I. Reverse Engineer Instagram API

**1. Entities and Attributes**

Here are the JSON representations of the main entities:

User

{

"id": "string",

"username": "string",

"fullName": "string",

"bio": "string",

"website": "string",

"profilePicture": "string",

"followersCount": "integer",

"followingCount": "integer",

"postsCount": "integer"

}

Post

{

"id": "string",

"userId": "string",

"caption": "string",

"imageUrl": "string",

"videoUrl": "string",

"likesCount": "integer",

"commentsCount": "integer",

"createdAt": "string"

}

Comment

{

"id": "string",

"postId": "string",

"userId": "string",

"text": "string",

"createdAt": "string"

}

Message

{

"id": "string",

"senderId": "string",

"receiverId": "string",

"text": "string",

"createdAt": "string"

}

**2. Actions and Involved Entities**

Create a Post: User, Post

Like a Post: User, Post

Comment on a Post: User, Post, Comment

Follow a User: User

Send a Message: User, Message

**3. REST API Documentation**

Create a Post

State before: User is authenticated and ready to post a new image or video.

HTTP Method: POST

Endpoint: /posts

HTTP Body:

{

"userId": "string",

"caption": "string",

"imageUrl": "string",

"videoUrl": "string"

}

State after: A new post is created and accessible by followers of the user.

Like a Post

State before: User is authenticated and viewing a post they want to like.

HTTP Method: POST

Endpoint: /posts/{postId}/like

HTTP Body:

{

"userId": "string"

}

State after: The post's like count is incremented by one.

Comment on a Post

State before: User is authenticated and viewing a post they want to comment on.

HTTP Method: POST

Endpoint: /posts/{postId}/comments

HTTP Body:

{

"userId": "string",

"text": "string"

}

State after: A new comment is added to the post, and the comment count is incremented by one.

Follow a User

State before: User is authenticated and viewing another user they want to follow.

HTTP Method: POST

Endpoint: /users/{userId}/follow

HTTP Body:

{

"followerId": "string"

}

State after: The follower count of the followed user is incremented by one, and the following count of the follower is incremented by one.

Send a Message

State before: User is authenticated and wants to send a message to another user.

HTTP Method: POST

Endpoint: /messages

HTTP Body:

json

{

"senderId": "string",

"receiverId": "string",

"text": "string"

}

State after: A new message is sent and stored in both the sender's and receiver's message lists.