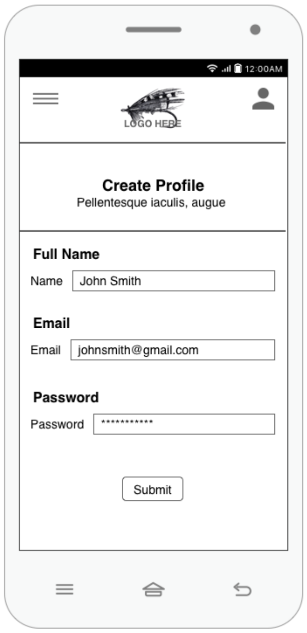
Matt Strayer

Service Layer Design

For the Fly Tying Companion, we will use a simple API service layer to help communicate between the UI layer and data layer of the application. The service layer will use HTTP protocols like POST, PUT, GET, and DELETE to retrieve and send information between the data and the user's screen. We'll look at each of the interactions and requests made to different endpoints used in the application broken down page by page.

Register Account Page



Create Profile
Pellentesque iaculis, augue

Full Name
Name: John Smith

Email
Email: johnsmith@gmail.com

Password
Password: ***********

Create Account – Endpoint 1

The user, if this is the first time using the application, will have the option of creating an account so that their information will be saved. This will use the POST HTTP method with the user info in JSON format being sent to the User account section of the database. This will also take them to the user Example:

Request:

```
{
 "userName" : "my_username",
   "email" : "user@example.test",
 "password" : {      "value" : "my_password"
```

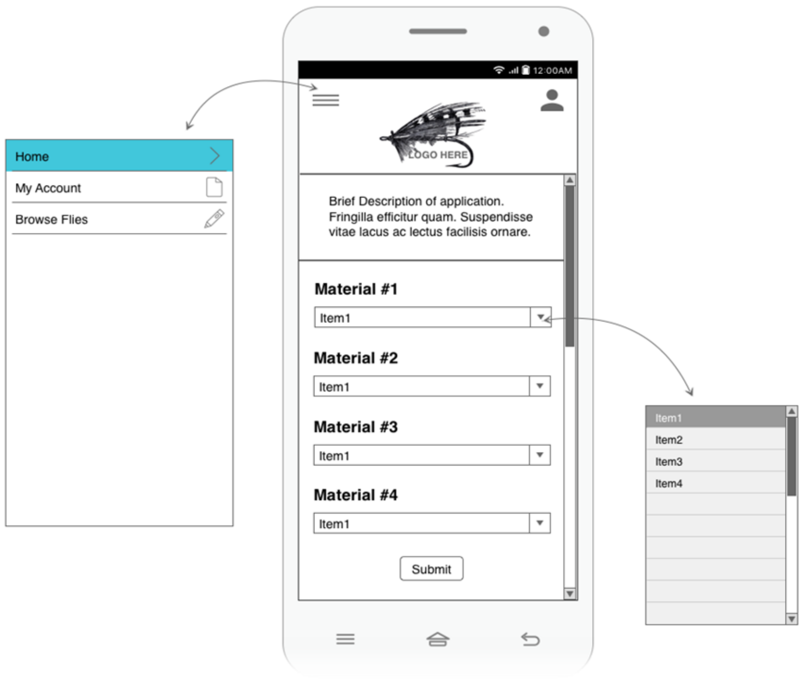
```
  },
   "active" : true
}
```

Responses:

200 – Request succeeded

405 – Invalid Input

<u>Home Page</u>



Logging in to Account – Endpoint 2

The user can log in to the application to access saved material. This functions as a GET Request that sends user information including a username and password, and if verified will log the user in and return a response saying it was successful or unsuccessful.

Request:

```
{
```

“username” : “username”,

“password” : “password”

}

Response:

200 – request succeeded

400 – Invalid Username/Password

Logging out of Account – Endpoint 3

The user logs out of the application. This is a GET Request that logs a user out of their account.

Response:

200 – request succeeded

Submitting Materials – Returning Flies – Endpoint 4

The user will input materials and then hit submit. Each material item will have their own ID and the submit button will initiate a GET request to the database. The request will contain the IDs and will return the flies with material IDs that match.

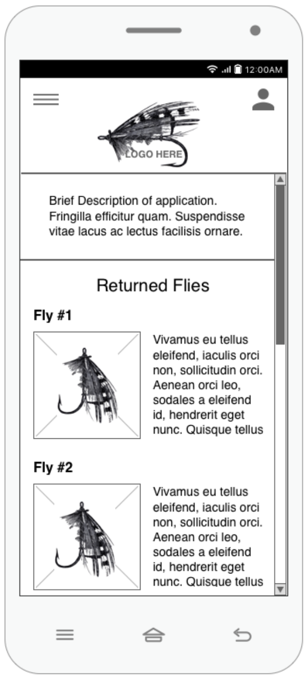
Request:

```
{
“materialIDList” : [
        “materialID”, “materialID”, “materialID”, “materialID”
        ]
}
```

Response:

```
{
Flies : [
fly: {
    flyID: “fly_ID”
    flyName: "Fly_name”,
    flyPicture: “picture”,
    flyType: "Fly_Type”,
]
}
```

This will refresh home page to display this information:


12:00AM
LOGO HERE
Brief Description of application. Fringilla efficitur quam. Suspendisse vitae lacus ac lectus facilisis ornare.
Returned Flies
Fly #1
Vivamus eu tellus eleifend, iaculis orci non, sollicitudin orci. Aenean orci leo, sodales a eleifend id, hendrerit eget nunc. Quisque tellus
Fly #2
Vivamus eu tellus eleifend, iaculis orci non, sollicitudin orci. Aenean orci leo, sodales a eleifend id, hendrerit eget nunc. Quisque tellus

Getting Individual Fly info – Endpoint 5

This functions as a GET Request that can be accessed on any page that fly information is presented, including the Home, User Account Page, and Browse Flies page. It takes a Fly ID that is captured from clicking on a fly and takes it to an individual page where you will see more detailed information.

Request:

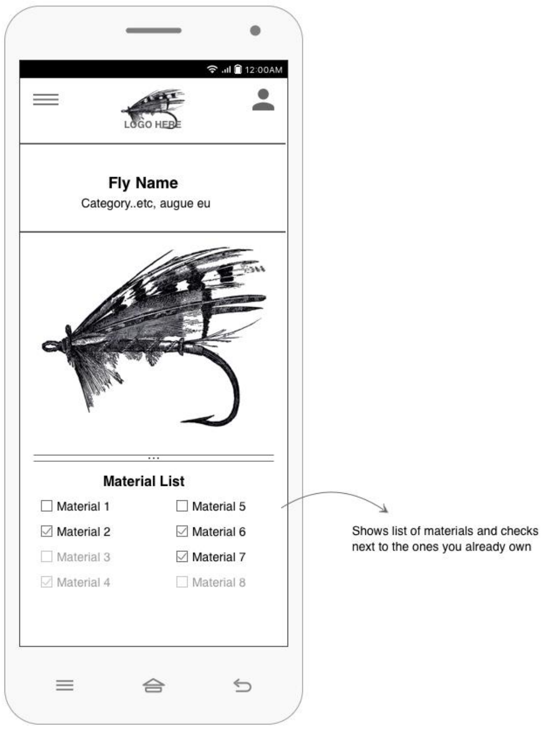
```
{
“flyID” : “flyIDString”
}
```

Response:

200 – request succeeded

```
fly: {
    flyID: Int
    flyName: String,
    flyPicture: String,
    flyType: String,
    materials: [
       {
       materialID: Int
       materialName: String,
       materialType: String,
       }
    ]
  }
```

This information will be displayed on the following page:



Individual Fly Page

Getting User Material – Endpoint 6

When a user is on the individual fly page they see the materials that are included in that fly's pattern, as well as which of those materials they have saved in their account/inventory. This will use a GET request to get user information.

Request:

{

"userID" : "userID"

"userMaterialList" : "userMaterialList"

}

Response:

404 – Not Found

200 – Request Succeeded

{

"userID" : "user_ID"
"userMaterialList" : [

"material" : "material1",

"material" : "material2",

Etc]

}

Browse Flies Page

Getting Flies by Category – Endpoint 7

On this page the user can browse through flies that are arranged according to their inventory.

Request:

{

Fly: {

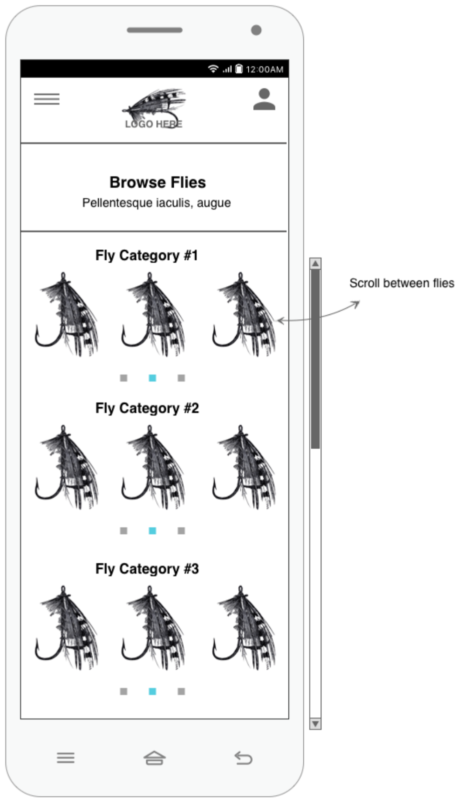
```
        “FlyCategory” : “FlyCategory”
        }
}
```

Response:

200 – Request Succeeded

```
{
“Fly” : {
        “FlyCategory” : [
                “FlyID” : “FlyID”,
                “FlyName” : “FlyName”,
                “FlyPicture” : “FlyPicture”,
                ]
        }
}
```

The flies will be displayed as shown below:



Account Page

Get User information – Endpoint 8

This page will show all relevant user information. It will display a profile name, their saved materials, as well as a list of flies they've saved to their account.

Request:

```
user: {
      userID: "ID",
      userName: "userName",
      email: "userEmail",
      userMaterials:  []
}
```

Response:

200 – Request Succeeded

404 – Not Found

```
User: {
          “userName” : “userName”,
          “email” : “userEmail”,
           “userMaterials” :  [
        {
           materialID: Int,
           materialName: String,
          materialType: String,
          }
          }
}
```

Updating Account – Endpoint 9

This will allow the user to update info in their account including name, email and password.

Request:

```
{
“userID” : “userID”
 "userName" : "my_username",
   "email" : "user@example.test",
 "password" : {      "value" : "my_password"
  },
   "active" : true
}
```

Responses:

200 – Request succeeded

405 – Invalid Input

Deleting Account – Endpoint 10

This will allow a user to delete their account and all saved information.

Request:

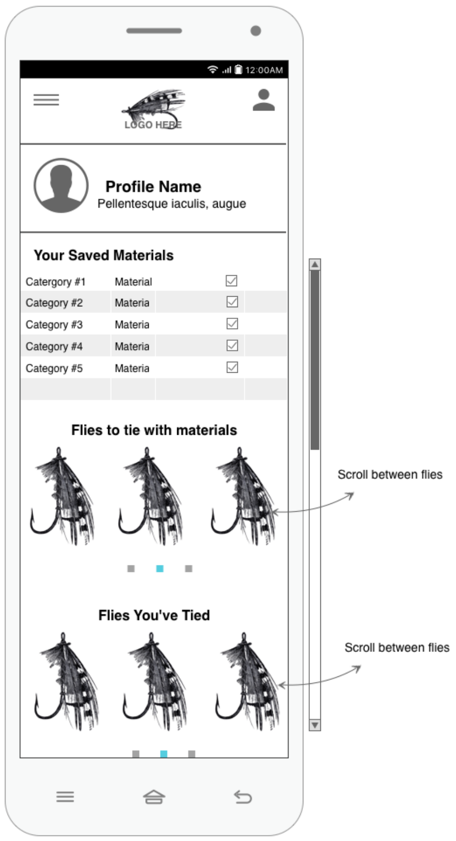
```
{
 "userID" : "userID"
}
```

Responses:

200 – Request succeeded

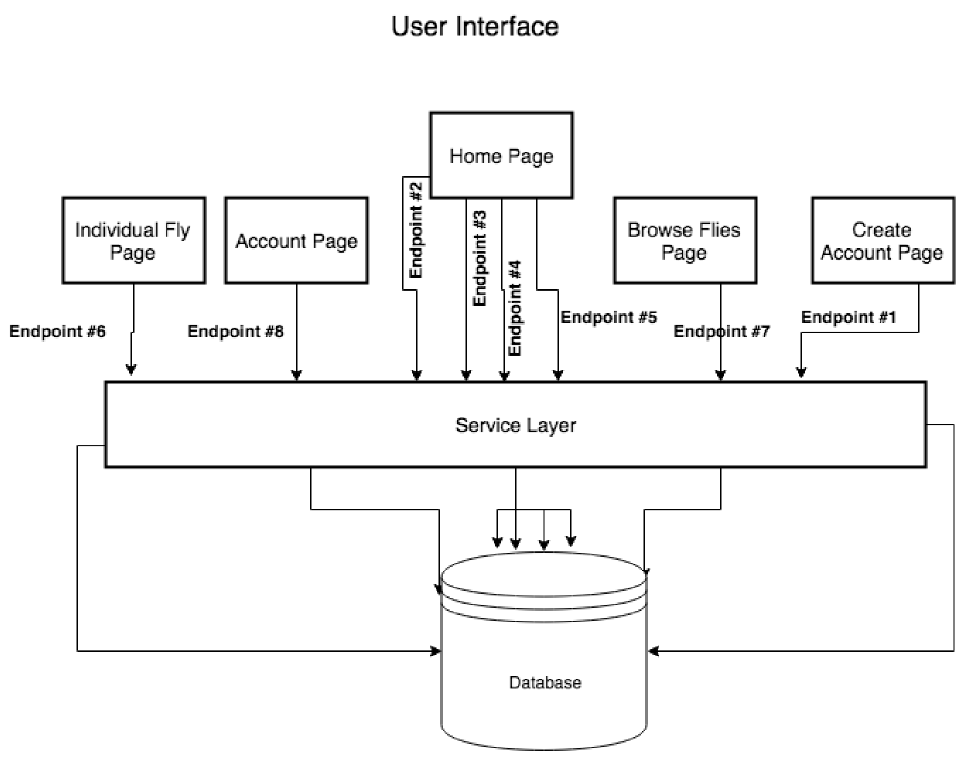
405 – Invalid Input

The user page:



Catergory #1	Material	✓
Category #2	Materia	✓
Category #3	Materia	✓
Category #4	Materia	✓
Category #5	Materia	✓

Endpoint Diagram


User Interface
Home Page
Individual Fly Page
Account Page
Browse Flies Page
Create Account Page
Endpoint #2
Endpoint #3
Endpoint #4
Endpoint #5
Endpoint #6
Endpoint #8
Endpoint #7
Endpoint #1
Service Layer
Database