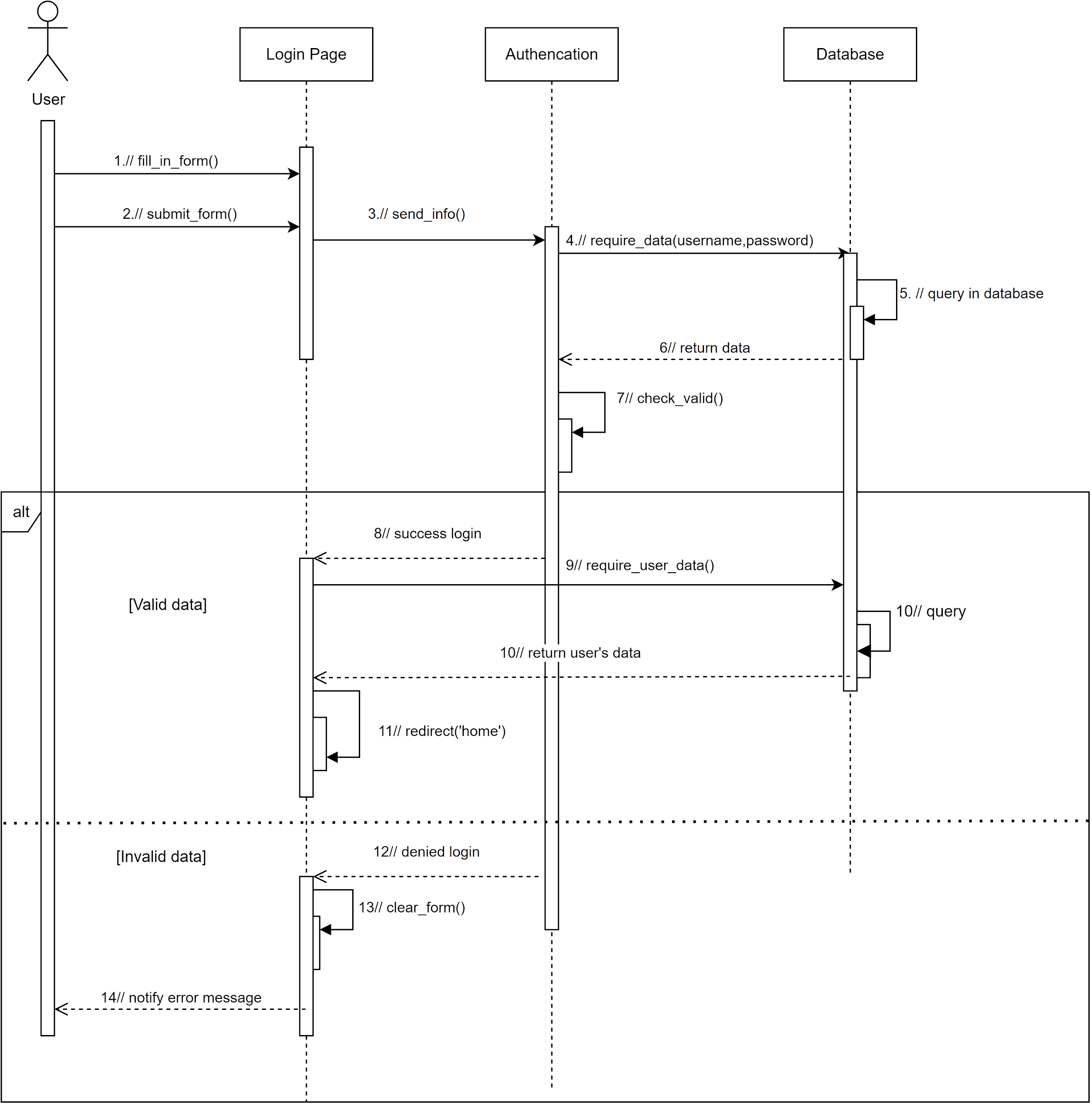
*1 - Biểu đồ thành phần hệ thống*

# Biểu đồ tuần tự các ca sử dụng chính

Trong các biểu đồ tuần tự của chúng tôi có các thành phần sau:

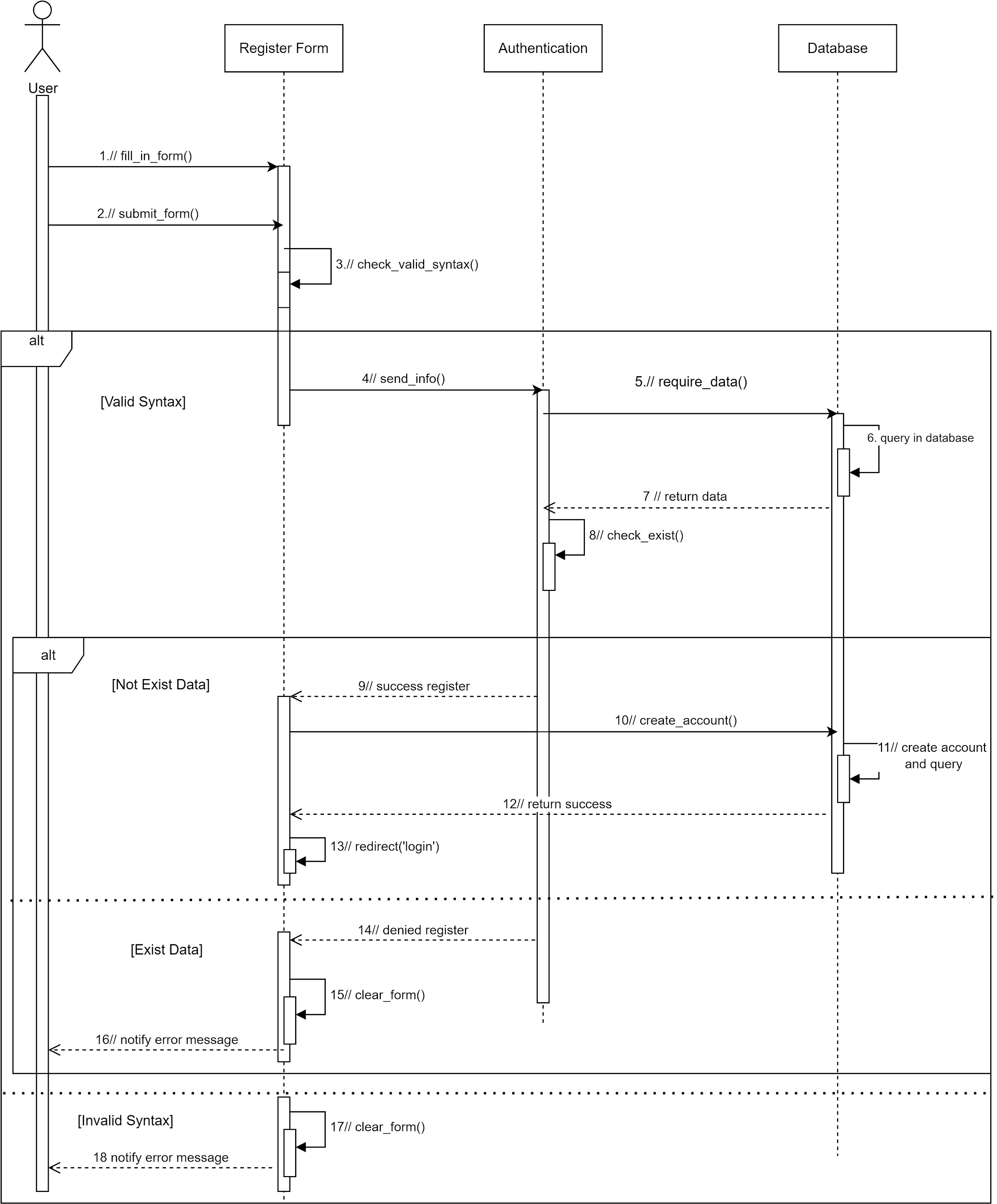
* **Các thành phần giao diện**: Là các thành phần hiển thị cho người dùng
* **Authentication:** Là nơi xử lý logic để thực hiện xác thực đăng nhập, đăng ký cho người dùng.
* **Controller:** Là nơi xử lý logic, yêu cầu người dùng, xử lý dữ liệu khi gửi và nhận từ database.
* **Database:** Là nơi lưu trữ dữ liệu

## Đăng nhập

[](https://app.diagrams.net/?page-id=2YBvvXClWsGukQMizWep&scale=auto#G1TCm0rhRVUfc91OPRSUVipbaBjsEjWSFo)

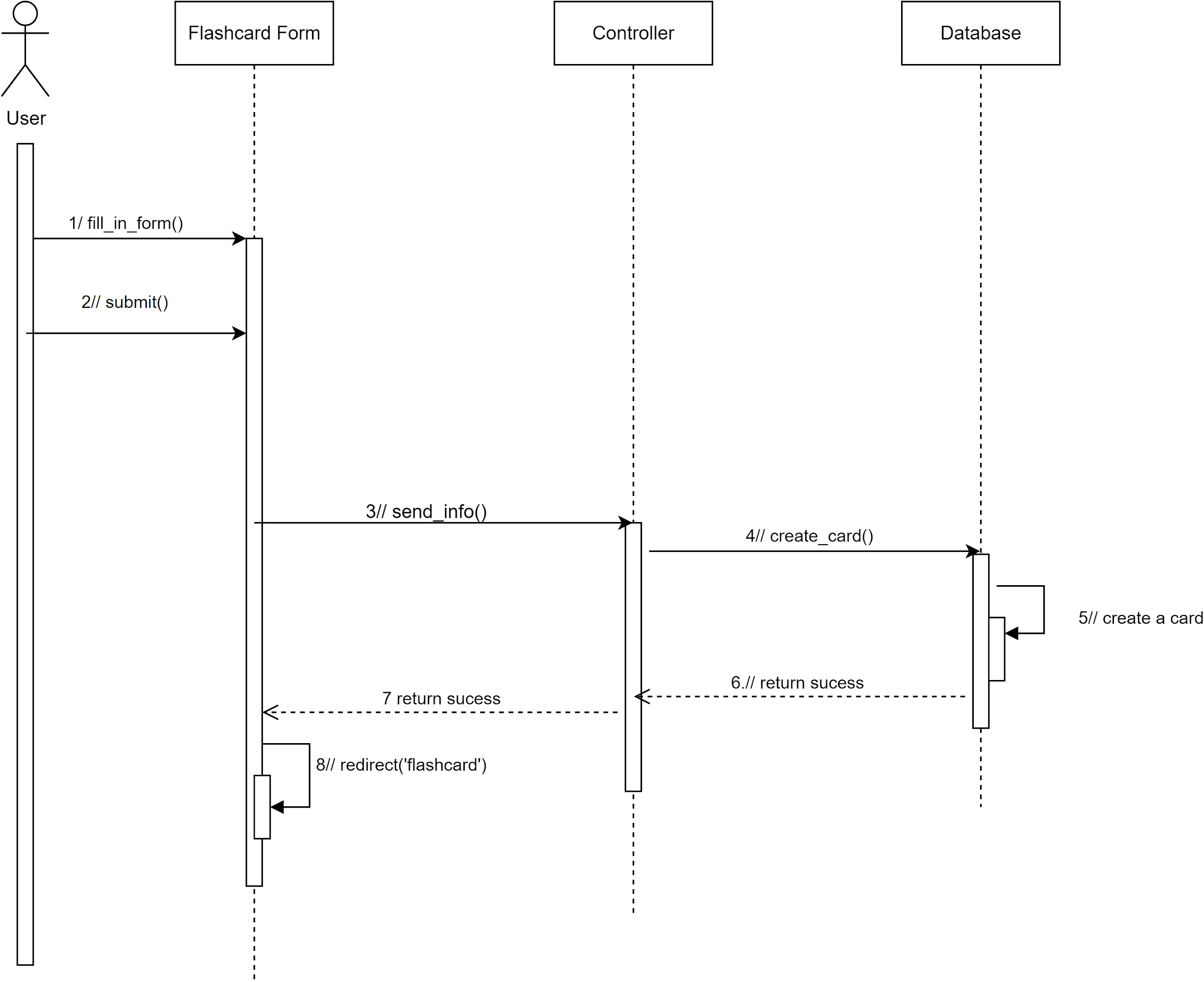
*2 - Biểu đồ đăng nhập*

## Đăng ký

[](https://app.diagrams.net/?page-id=2YBvvXClWsGukQMizWep&scale=auto#G1zWSLrfnumyfs1RI7xWxIEXtKAfohnYgK)

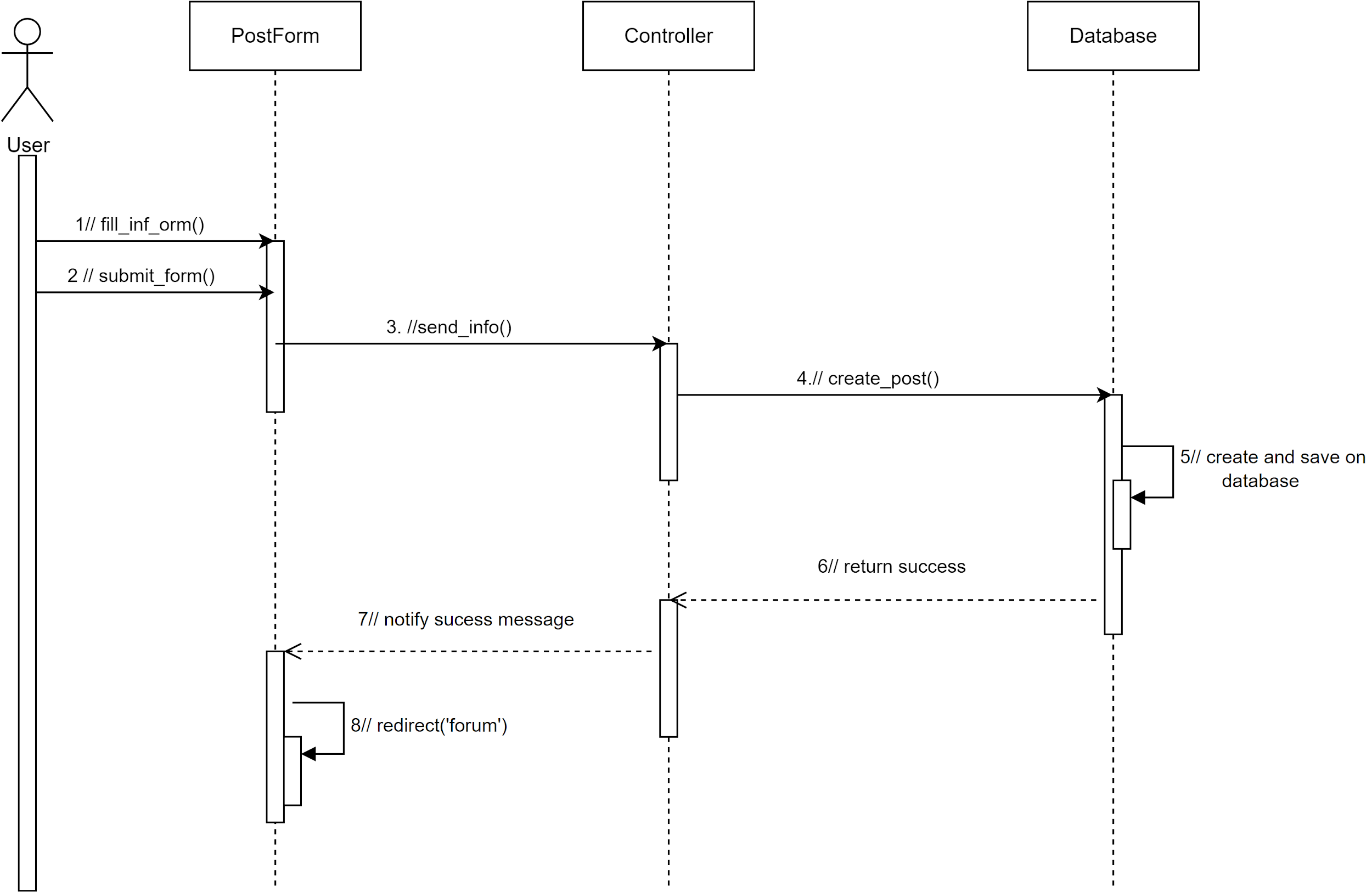
*3 - Biểu đồ đăng ký*

## Tạo flashcard

[](https://app.diagrams.net/?page-id=AeKmbvrvmHNS91Ak4E4n&scale=auto#G1D7pMhZei7I0H2qyo9cPDdqpwh0ljBWZ3)

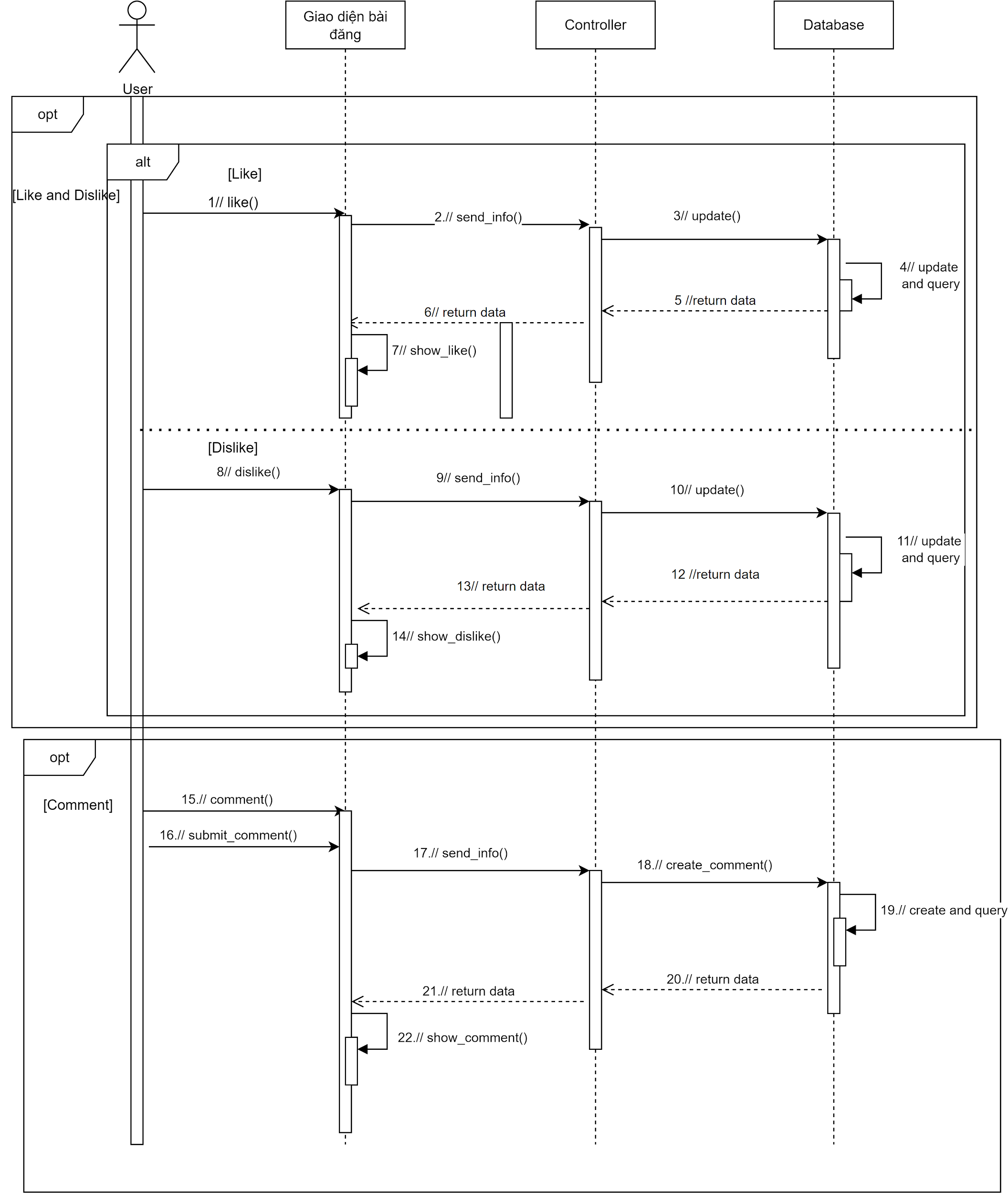
*4 - Biểu đồ tạo flashcard*

## Đăng bài viết

[](https://app.diagrams.net/?page-id=-VWH7j5NQakaeXL_V2_e&scale=auto#G1tpKmlweu2bNPMAQ0a1YbcnJ96aiYEjjc)

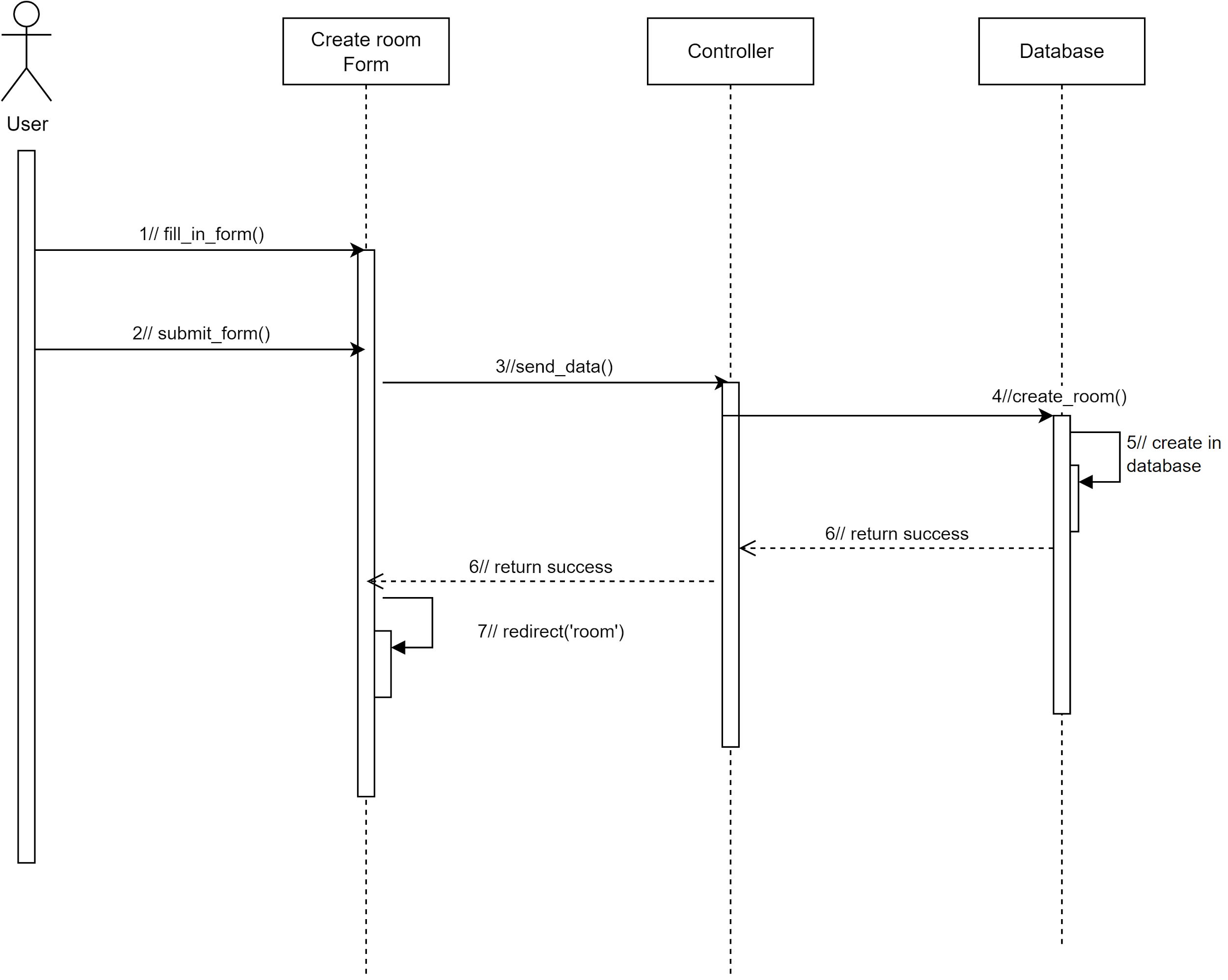
*5 - Biểu đồ đăng bài viết*

## Tương tác với bài viết

[](https://app.diagrams.net/?page-id=tDOMgvErVI7lr2fFGuam&scale=auto#G1nn-S6ZDauk9zsq8dCiv41v-6iCleKwra)

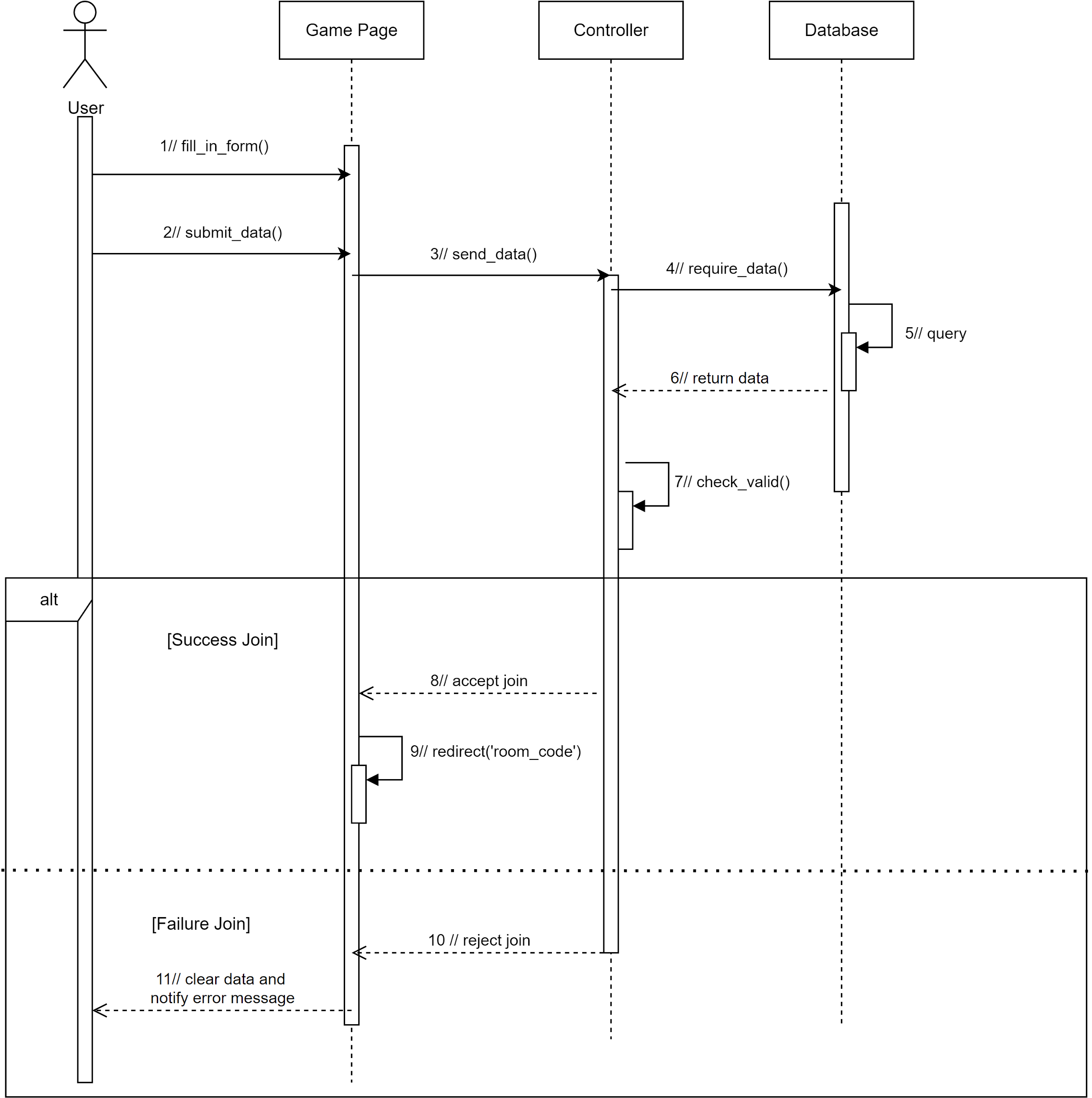
*6 - Biểu đồ tương tác bài viết*

## Tạo phòng

[](https://app.diagrams.net/?page-id=lVZRDcqH54_zMGpRjMKC&scale=auto#G16dtQ2iK7QN6GO57fEcn7hIH5Em5O2NbY)

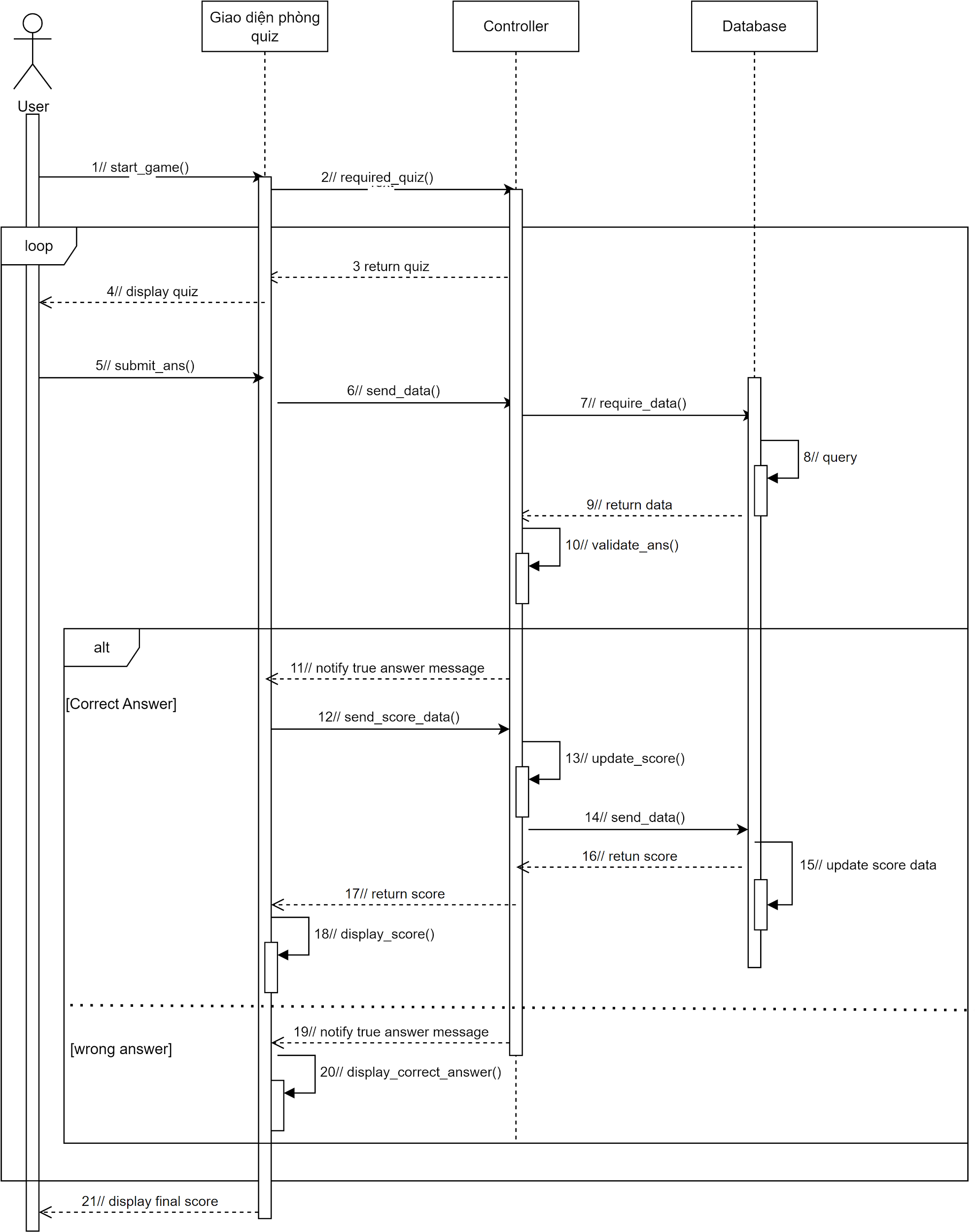
*7 - Biểu đồ tạo phòng*

## Tham gia phòng

[](https://app.diagrams.net/?page-id=Dl38NBbLyUkR68YZPBdE&scale=auto#G1OMmOjYDUOJqIEZqWkNzdFhMOB7Bl8__g)

*8 - Biểu đồ tham gia phòng*

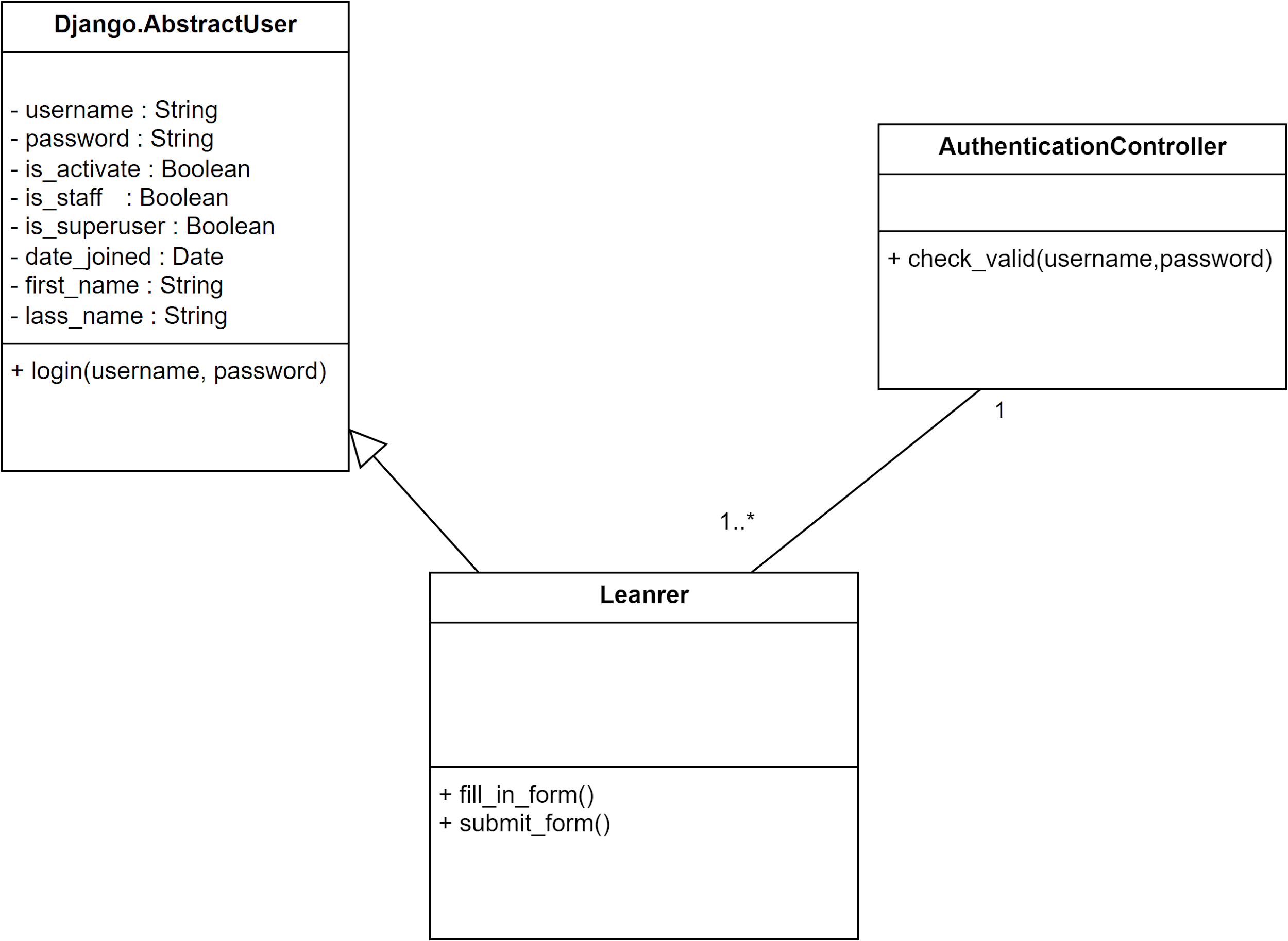
## Chơi game

[](https://app.diagrams.net/?page-id=ukUQNTWw9JEm3HLh-mbu&scale=auto#G1V8iBLvZC_ZAA3to_7IYbhlRZqLGwht4R)

*9 - Biểu đồ chơi game*

# Biểu đồ lớp các ca sử dụng

## Đăng nhập

[](https://app.diagrams.net/?page-id=C5RBs43oDa-KdzZeNtuy&scale=auto#G1v8OCFf2sRWixfFdvan4SvkQ1Uw5laEOi)

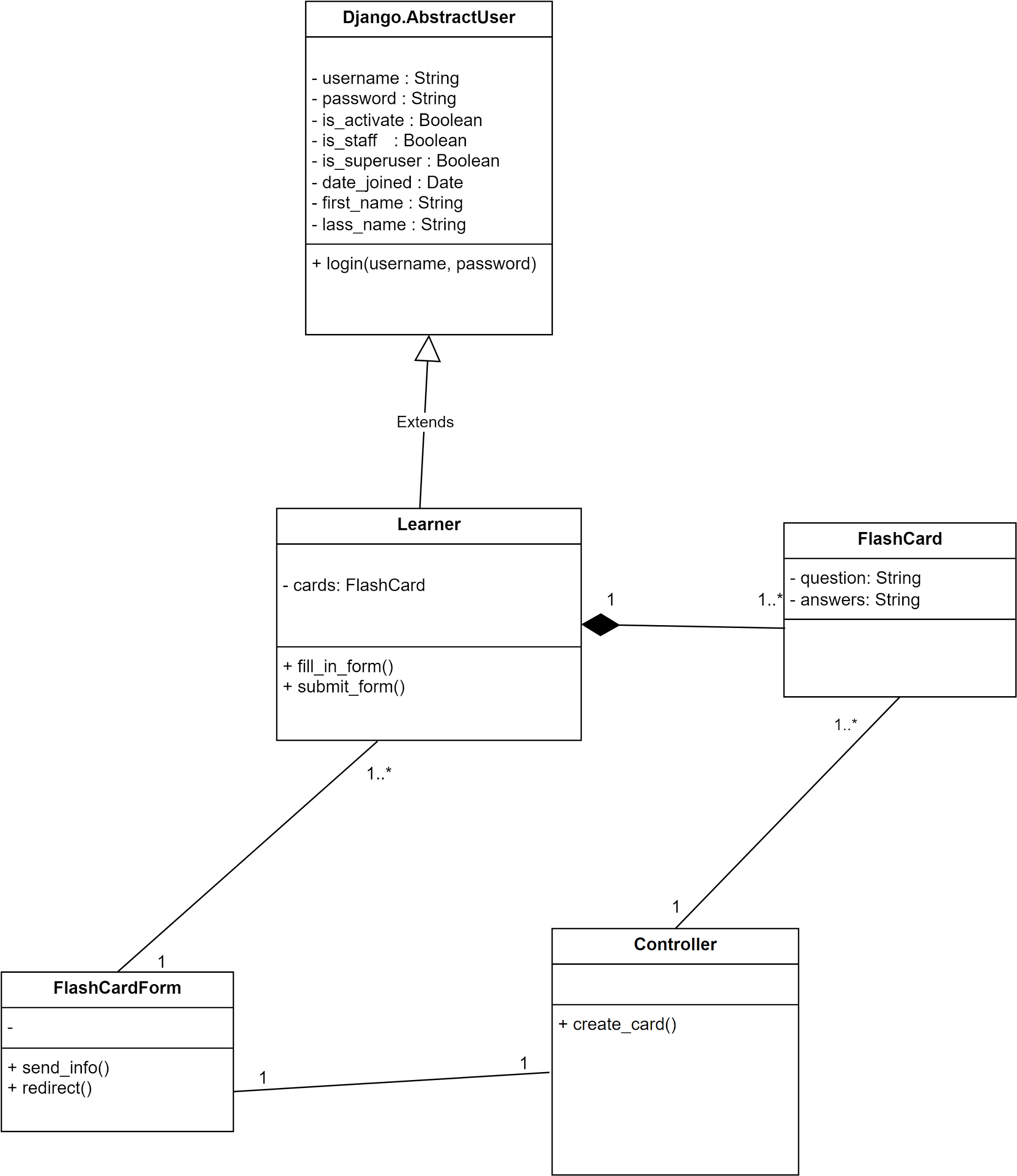
*10 - Biểu đồ lớp - Đăng nhập*

## Đăng ký

[](https://app.diagrams.net/?page-id=C5RBs43oDa-KdzZeNtuy&scale=auto#G1kXfgwfX3ePszs7bEvmluqzXHp6s3QgSL)

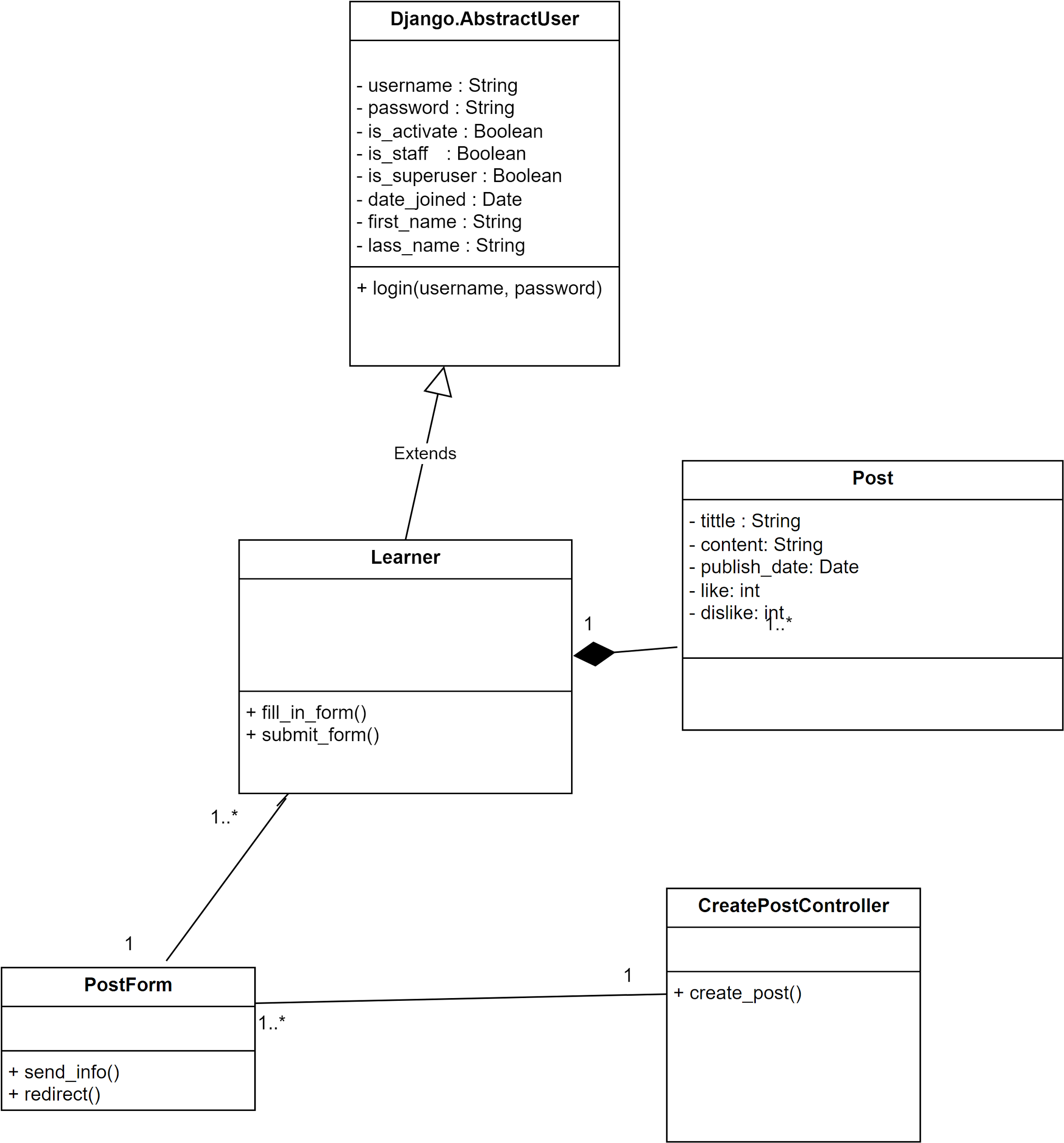
*11 - Biểu đồ lớp - Đăng ký*

## Tạo flashcard

[](https://app.diagrams.net/?page-id=aQLyZZCIkywfewllhT78&scale=auto#G1W_Isu8Pm8jzd6OTPEqYB9LVOuhYEf-a4)

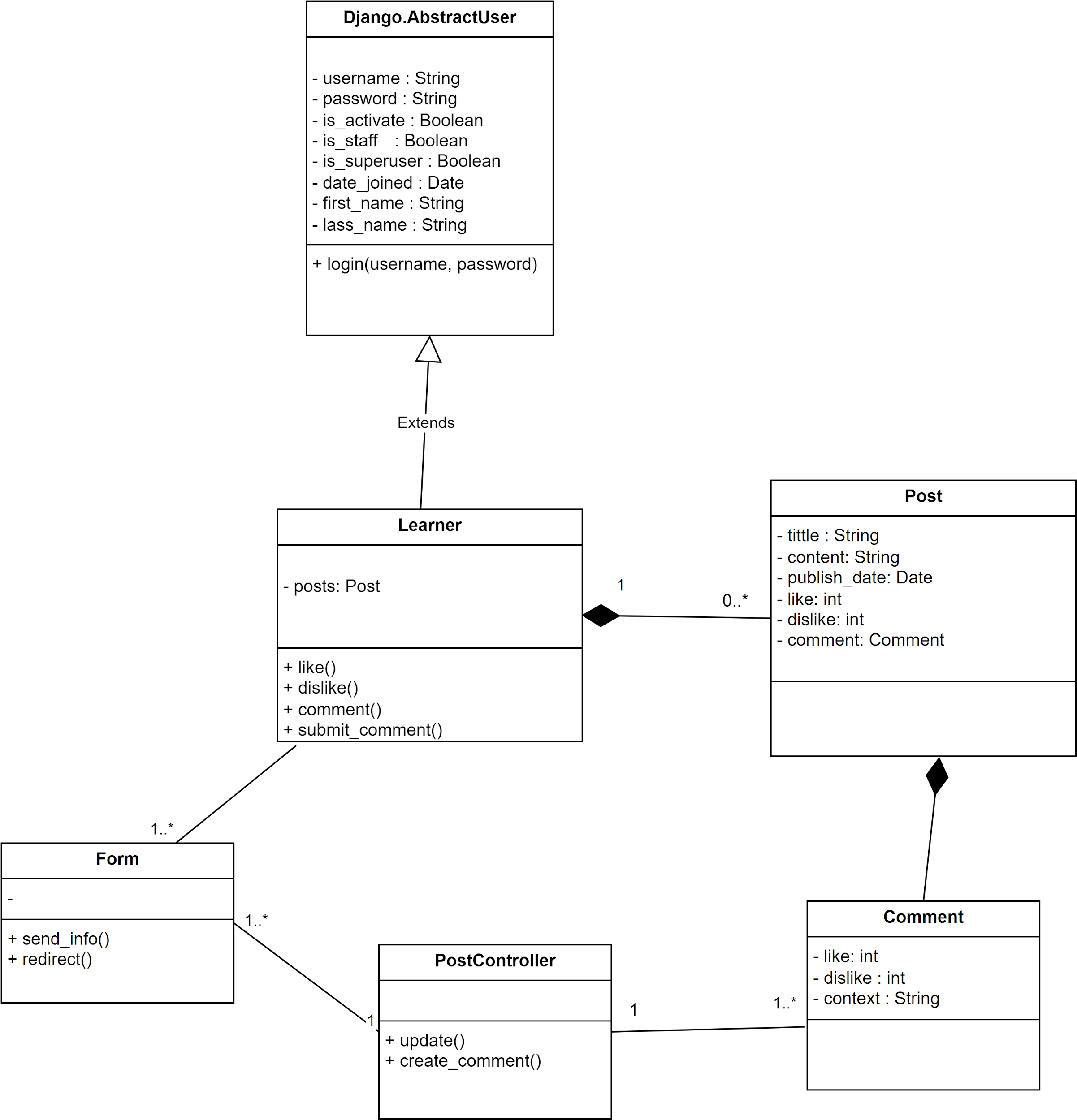
*12 - Biểu đồ lớp - Tạo flashcard*

## Tạo bài đăng

[](https://app.diagrams.net/?page-id=PpWoSl5-L8w25ohJg_TL&scale=auto#G1hj6MAH1kXHtofVVJNLG65JJixHERExql)

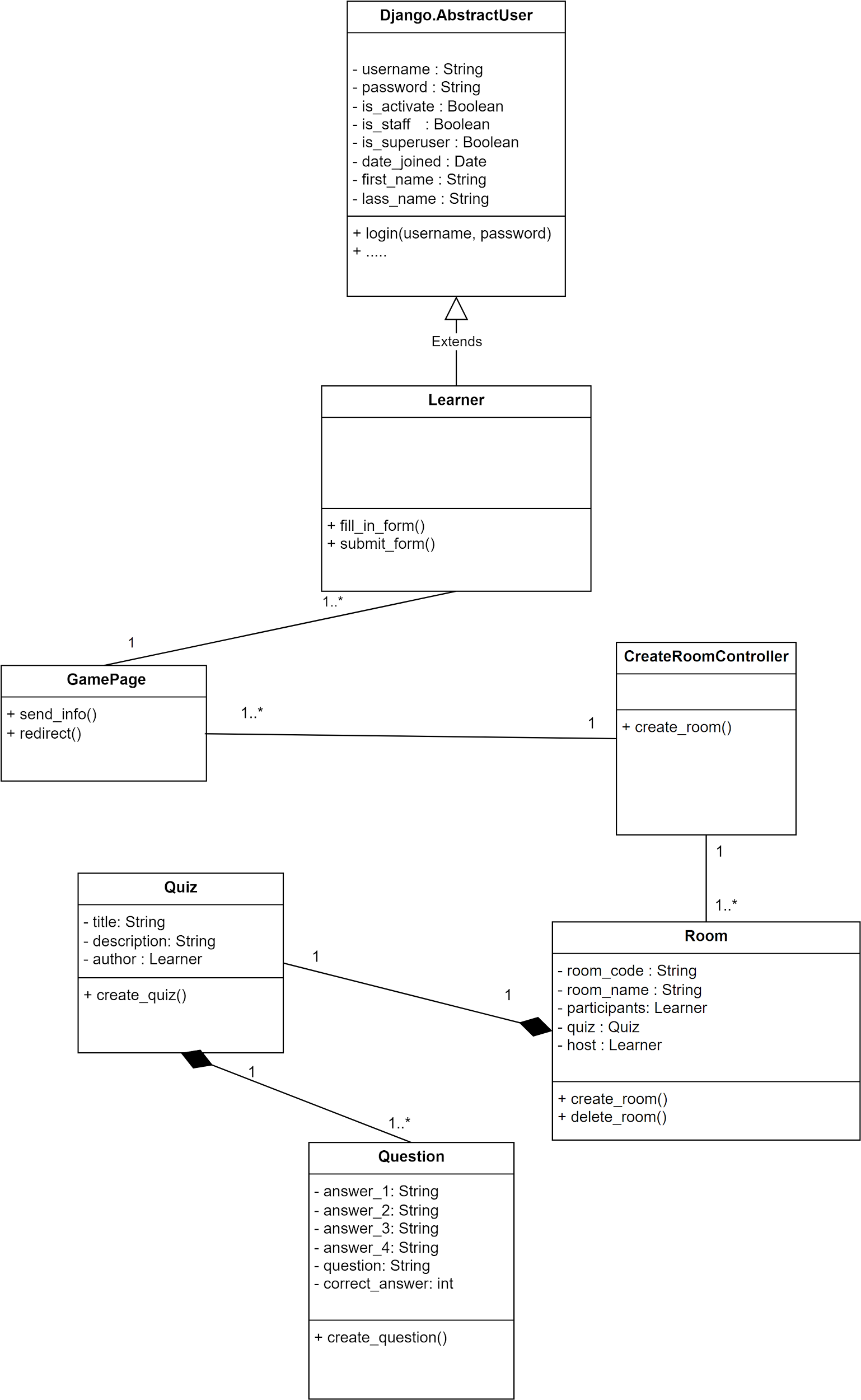
*13 - Biểu đồ lớp - Tạo bài đăng*

## Tương tác với bài đăng

[](https://app.diagrams.net/?page-id=AETPzHvgAv8_pPrgTktr&scale=auto#G1qrDDtTld3pQxIXWBZRY7O68BJEeqnX4a)

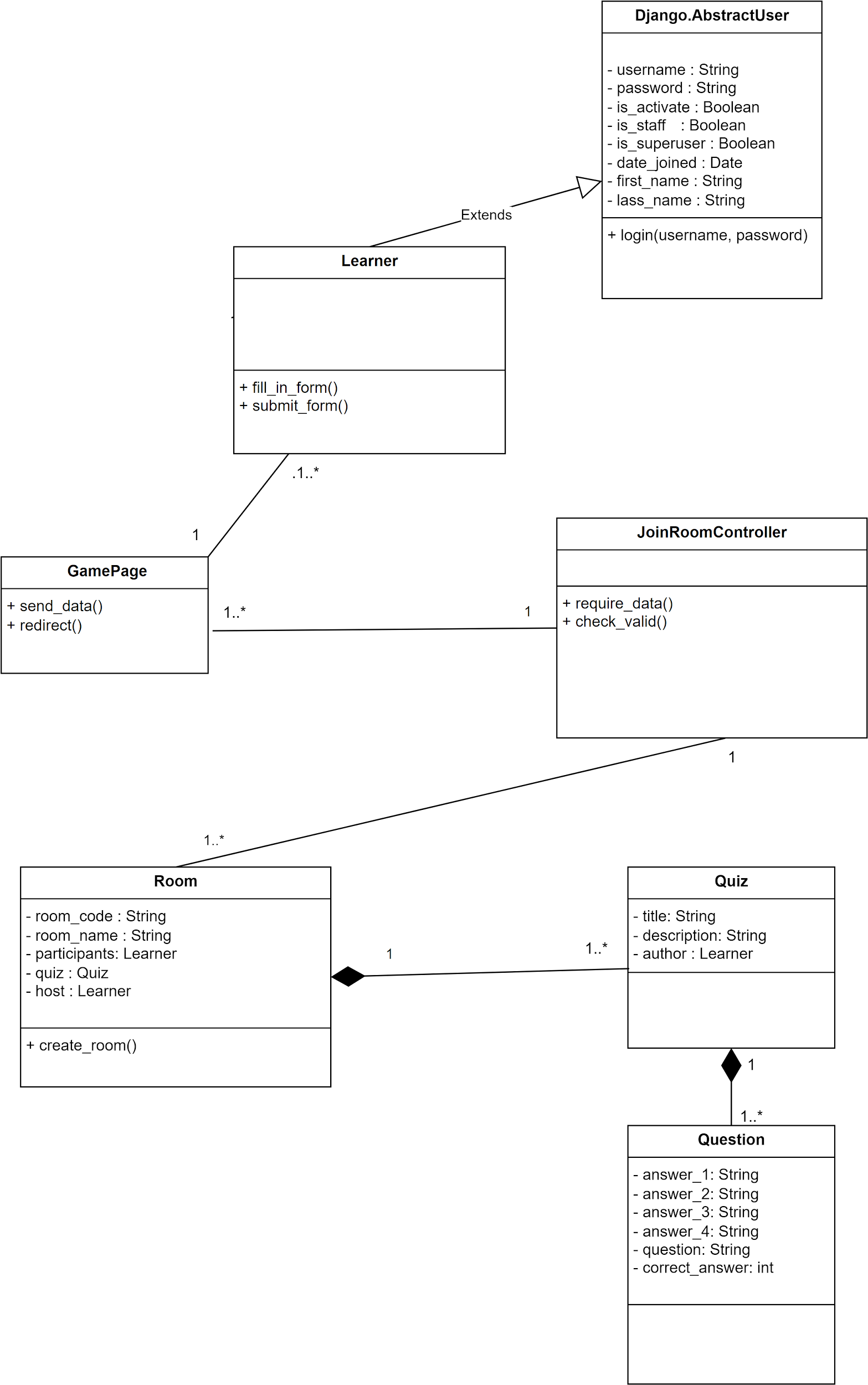
*14 - Biểu đồ lớp - Tương tác bài đăng*

## Tạo phòng

[](https://app.diagrams.net/?page-id=9HSmnZOw_HodIRpLSpn4&scale=auto#G17QJAcVm29mhbASgqo3QbIWHMzrqYfyv7)

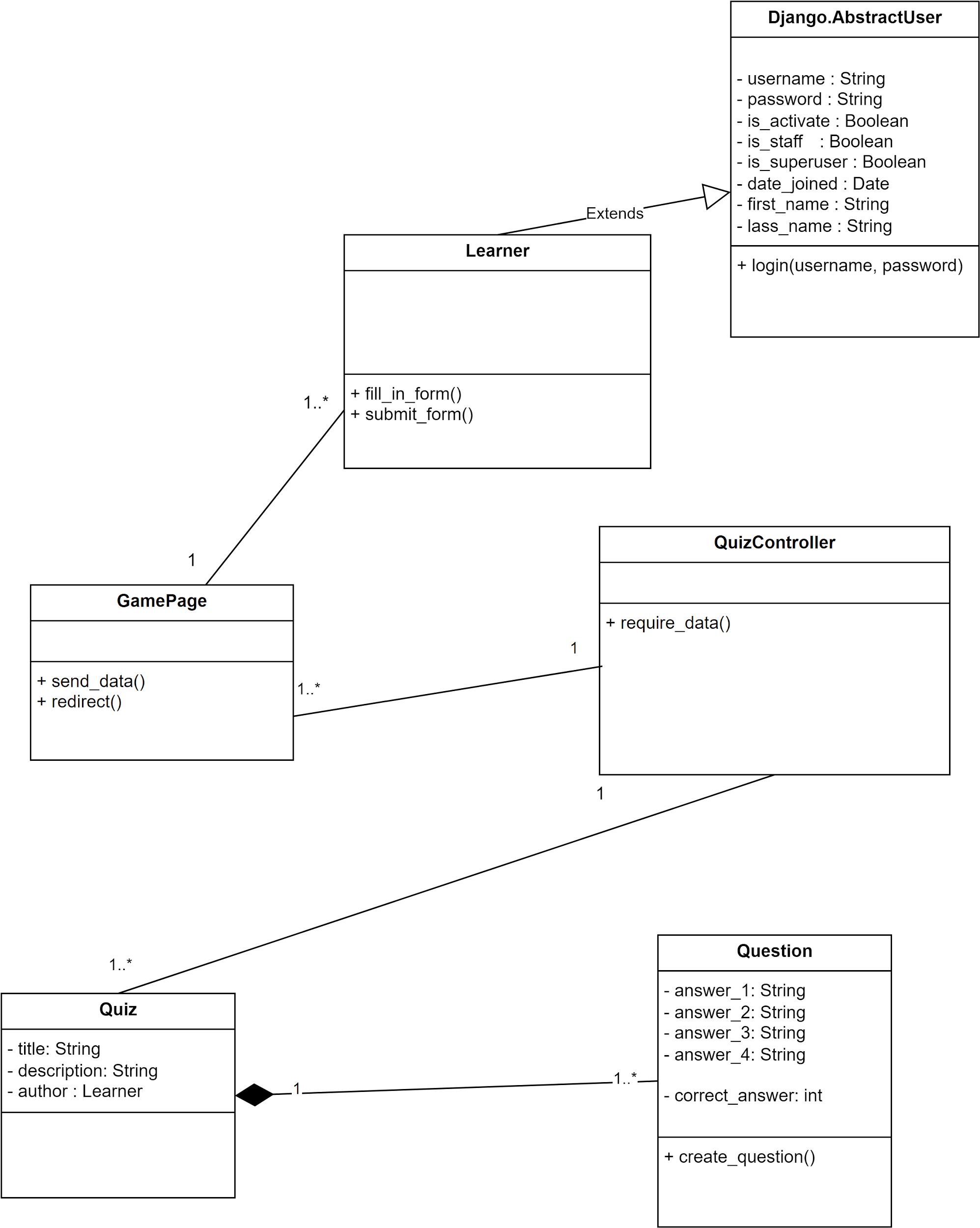
*15 - Biểu đồ lớp - Tạo phòng*

## Tham gia phòng

[](https://app.diagrams.net/?page-id=vQ3H0H758EUg6TH6Jkc9&scale=auto#G1e1TDcJuSWOVDDa8QbJDHsZc_SJSmpmof)

*16 - Biểu đồ lớp - Tham gia phòng*

## Chơi game

[](https://app.diagrams.net/?page-id=PWbRx_ZOfiUIbS1-I8fl&scale=auto#G1-dWiZmz_GLiM3iTHNbWgpo66YhZMNEFf)

*17 - Biểu đồ lớp - Chơi game*

# Mô phỏng giao diện các chức năng

## Đăng ký / Đăng nhập:

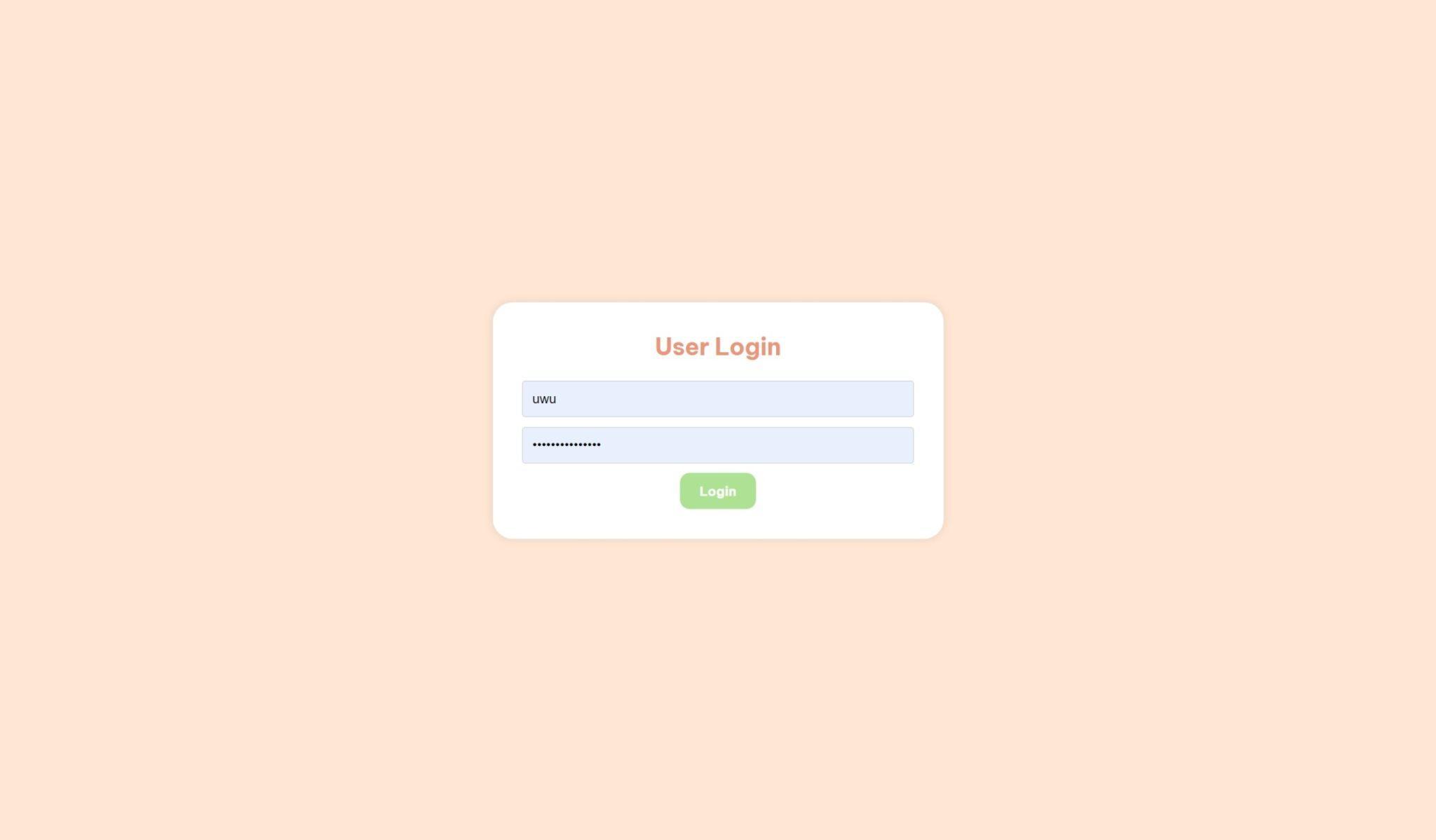
Để có thể sử dụng toàn bộ các chức năng chính của Web, người dùng cần phải có tài khoản để đăng nhập, hoặc đăng ký (nếu chưa có tài khoản)

* **Đối với đăng ký**: Người dùng sẽ cần điền đầy đủ thông tin vào 3 trường: Username (tên để hiển thị trong suốt quá trình dùng), Email (địa chỉ email để lấy lại mật khẩu nếu quên), Password.

Giao điện đăng ký tài khoản


*18 - Giao diện đăng ký tài khoản*

* **Đối với đăng nhập**: Sau khi đã tạo tài khoản thành công hoặc đã có tài khoản trước đó, người dùng cần đăng nhập để sử dụng mọi tính năng của Web. Người dùng cần điền username và mật khẩu của tài khoản đó.



*19 - Giao diện đăng nhập*

## Flashcard:

Sau khi đăng nhập thành công, trang chủ của Web sẽ hiển thị với nội dung được mô tả như bên dưới cho người dùng. Người dùng sau đó có thể chọn thêm/xóa Flashcard hoặc chọn một môn học để học

A screenshot of a computer

Description automatically generated

*20 - Trang chủ flashcards*

* **Chọn môn học để học:** 
  + Nếu người dùng click vào một trong những môn học được hiển thị trên Web, giao diện như bên dưới sẽ hiện ra bao gồm các bộ/gói Flashcard của môn học đó. Ví dụ như click vào Math thì sẽ có giao diện như sau

A screenshot of a computer

Description automatically generated

*21 - Chọn một chủ đề flashcards*

* Sau khi click chọn một bộ/gói Flashcard, người dùng sẽ có thể bắt đầu học và thao tác cùng với Flashcard như bên dưới

A screenshot of a computer

Description automatically generated

*22 - Giao diện một màn hình học 1 flashcard*

A screenshot of a computer

Description automatically generated

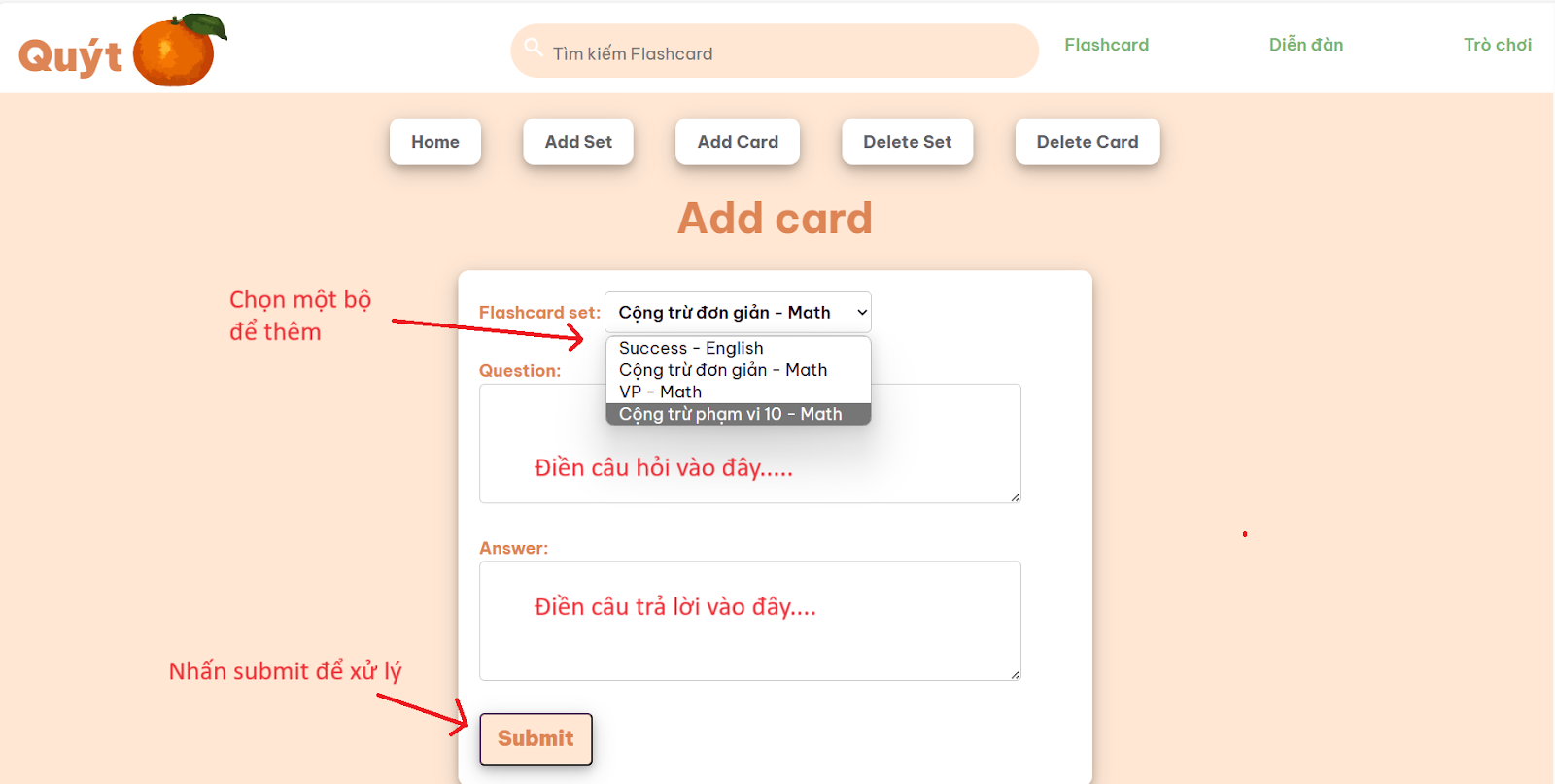
*23 - Giao diện một màn hình học 1 flashcards*

* **Thêm một Flashcard vào một bộ có sẵn:**
  + Người dùng click vào nút “Add Card” (luôn được hiển thị ở bên trên trong các nút chức năng) và một giao diện để thêm một Flashcard sẽ hiển thị như sau:

A screenshot of a computer

Description automatically generated

*24 - Thêm flashcards vào một bộ có sẵn*



*25 - Thêm một flashcard mới vào một bộ có sẵn*

* **Thêm một bộ Flashcard:**
  + Người dùng click vào “Add Set”. Sau khi thêm thành công một bộ thì sẽ nhận được thông báo và chuyển đến trang để thêm từng Flashcard vào bộ đó

A screenshot of a computer

Description automatically generated

*26 - Thêm một bộ flashcards mới*

A screenshot of a card

Description automatically generated

*27 - Thêm một bộ flashcards mới*

* **Xóa một bộ Flashcard:**
  + Người dùng click vào “Delete Set” để thực hiện xóa một bộ Flashcard do mình sở hữu (chỉ có thể xóa bộ Flashcard do mình tạo). Sau đó, người dùng sẽ tiến hành chọn bộ Flashcard muốn xóa và nhấn Delete.

A screenshot of a computer

Description automatically generated

*28 - Xóa một bộ flashcard*

A screenshot of a flashcard set

Description automatically generated

*29 - Xóa một bộ flashcards*

* **Xóa một Flashcard:**
  + Người dùng click vào “Delete Card” để thực hiện xóa một bộ Flashcard do mình sở hữu (chỉ có thể xóa bộ Flashcard do mình tạo). Người dùng sẽ chọn bộ Flashcard của mình chứa Flashcard cần xóa đó, và chọn câu hỏi và câu trả lời để đảm bảo rằng người dùng thực sự muốn xóa Flashcard đó (giao diện cũng giống như thêm một thẻ).

A screenshot of a computer

Description automatically generated

*30 - Xóa một flashcard trong một bộ*

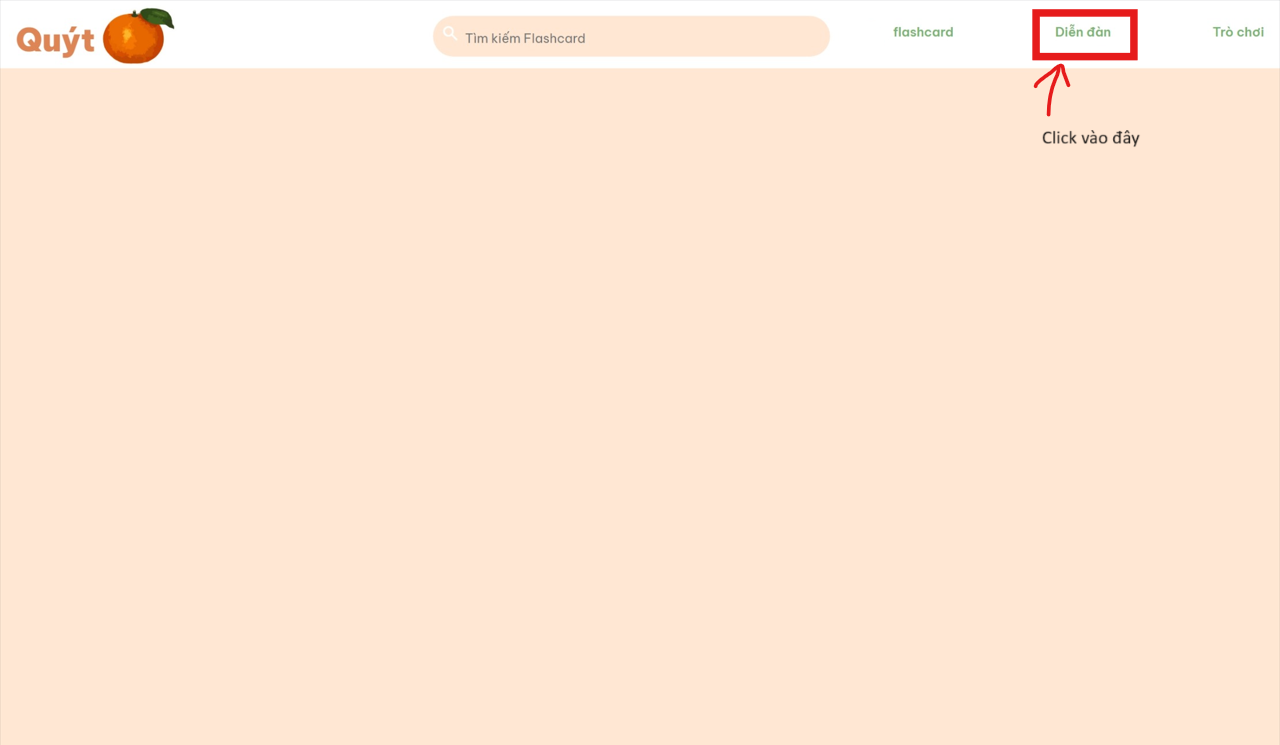
A screenshot of a delete card

Description automatically generated

*31 - Xóa một flashcard trong một bộ*

## Forum:

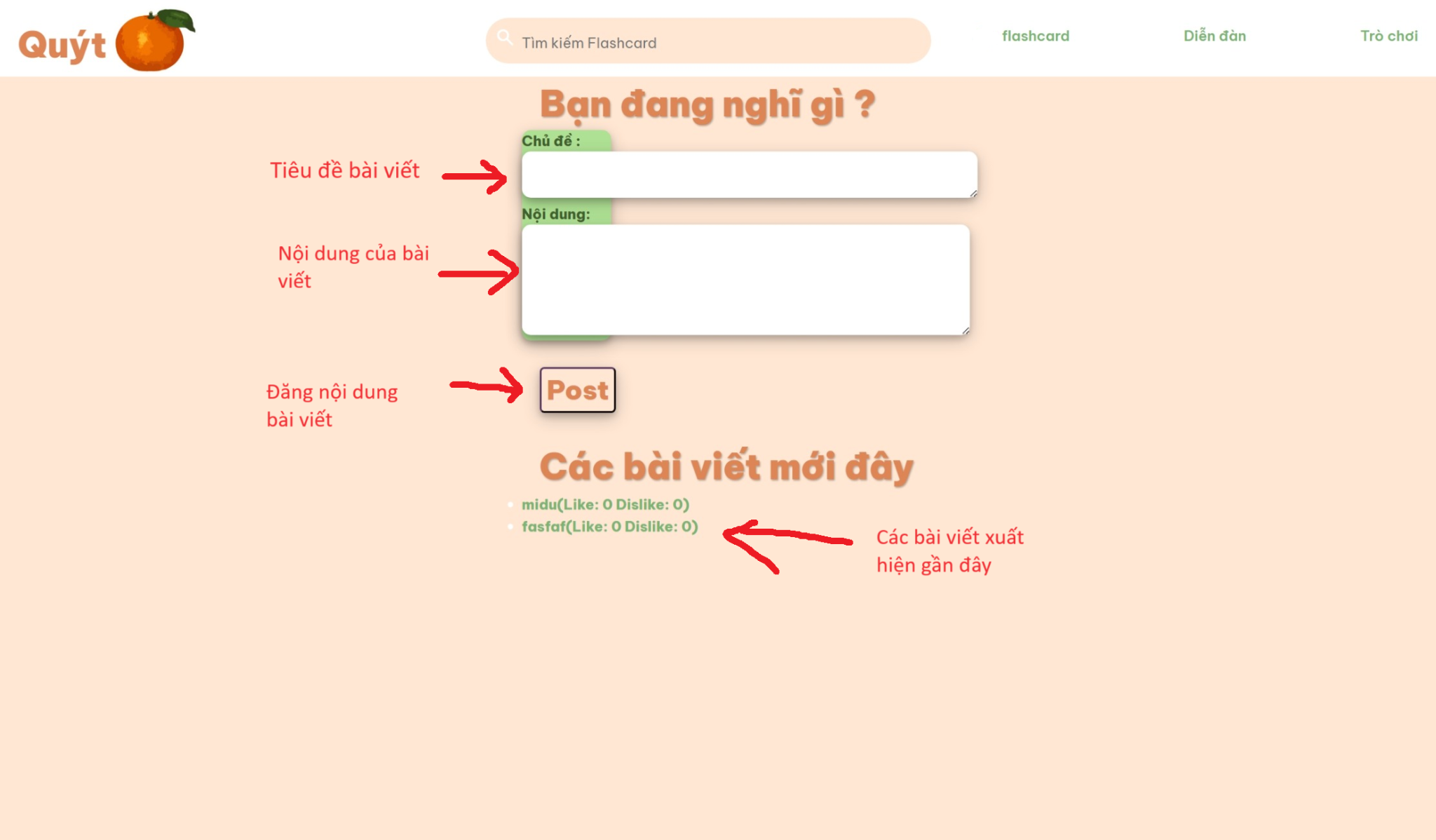
Để sử dụng Forum, người dùng sẽ click vào phần “Diễn đàn” trên thanh điều hướng. Sau đó, Web sẽ đưa người dùng đến giao diện của diễn đàn



*32 - Diễn đàn*

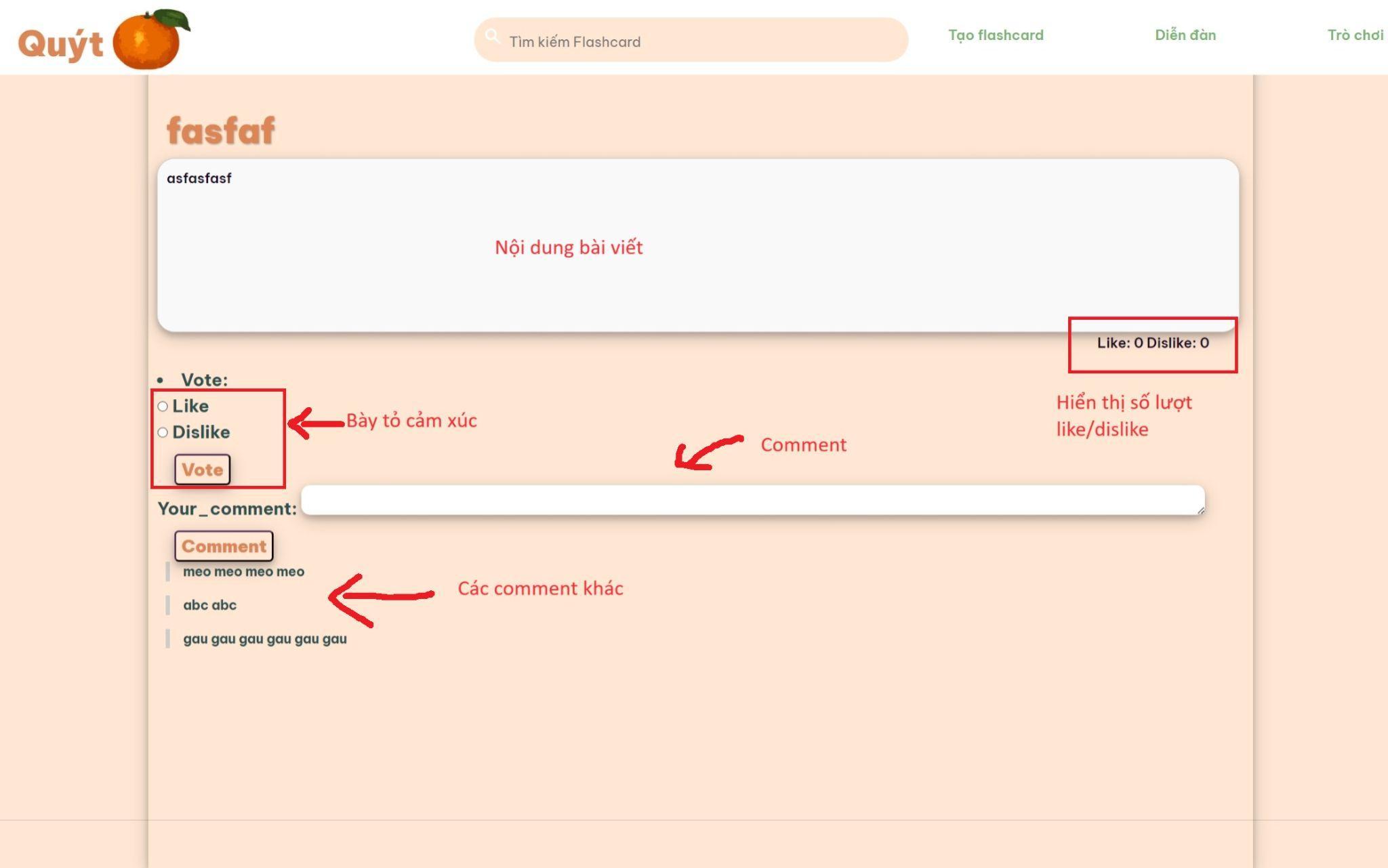
Sau khi click vào “Diễn đàn” trên thanh điều hướng, giao diện mới sẽ hiện ra, cho phép người dùng tạo thêm bài viết mới hoặc xem những bài viết của người khác.

* Trong trường hợp muốn tạo thêm bài viết mới, người dùng điền vào “Tiêu đề” và “Nội dung”, sau đó nhấn “Post” để đăng



*33 - Tạo bài viết*

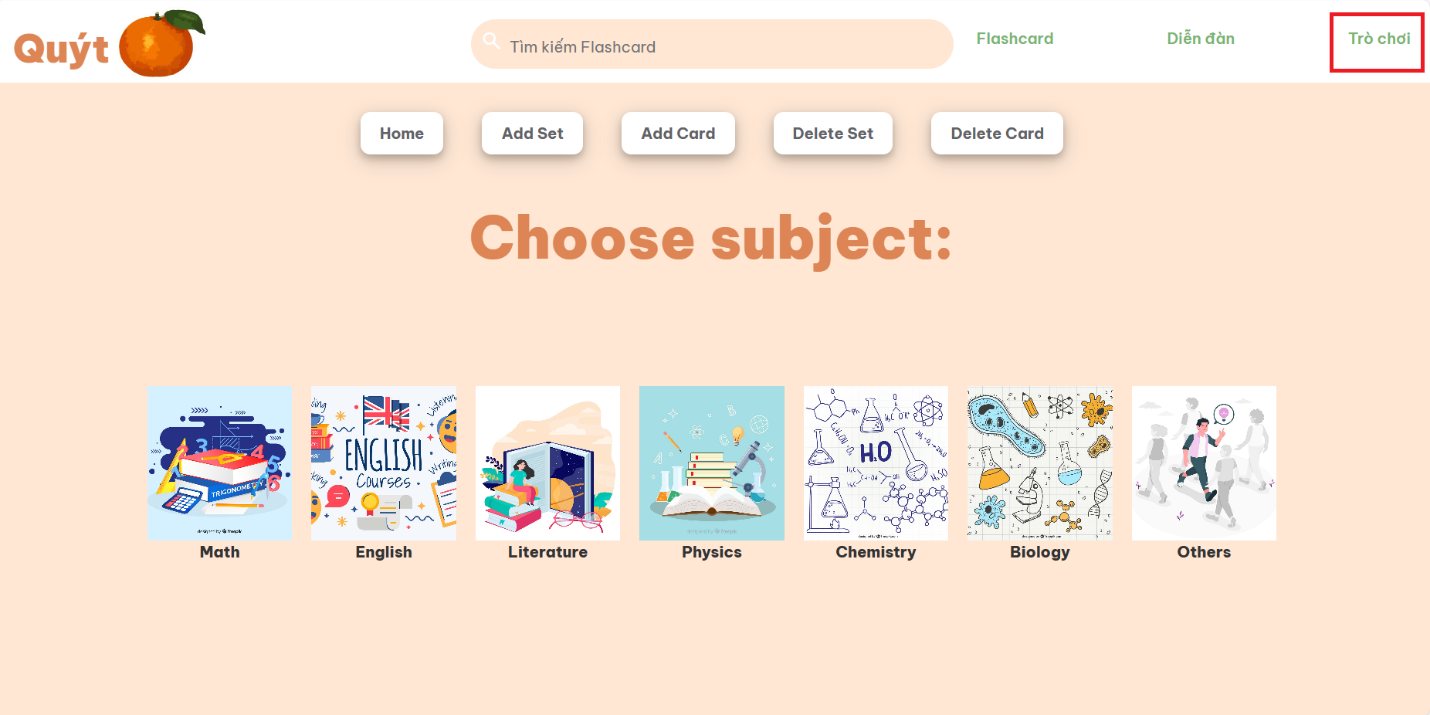
* Trong trường hợp muốn xem bài viết của người khác, click vào một bài bất kỳ trong phần “Các bài viết mới đây”, sau đó Web sẽ hiển thị bài viết mình muốn xem như hình bên dưới. Người dùng có thể tương tác trực tiếp với bài viết bằng cách comment/thả reaction



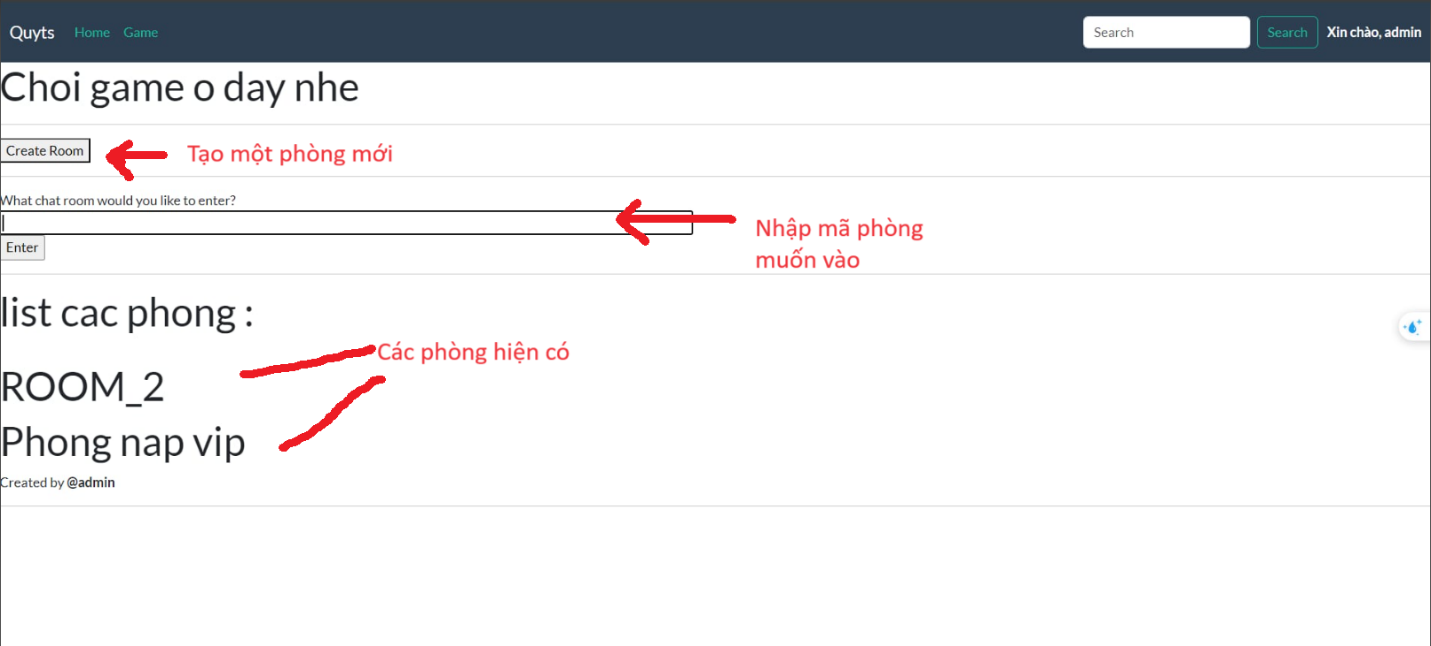
*34 - Xem/tương tác với bài viết*

## Game

Để có thể chơi game, người dùng có thể click vào phần “Trò chơi” ở góc phải trên cùng màn hình.



*35 - Trò chơi*

Sau khi click, một giao diện như bên dưới sẽ hiện ra. Người dùng có thể chọn “Create room” để tạo phòng mới hoặc nhập mã phòng muốn vào.

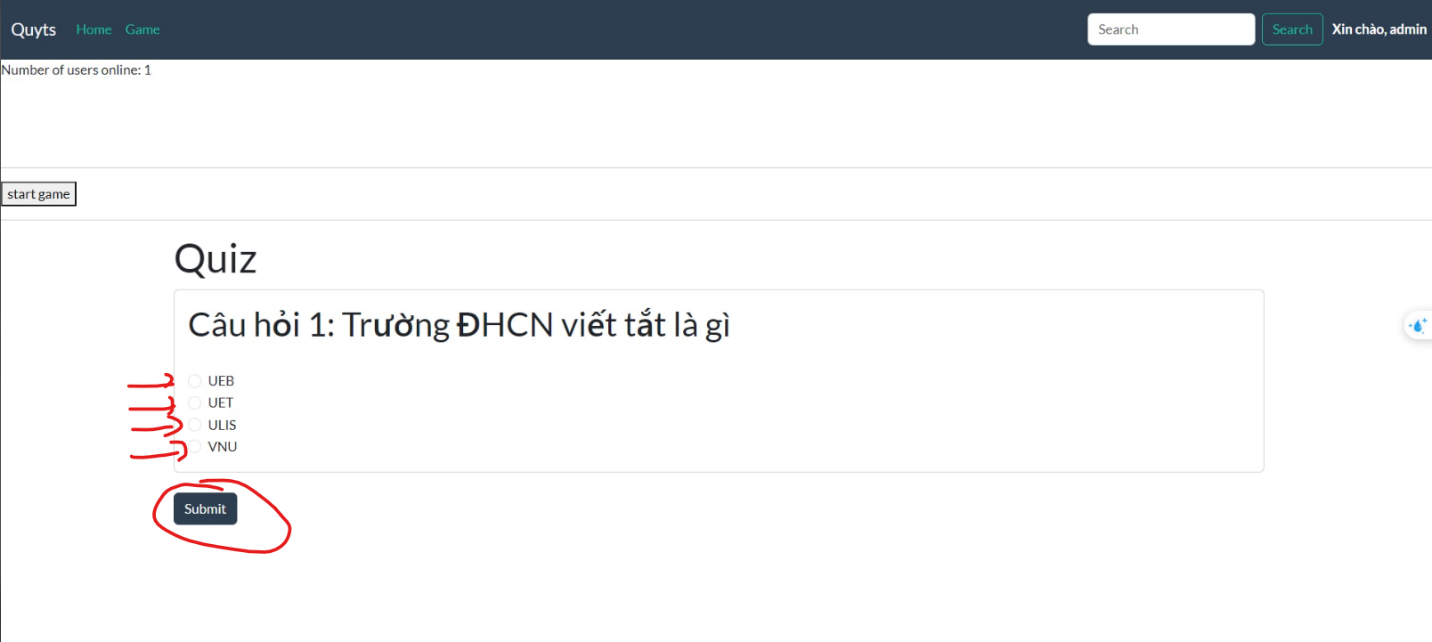
*36 - Tạo phòng*

* **Tạo phòng mới**: Khi nhấn “Create Room”, một giao diện sẽ hiển thị như sau và người dùng sẽ nhập thông tin theo quy định của giao diện



*37 - Phòng*

* **Nhập mã phòng và tham gia**: Khi nhập mã phòng và tham gia, người dùng sẽ thực hiện trả lời các câu hỏi của phòng chơi đó và nhấn “Submit” để nộp câu trả lời



*38 - Tham gia phòng*

# Thiết kế cơ sở dữ liệu

## Giới thiệu

Hệ thống của chúng tôi sử dụng PostgreSQL để quản lý cơ sở dữ liệu. Với tính năng truy vấn phức hợp và truy vấn xử lý song song, PostgreSQL giúp hệ thống của chúng tôi thực hiện các truy vấn dữ liệu từ kho dữ liệu khổng lồ về kiến thức của người dùng, từ đó mang lại trải nghiệm phản hồi nhanh chóng cho người dùng.

Bên cạnh đó, PostgreSQL còn có khả năng bảo mật với tính xác thực GSSAPI và khôi phục dữ liệu tốt bằng cơ chế khôi phục điểm theo thời gian (Point-in-time-recovery - PITR).

Với số lượng người dùng và lượng dữ liệu sinh ra khá là lớn, chúng tôi chọn sử dụng PostgreSQL vì khả năng mở rộng cao về cả số lượng dữ liệu và số lượng người dùng thao tác cùng lúc.

Trong PostgreSQL có ba khái niệm quan trọng:

* Table: Là một bảng gồm các cột (colum) và các hàng (row) thể hiện một thực thể trong cơ sở dữ liệu.
* Colum: Là các trường / thuộc tính của một thực thể trong cơ sở dữ liệu.
* Row: Là tập hợp các bản ghi của một thực thể.

## Thiết kế

Thực thể learner là nơi lưu trữ dữ liệu của người dùng.

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| id | Khóa chính |
| username\_validator | Thông tin xác thực tên người dùng |
| username | Tên đăng nhập người dùng |
| first\_name | Họ (Họ và tên) của người dùng |
| last\_name | Tên (Họ và tên) của người dùng |
| email | Email của người dùng |

Thực thể Subject là nơi lưu trữ dữ liệu về lĩnh vực kiến thức mà flashcard đó thuộc về.

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| id | Khóa chính |
| subject\_name | Tên môn học của flashcard |

Thực thể Tag là nơi lưu trữ thông tin về các bộ flashcard.

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| id | Khóa chính |
| name | Tên bộ |
| subject\_id | Tên môn học |
| owner\_id | Index của người sở hữu (FK) |

Thực thể Cards là nơi lưu trữ thông tin của một tấm thẻ flashcard.

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| id | Khóa chính |
| question | Câu hỏi / thuật ngữ |
| answer | Đáp án / định nghĩa |
| tag\_id | Index bộ Flashcard (FK) |

Thực thể Post là nơi lưu trữ thông tin các bài đăng trên forum.

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| id | Khóa chính |
| post\_title | Tiêu đề |
| post\_content | Nội dung |
| post\_like | Số lượt thích |
| post\_dislike | Số lượt không thích |
| pub\_date\_post | Ngày đăng bài |
| learner\_id | Index của người đăng |

Thực thể Comment là nơi lưu trữ các bình luận của bài đăng trên forum.

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| id | Khóa chính |
| post | Bài đăng |
| comment\_context | Nội dung bình luận |
| comment\_like | Lượt thích |
| comment\_dislike | Lượt không thích |
| pub\_date\_comment | Thời gian comment |
| learner\_id | Index của người comment |

Thực thể Quiz là nơi lưu trữ thông tin các bộ câu hỏi phục vụ cho phần chơi game.

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| id | Khóa chính |
| title | Tiêu đề của bộ câu hỏi |
| description | Mô tả của bộ câu hỏi |
| author\_id | Tên người tạo bộ câu hỏi |

Thực thể Room là nơi lưu trữ các phòng chơi game.

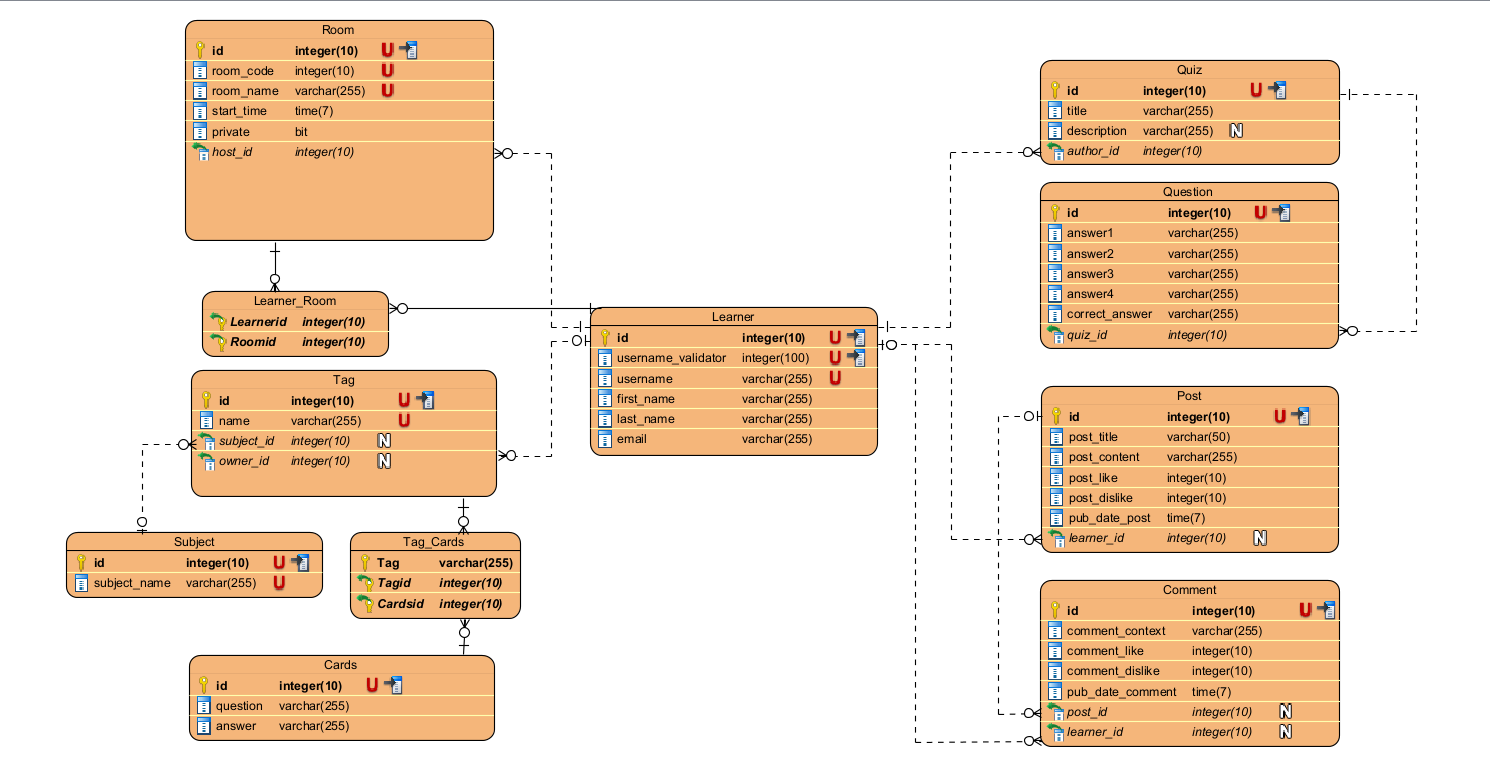
|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| id | Khóa chính |
| room\_code | Mã để vào phòng chơi |
| room\_name | Tên phòng chơi |
| host\_id | Người tạo phòng |
| participants | Người tham gia phòng chơi |
| start\_time | Thời gian bắt đầu chơi |
| quiz\_id | Bộ câu hỏi dùng trong phòng chơi |
| private | Public hay private phòng chơi |

Thực thể Question là nơi lưu trữ các câu hỏi trong một bộ Quiz

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| id | Khóa chính |
| quiz\_id | Bộ câu hỏi mà nó thuộc về |
| answer1 | Đáp án A |
| answer2 | Đáp án B |
| answer3 | Đáp án C |
| answer4 | Đáp án D |
| correct\_answer | Đáp án đúng |

## Tổng kết

Sau đây là biểu đồ trực quan cơ sở dữ liệu của chúng tôi:



*39 - Biểu đồ cơ sở dữ liệu*

# Giao diện ngoài

Để phục vụ cho việc tạo ra đáp án tự động ở phần trò chơi, chúng tôi quyết định ứng dụng AI. Bằng việc call API Gemini - một AI của google, chúng tôi có thể tạo ra đáp án tự động cho phần game

## Tổng quan về Gemini

Gemini là một mô hình ngôn ngữ lớn của Google AI. Nó được đào tạo trên một tập dữ liệu văn bản và mã khổng lồ, cho phép nó giao tiếp và tạo ra văn bản giống con người để đáp lại các câu hỏi và yêu cầu của con người.

Nó có thể thực hiện nhiều nhiệm vụ khác nhau, chẳng hạn như:

* Trả lời các câu hỏi một cách giàu thông tin, ngay cả khi chúng có kết thúc mở, mang tính thử thách hoặc lạ lùng.
* Tạo các định dạng văn bản sáng tạo khác nhau, như thơ, mã, kịch bản, bản nhạc, email, thư, v.v.
* Trả lời các câu hỏi của bạn một cách giàu thông tin, ngay cả khi chúng có kết thúc mở, mang tính thử thách hoặc lạ lùng.

## Các bước thực hiện

* Cài đặt môi trường và lấy token để truy cập vào API.
* Tạo phản hồi chữ từ yêu cầu chữ (Generate \*\*text from text \*\*input).