\*Entity:

-Comment:

+ Bổ sung @NotNull cho thuộc tính post, user, content.

+Bổ sung @Size(min=1, max=1000)

-Like: Bổ sung hàm prePersist và validate cho comment để like chỉ 1 trong hai

-Post:

+Bổ sung @NotNull cho user.

+Bổ sung @JsonIgnoreProperties({"posts", "comments", "likes"}).

+Bỏ columnDefinition = "TEXT" của imageUrl và bổ sung thêm @Lob cho imageUrl. +Bổ sung thêm @Column(name = "image\_content\_type") private String imageContentType và các hàm get set của imageContentType.

+Bổ sung thêm @Column(name = "status") private String status và @Column(name = "video\_url") private String videoUrl

+Đã xóa @OneToMany(mappedBy = "post", cascade = CascadeType.ALL, orphanRemoval = true) private List<Notification> notifications = new ArrayList<>();

+Hàm setUser() có throw new IllegalArgumentException

+Bổ sung @JsonIgnore cho mỗi Mapped

-User:

+ Bổ sung @Column(nullable = false) private String status và các hàm get set của status

+Bổ sung @Lob @Column(name = "avatar\_image") private String avatarImage và các hàm get set của avatarImage. //Đây là nơi lưu trữ ảnh mã hóa

+Bổ sung @Column(name = "avatar\_content\_type") private String avatarContentType và các hàm get set của avatarContentType

+Bổ sung @JsonIgnore cho mỗi Mapped

\*Service:

-Comment:

+Thêm hàm deleteComment

+Thêm hàm getAllComment

+Thêm hàm getCommentById

-Friendship:

+Thêm hàm deleteFriendship (để hủy kết bạn)

+Thêm hàm getAcceptedFriends (để lấy người dùng được gửi lời mời và lời mời ở trạng thái accepted)

-Like:

+Thêm hàm deleteLike (để không dislike)

-Post:

+Chỉnh sửa hàm createPost, không truyền tham số Post và truyền vào từng thuộc tính bao gồm String content, MultipartFile imageFile, String videoUrl, String status, User user. Thêm phần mã hóa ảnh rồi lưu vào ImageUrl lấy contentType của file rồi lưu vào ImageContentType.

+Hàm updatePost cũng thay đổi tham số truyền vào bằng Long id, String content, MultipartFile imageFile, String videoUrl, String status. Cũng có phần mã hóa file và lấy contentType lưu giống với hàm createPost.

+Thêm hàm deletePost để xóa bài đăng truyền tham số là id của bài đăng.

+Thêm hàm getPostById để lấy post theo id của post tham số truyền là id của post

+Thêm hàm ocuntPost dùng để đếm số lượng post có trong cơ sở dữ liệu, mục đích để hiển thị post ở phần giao diện mới vào của admin

-User:

+Hàm createUser đổi tên thảnhegisterUser. Thay đổi tham số truyền vào bằng các tham số String name, String email, String bio, String password, String status, MultipartFile avatarFile. Có mã hóa file và lưu vào AvatarImage, lấy contentType của file rồi lưu vào AvatarContentType cùng với các thuộc tính khác tất cả được set vào biến user rồi gọi hàm save của repository để truyền user vào

+Thêm hàm deleteUser để xóa người dùng. Tham số truyền vào là id của người dùng

+Thêm hàm findAll để lấy tất cả người dùng

+Thêm hàm updateUser. Tham số truyền vào bao gồm Long id, String name, String email, String password, String status, String bio, MultipartFile avatarFile. Bên trong hàm cũng có mã hóa file và lấy contentType của file cùng với các thuộc tính khác được set vào biến user rồi gọi hàm save của repository để truyền user vào.

+Thêm hàm countUser để đếm số lượng người dùng có trong cơ sở dữ liệu. Dùng để hiển thị số lượng người dùng ra phần giao diện mới vào của admin

\*Controller:

-Comment:

+Thêm hàm updateComment truyền tham số id của comment và biến comment chứa thông tin chỉnh sửa.

+Thêm hàm deleteComment để xóa comment theo id của comment

+Thêm hàm getAllComments để lấy toàn bộ comment từ cơ sở dữ liệu

-Friendship:

+Chỉnh sửa hàm createFriendship với @PostMapping("/addFriend"). Thay tham số truyền vào @RequestBody Friendship bằng các biến @RequestParam("requesterId") Long requesterId, @RequestParam("receiverId") Long receiverId, @RequestParam("status") String status. Set các tham số vào friendship rồi gọi hàm createFriendship để truyền friendship vào.

+Thêm hàm deleteFriendship với @DeleteMapping("/deleteFriend/{id}") để hủy kết bạn. Truyền tham số là id của friendship vào deleteFriendship của friendshipService.

+Bổ sung thêm hàm getFriendsAcceptedReceive. Truyền tham số là id của người dùng được chọn để xem danh sách bạn bè. Hàm sẽ gọi getAcceptedFriends và getFriends từ friendshipService để truyền tham số id vào, hai hàm trả về 2 list friendship mục đích là để gộp 2 list này lấy toàn bộ friendship có chứa id của người dùng ở cả requester và receiver

-Like:

+Thêm hàm deleteLike rồi truyền tham số id của like vào để dislike

-Post:

+Thay thế tham số truyền vào của hàm createPost gồm các @RequestParam có @RequestParam("content") String content,

@RequestParam(value = "imageFile", required = false) MultipartFile imageFile,

@RequestParam(value = "videoUrl", required = false) String videoUrl, @RequestParam("status") String status,

@RequestParam(value = "videoFile", required = false) MultipartFile videoFile

Có bước kiểm tra videoFile, nếu videoFile có tồn tại thì thực hiện lưu video vào tệp vật lí và lưu đường dẫn url của video vào mysql

+Thay thế tham số truyền vào của hàm updatePost gồm các @RequestParam có

@PathVariable Long id, @RequestParam("content") String content, @RequestParam(value = "imageFile", required = false) MultipartFile imageFile, @RequestParam(value = "videoUrl", required = false) String videoUrl, @RequestParam("status") String status,

@RequestParam(value = "videoFile", required = false) MultipartFile videoFile

Có bước lưu videoFile giống với hàm createPost

+Thêm hàm deletePost truyền vào tham số id của post để xóa post

+Thêm hàm countPost để đếm tổng số post

-User:

+Hàm register bổ sung thêm ‘consumes = MediaType.MULTIPART\_FORM\_DATA\_VALUE’ vào @PostMapping. Hàm thay thế tham số truyền vào @RequestBody user bằng các tham số sau:

@RequestParam("name") String name,

@RequestParam("email") String email,

@RequestParam(value="bio", required = false) String bio, @RequestParam("password") String password,

@RequestParam("status") String status,

@RequestParam(value="avatarFile", required = false) MultipartFile avatarFile

Sau đó set các tham số này vào biến user rồi truyền biến user vào hàm registerUser của userService

+Bổ sung thêm hàm editUser cũng có ‘consumes = MediaType.MULTIPART\_FORM\_DATA\_VALUE’ trong @PutMapping. Truyền các tham số sau:  
@PathVariable Long id,

@RequestParam("name") String name,

@RequestParam("email") String email,

@RequestParam(value = "bio", required = false) String bio,

@RequestParam(value = "password", required = false) String password, @RequestParam("status") String status,

@RequestParam(value = "avatarFile", required = false) MultipartFile avatarFile

Kiểm tra user có tồn tại thông qua id. Sau khi kiểm tra nếu có tồn tại thì truyền các tham số vào hàm updateUser của userService để cập nhật user

+Thêm hàm deleteUser truyền vào tham số là biến id của user

+Thêm hàm getAllUser để lấy toàn bộ user

+Thêm hàm countUser để đếm user

\*Các bổ sung khác

-Thêm tệp Uploads vô tệp src trong tệp backend

-File StaticResourceConfig trong tệp main/java