

## Service Layer Design (Updated)

*Update:* After reviewing the feedback on my service layer design, I have decided to follow the recommendations provided and alter the labels for my app's API endpoints. These alterations will be reflected in this document. While my app will potentially incorporate stretch features in the near future, these stretch features, to the best of my knowledge, will not result in the generation of new endpoints, and, as such, the service layer for my app will remain fundamentally unchanged.

The service layer will facilitate creating new profiles, retrieving profile data for users, updating profile details, and deleting profiles via API endpoints. The endpoints will be structured using the following criteria:

- POST - /register (Create new user)

- Request:

```
curl -X POST "https://petstore.swagger.io/v2/register" -H "accept: application/json" -H "Content-Type: application/json" -d '{"id": 0, "username": "string", "firstName": "string", "lastName": "string", "email": "string", "password": "string", "phone": "string", "userStatus": 0}'
```

- Response:

- Code 200: Successful operation

```
{
  "code": 200,
  "type": "unknown",
  "message": "14881488271168"
}
```

```
access-control-allow-headers: Content-Type,api_key,Authorization
access-control-allow-methods: GET,POST,DELETE,PUT
access-control-allow-origin: *
content-type: application/json
date: Tue,27 Oct 2020 21:33:54 GMT
server: Jetty(9.2.9.v20150224)
status: 200
```

- UI Diagram:

The diagram shows a web browser window titled "Eye Profile". Inside the window, there is a navigation bar with links: "Home", "Active Timer", "Rest Timer", and "Profile Page". The main content area is titled "Login to Your Profile" and contains a login form. The form has two input fields: "Username" and "Password". Below these fields are two buttons: "Login" and "Create New Profile". The "Create New Profile" button is highlighted with an oval. At the bottom of the window is a "Footer" section.

- GET - /login (Logs user into the system)

- Request:

```
curl -X GET "https://petstore.swagger.io/v2/login?username=test&password=testpass"
-H "accept: application/json"
```

- Response:

- Code 200: Successful operation

```
{
  "code": 200,
  "type": "unknown",
  "message": "logged in user session:1603845810463"
}
```

```
access-control-allow-headers: Content-Type,api_key,Authorization
access-control-allow-methods: GET,POST,DELETE,PUT
access-control-allow-origin: *
content-type: application/json
date: Wed,28 Oct 2020 00:43:30 GMT
server: Jetty(9.2.9.v20150224)
status: 200
x-expire-after: Wed Oct 28 01:43:30 UTC 2020
x-rate-limit: 5000
```

- Code 400: Invalid username/password provided

```
{
  "code": 1,
  "type": "error",
  "message": "Invalid username/password provided"
}
```

```
access-control-allow-headers: Content-Type,api_key,Authorization
access-control-allow-methods: GET,POST,DELETE,PUT
access-control-allow-origin: *
content-type: application/json
date: Wed,28 Oct 2020 00:43:30 GMT
server: Jetty(9.2.9.v20150224)
status: 400
```

- UI Diagram:

The diagram illustrates the layout of the 'Eye Profile' web page. At the top, there is a header bar with the title 'Eye Profile'. Below the header is a navigation menu with four links: 'Home', 'Active Timer', 'Rest Timer', and 'Profile Page'. The main content area features a central box titled 'Login to Your Profile'. Inside this box, there are two input fields labeled 'Username' and 'Password'. Below these fields are two buttons: 'Login' and 'Create New Profile'. The 'Login' button is highlighted with an oval. At the bottom of the page is a 'Footer' section.

- GET - / (Get user by username for printout)

- Request:

```
curl -X GET "https://petstore.swagger.io/v2/" -H "accept: application/json"
```

- Response:

- Code 200: Successful operation

```
{
  "id": 0,
  "username": "string",
  "firstName": "string",
  "lastName": "string",
  "email": "string",
  "password": "string",
  "phone": "string",
  "userStatus": 0
}
```

```
access-control-allow-headers: Content-Type,api_key,Authorization
access-control-allow-methods: GET,POST,DELETE,PUT
access-control-allow-origin: *
content-type: application/json
date: Tue,27 Oct 2020 21:33:54 GMT
server: Jetty(9.2.9.v20150224)
status: 200
```

- Code 400: Invalid username supplied

```
{
  "code": 1,
  "type": "error",
  "message": "Invalid username supplied"
}
```

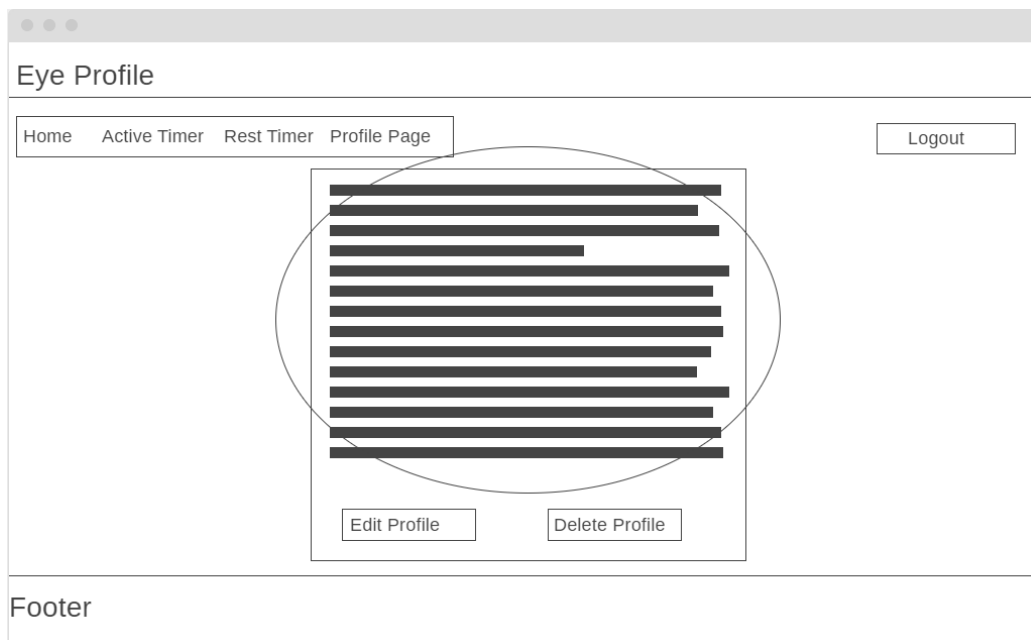
```
access-control-allow-headers: Content-Type,api_key,Authorization
access-control-allow-methods: GET,POST,DELETE,PUT
access-control-allow-origin: *
content-type: application/json
date: Wed,28 Oct 2020 00:43:30 GMT
server: Jetty(9.2.9.v20150224)
status: 400
```

- Code 404: User not found

```
{
  "code": 1,
  "type": "error",
  "message": "User not found"
}
```

```
access-control-allow-headers: Content-Type,api_key,Authorization
access-control-allow-methods: GET,POST,DELETE,PUT
access-control-allow-origin: *
content-type: application/json
date: Wed,28 Oct 2020 03:35:12 GMT
server: Jetty(9.2.9.v20150224)
status: 404
```

- UI Diagram:



- GET - /logout (Logs out current logged in user session)

- Request:

```
curl -X GET "https://petstore.swagger.io/v2/logout" -H "accept: application/json"
```

- Response:

- Code 200: Successful operation

```
{  
  "code": 200,  
  "type": "unknown",  
  "message": "ok"  
}
```

```
access-control-allow-headers: Content-Type,api_key,Authorization  
access-control-allow-methods: GET,POST,DELETE,PUT  
access-control-allow-origin: *  
content-type: application/json  
date: Wed,28 Oct 2020 20:24:23 GMT  
server: Jetty(9.2.9.v20150224)  
status: 200
```

- UI Diagram:



- PUT - /{username} (Updates user information)

- Request:

```
curl -X PUT "https://petstore.swagger.io/v2/test" -H "accept: application/json" -H "Content-Type: application/json" -d "{ \"id\": 0, \"username\": \"string\", \"firstName\": \"string\", \"lastName\": \"string\", \"email\": \"string\", \"password\": \"string\", \"phone\": \"string\", \"userStatus\": 0}"
```

- Response:

- Code 200: Successful response

```
{
  "code": 200,
  "type": "unknown",
  "message": "1"
}
```

```
access-control-allow-headers: Content-Type,api_key,Authorization
access-control-allow-methods: GET,POST,DELETE,PUT
access-control-allow-origin: *
content-type: application/json
date: Wed,28 Oct 2020 20:47:05 GMT
server: Jetty(9.2.9.v20150224)
status: 200
```

- Code 400: Invalid user supplied

```
{
  "code": 1,
  "type": "error",
  "message": "Invalid username supplied"
}
```

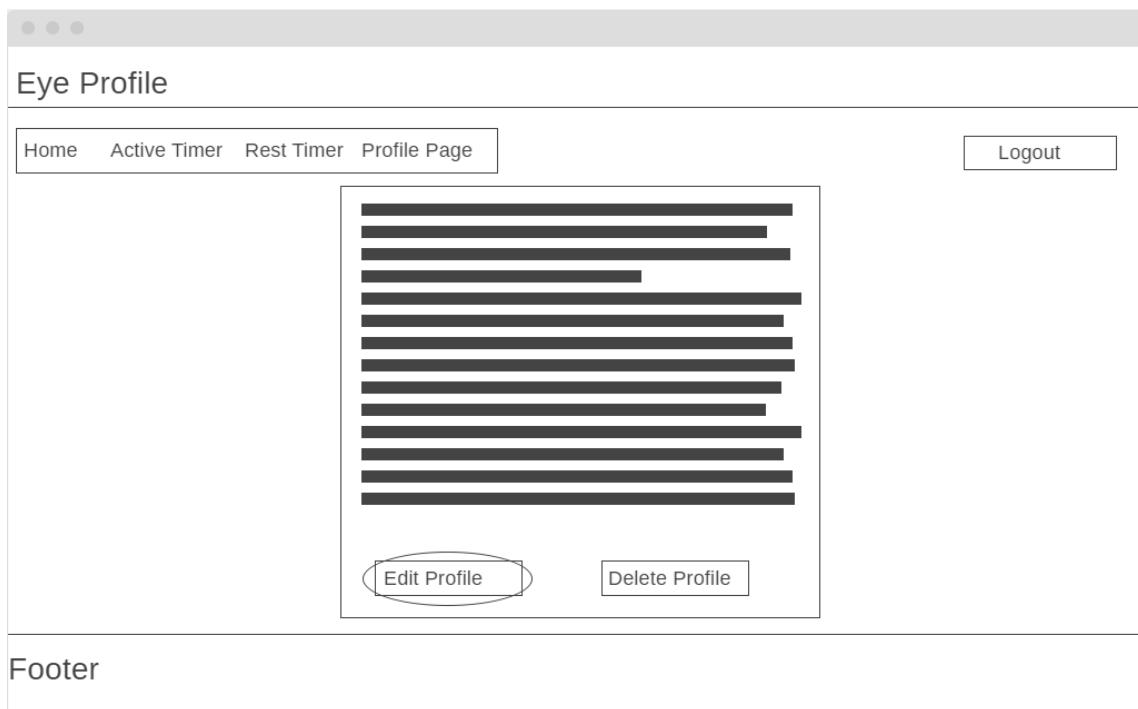
```
access-control-allow-headers: Content-Type,api_key,Authorization
access-control-allow-methods: GET,POST,DELETE,PUT
access-control-allow-origin: *
content-type: application/json
date: Wed,28 Oct 2020 00:43:30 GMT
server: Jetty(9.2.9.v20150224)
status: 400
```

- Code 404 User not found

```
{
  "code": 1,
  "type": "error",
  "message": "User not found"
}
```

```
access-control-allow-headers: Content-Type,api_key,Authorization
access-control-allow-methods: GET,POST,DELETE,PUT
access-control-allow-origin: *
content-type: application/json
date: Wed,28 Oct 2020 03:35:12 GMT
server: Jetty(9.2.9.v20150224)
status: 404
```

- UI Diagram:



- DELETE - /{username} (Deletes user)

- Request:

```
curl -X DELETE "https://petstore.swagger.io/v2/test" -H "accept: application/json"
```

- Response



- Code 200: Successful operation

```
{
  "code": 200,
  "type": "unknown",
  "message": "1"
}
```

```
access-control-allow-headers: Content-Type,api_key,Authorization
access-control-allow-methods: GET,POST,DELETE,PUT
access-control-allow-origin: *
content-type: application/json
date: Wed,28 Oct 2020 20:47:05 GMT
server: Jetty(9.2.9.v20150224)
status: 200
```

- Code 400: Invalid user supplied

```
{
  "code": 1,
  "type": "error",
  "message": "Invalid username supplied"
}
```

```
access-control-allow-headers: Content-Type,api_key,Authorization
access-control-allow-methods: GET,POST,DELETE,PUT
access-control-allow-origin: *
content-type: application/json
date: Wed,28 Oct 2020 00:43:30 GMT
server: Jetty(9.2.9.v20150224)
status: 400
```

- Code 404: User not found

```
{
  "code": 1,
  "type": "error",
  "message": "User not found"
}
```

```
access-control-allow-headers: Content-Type,api_key,Authorization
access-control-allow-methods: GET,POST,DELETE,PUT
access-control-allow-origin: *
content-type: application/json
date: Wed,28 Oct 2020 03:35:12 GMT
server: Jetty(9.2.9.v20150224)
status: 404
```

- UI Diagram:

