



# Pilot Logbook Manager

Jennifer Sebek Kemp

March 25, 2023

## Design Service Layers

### Overview

This application will utilize the Django REST Framework to create the API. Routes will point to specific endpoints, and service layers will be used when business logic is needed to manipulate the data returned from the database. I plan to use Django authentication to handle user registration, sign-in, and logout. Once the user is signed in, they will be redirected to the user home page which will have links to the main functionality of the application.

### Specifications

I have used example URL's since the project is not currently hosted and have created a sample set of the data locally for the examples. The following are the planned endpoints and how they will be utilized:

#### Account Detail

1. Get Account Detail

Method: GET

URL: <https://logbookmanager.com/logbook/account-detail/>

Purpose: When the user is logged into the application, they will be able to click on the link for their account detail and this endpoint will be called. If they have already entered their account details, the data will render on the page.

Example requests:

```
curl --request GET --url https://logbookmanager.com/logbook/account-detail/
```

### Sample Success Response:

```
$ curl -i http://127.0.0.1:8000/account/
HTTP/1.1 200 OK
Date: Sat, 25 Mar 2023 23:54:45 GMT
Server: Werkzeug/0.2 CPython/3.10.10
Content-Type: application/json
Vary: Accept, Cookie
Allow: GET, POST, HEAD, OPTIONS
X-Frame-Options: DENY
Content-Length: 134
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin

[{"pilot":1,"id":1,"first_name":"Sample","last_name":"Pilot","age":45,"pilot_in_command":true,"current_aircraft_type":"Multi-engine"}]
```

### Sample Error Response:

```
{
  "status": 404,
  "response": "Account detail not completed for this account yet."
}
```

### Diagram:



## 2. Create a new account detail record:

Method: POST

URL: <https://logbookmanager.com/logbook/account-detail/>

Purpose: When the user is logged into the application and has navigated to the account detail page, they will be able to fill out the form for their account detail and this endpoint will be called. Their account detail data will then be rendered on the page.

Example requests:

```
curl -i https://logbookmanager.com/logbook/account-detail/ \
```

-X POST \

-H 'Content-Type: application/json' \

-d '{"pilot": 1,"first\_name": "Another","last\_name": "Pilot","age": 25,  
"pilot\_in\_command": false,"current\_aircraft\_type": "single-engine"}' \

-w '\n'

Sample Success Response:

```
$ curl -i http://127.0.0.1:8000/account/ \
-X POST \
-H 'Content-Type: application/json' \
-d '{"pilot": 1,"first_name": "Another","last_name": "Pilot","age": 25, "pilot_in_command": false,"current_aircraft_type": "single-engine"}' \
-w '\n'
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 271 100 135 100 136 8801 8866 --:--:-- --:--:-- --:--:-- 19357HTTP/1.1 201 Created
Date: Sun, 26 Mar 2023 00:35:41 GMT
Server: WSGIServer/0.2 CPython/3.10.10
Content-Type: application/json
Vary: Accept, Cookie
Allow: GET, POST, HEAD, OPTIONS
X-Frame-Options: DENY
Content-Length: 135
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
{"pilot":1,"id":2,"first_name":"Another","last_name":"Pilot","age":25,"pilot_in_command":false,"current_aircraft_type":"single-engine"}
```

Sample Error Response:

```
{
  "status": 400,
  "response": "Missing required fields. Please fill out all information."
}
```

Diagram:



### 3. Update Account Detail

Method: PUT

URL: [https://logbookmanager.com/logbook/account-detail/<account-id>/](https://logbookmanager.com/logbook/account-detail/<account-id>)

Purpose: When the user is logged into the application and has navigated to the account detail page, they will be able to click update for their account detail. They will get a form populated with the current data, after making changes to their data they can click save and this endpoint will be called. Their updated account detail data will then be rendered on the page.

Example Request:

```
curl -i https://logbookmanager.com/logbook/account-detail/2/ \
```

```
-X PUT \
```

```
-H 'Content-Type: application/json' \
```

```
-d '{"pilot": 2, "first_name": "Another", "last_name": "Pilot", "age": 35, "pilot_in_command": false, "current_aircraft_type": "single-engine"}' \
```

```
-w '\n'
```

Sample Success Response:

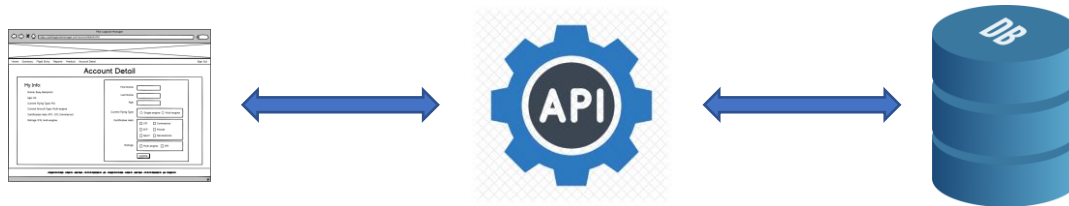
```
$ curl -i http://127.0.0.1:8000/account/2/ \
-X PUT \
-H 'Content-Type: application/json' \
-d '{"pilot": 2, "first_name": "Another", "last_name": "Pilot", "age": 35, "pilot_in_command": false, "current_aircraft_type": "single-engine"}' \
-w '\n'
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left  Speed
100 271    100 135 100 136   9866   9939  --:--:-- --:--:-- --:--:-- 20846HTTP/1.1 200 OK
Date: Sun, 26 Mar 2023 01:00:13 GMT
Server: WSGIServer/0.2 CPython/3.10.10
Content-Type: application/json
Vary: Accept, Cookie
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
X-Frame-Options: DENY
Content-Length: 135
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin

{"pilot":2,"id":2,"first_name":"Another","last_name":"Pilot","age":35,"pilot_in_command":false,"current_aircraft_type":"single-engine"}
```

Sample Error Response:

```
{
  "status": 400,
  "response": "Missing required fields. Please fill out all information."
}
```

Diagram:



## Flights

#### 4. Get Flights

Method: GET

URL: <https://logbookmanager.com/logbook/flights/>

Purpose: When the user is logged into the application, they will be able to click on the link for their flight summary and this endpoint will be called. If they have already entered flight data in the database, a list of their flight data will render on the page.

The reports page will also utilize this endpoint to gather data for generating reports. The user will select the type of report they want to view and click submit, and this endpoint will be called. A reports service will perform the business logic on the data and return the report information to the page.

Example requests:

```
curl --request GET --url https://logbookmanager.com/logbook/flights/
```

Sample Success Response:

[illegible]

Sample Error Response:

 $\{$ 

```
"status" : 404,
```

“response” : “No flights entered for this account yet.”

}

Diagram:

Summary Page:



##### 5. Create a new flight record:

Method: POST

URL: <https://logbookmanager.com/logbook/flights/>

Purpose: When the user is logged into the application and has navigated to the flight entry page, they will be able to fill out the form with their flight information and click submit and this endpoint will be called. They will receive a message that their data was successfully entered.

Example requests:

```
curl -i https://logbookmanager.com/logbook/flights/ \
```

```
-X POST \
```

```
-H 'Content-Type: application/json' \
```

```
-d '{"pilot": 2,"flight_date": "2023-03-21","origin": "ord","destination":  
"tys","tail_number": 894,"total_time": 3.2,"landings": 1,"multi_engine_time":  
3.2,"single_engine_time": 0.0,"vfr_time": 3.0,"ifr_time": 0.2,"night_time":  
0.0,"notes": "ord-tys"}' \
```

```
-w '\n'
```

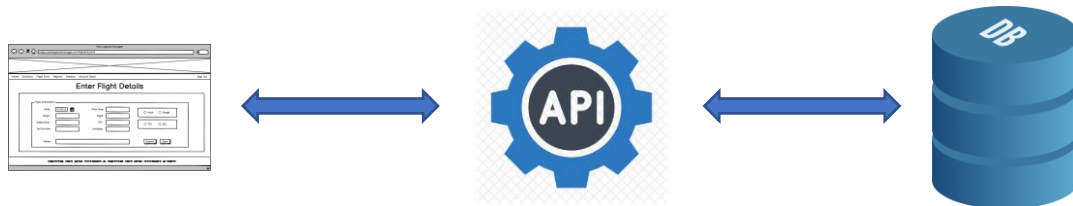
## Sample Success Response:

```
curl -i http://127.0.0.1:8000/flights/ \
-X POST \
-H 'Content-Type: application/json' \
-d '{"pilot": 2, "flight_date": "2023-03-21", "origin": "ord", "destination": "tys", "tail_number": 894, "total_time": 3.2, "landings": 1, "multi_engine_time": 3.2, "single_engine_time": 0.0, "vfr_time": 3.0, "ifr_time": 0.2, "flight_time": 0.0, "notes": "ord-tys"}' \
-w '%{total} \n'
% Total % Received % Xferd Average Speed Time Time Time Current
100 400 100 242 100 242 12713 13028 --:--:-- --:--:-- --:--:-- 27222HTTP/1.1 201 Created
Date: Sun, 26 Mar 2023 01:30:59 GMT
Server: Werkzeug/0.2.4 Python/3.10.10
Content-Type: application/json
Vary: Accept, Cookie
Allow: GET, POST, HEAD, OPTIONS
X-Frame-Options: DENY
Content-Length: 242
X-Content-Type-Options: nosniff
Referer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
{"pilot":2,"flight_date":"2023-03-21","origin":"ord","destination":"tys","tail_number":894,"total_time":3.2,"landings":1,"multi_engine_time":3.2,"single_engine_time":0.0,"vfr_time":3.0,"ifr_time":0.2,"flight_time":0.0,"notes":"ord-tys"}
```

## Sample Error Response:

```
{
  "status": 400,
  "response": "Missing required fields. Please fill out all information."
}
```

## Diagram:



## 6. Update Flight Data

Method: PUT

URL: [https://logbookmanager.com/logbook/flights/<flight-id>/](https://logbookmanager.com/logbook/flights/<flight-id>)

Purpose: When the user is logged into the application and has navigated to the summary page, they will see a list of flights and be able to click update for a flight. They will get a form populated with the current data, after making changes to their data they can click save and this endpoint will be called. Their updated flight data will then be rendered on the page.

## Example Request:

```
curl -i https://logbookmanager.com/logbook/flights/2/ \
```

-X PUT \

-H 'Content-Type: application/json' \

-d '{"pilot": 2,"flight\_date": "2023-03-22","origin": "ord","destination":  
"tys","tail\_number": 894,"total\_time": 3.2,"landings": 1,"multi\_engine\_time":  
3.2,"single\_engine\_time": 0.0,"vfr\_time": 3.0,"ifr\_time": 0.2,"night\_time":  
0.0,"notes": "ord-tys"}' \

-w '\n'

Sample Success Response:

```
$ curl -i http://127.0.0.1:8000/flights/2/ \
-X PUT \
-H 'Content-Type: application/json' \
-d '{"pilot": 2,"flight_date": "2023-03-21","origin": "ord","destination": "tys","tail_number": 894,"total_time": 3.2,"landings": 1,"multi_engine_time": 3.2,"single_engine_time": 0.0,"vfr_time": 2.8,"ifr_time": 0.4,"night_time": 0.0,"notes": "ord-tys"}' \
-w '\n'
% total % received % xferd Average Speed Time Time Time Current
100 490 100 242 100 248 17305 17734 --:--:-- --:--:-- --:--:-- 37692HTTP/1.1 200 OK
Date: Sun, 26 Mar 2023 02:39:25 GMT
Server: WSGIServer/0.2 (Python/3.10.10)
Content-Type: application/json
Vary: Accept, Cookie
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
X-Frame-Options: deny
Content-Length: 242
X-Content-Type-Options: nosniff
Referer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
{"pilot":2,"id":2,"flight_date":"2023-03-21","origin":"ord","destination":"tys","tail_number":894,"total_time":3.2,"landings":1,"multi_engine_time":3.2,"single_engine_time":0.0,"vfr_time":2.8,"ifr_time":0.4,"night_time":0.0,"notes":"ord-tys"}
```

Sample Error Response:

```
{
  "status": 400,
  "response": "Missing required fields. Please fill out all information."
}
```

Diagram:



## 7. Delete Flight

Method: DELETE

URL: <https://logbookmanager.com/logbook/flights/<flight-id>/>

Purpose: When the user is logged into the application and has navigated to the summary page, they will see a list of flights and be able to click delete for a



flight they would like to remove. They will get a message to confirm the delete and if they click yes, this endpoint will be called. Their updated flight data will then be rendered on the page.

Example request:

```
curl -X DELETE https://logbookmanager.com/logbook/flights/4/
```

Sample Success Response:

```
jsebe@JenniferWork MINGW64 ~
$ curl -X DELETE http://127.0.0.1:8000/flights/4/
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           0         0     0         0          0      0      0     0
0         0         0     0         0          0      0      0     0

jsebe@JenniferWork MINGW64 ~
$ curl -i -X DELETE http://127.0.0.1:8000/flights/3/
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           0         0     0         0          0      0      0     0
0         0         0     0         0          0      0      0     0
Date: Sun, 26 Mar 2023 14:16:00 GMT
Server: WSGIServer/0.2 CPython/3.10.10
Vary: Accept, Cookie
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
X-Frame-Options: DENY
Content-Length: 0
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
0HTTP/1.1 204 No Content
```

Sample Error Response:

```
{
  "status": 404,
  "response": "That flight data was not found."
}
```

Diagram:



## Medical

### 8. Get Medical Information

Method: GET

URL: <https://logbookmanager.com/logbook/medical/>

Purpose: When the user is logged into the application, they will be able to click on the link for their medical information and this endpoint will be called. If they have already entered medical data in the database, their medical data will render on the page.

Example requests:

```
curl --request GET --url https://logbookmanager.com/logbook/medical/
```

Sample Success Response:

```
curl -i https://127.0.0.1:8000/medical/ -H "..."
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           % Done    0         0             0      0      0      0      0
100  443  100  443    0    104k    0      0      0      144kHTTP/1.1 200 OK
Date: Sun, 28 Mar 2023 18:11:16 GMT
Server: WSGIServer/0.2 Python/3.10.10
Content-Type: application/json
Vary: Accept, Cookie
Allow: GET, POST, HEAD, OPTIONS
X-Frame-Options: DENY
Content-Length: 483
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin

[{"pilot":1,"id":1,"medical_class":1,"medical_date":"2023-01-17","examiner_name":"Dr. Smith","examiner_phone":"865-555-1212","notes":"ecg done"}, {"pilot":3,"id":2,"medical_class":2,"medical_date":"2022-11-28","examiner_name":"Dr. Chen","examiner_phone":"865-555-1213","notes":"vision test"}, {"pilot":2,"id":3,"medical_class":2,"medical_date":"2023-03-05","examiner_name":"Dr. Green","examiner_phone":"865-555-1214","notes":"ecg due next visit"}]
```

Sample Error Response:

```
{
  "status": 404,
  "response": "No medical information entered for this account yet."
}
```

Diagram:



9. Create a new medical record:

Method: POST

URL: <https://logbookmanager.com/logbook/medical/>

Purpose: When the user is logged into the application and has navigated to the medical page, they will be able to fill out the form with their medical information and click submit and this endpoint will be called. They will receive a message that their data was successfully entered.

Example requests:

```
curl -i https://logbookmanager.com/logbook/medical/ \
```

```
-X POST \
```

```
-H 'Content-Type: application/json' \
```

```
-d '{"pilot": 1,"medical_class": 1, "medical_date": "2022-07-17","examiner_name": "Dr Smith","examiner_phone": "865-555-1212","notes": "ecg due next visit"}' \
```

```
-w '\n'
```

Sample Success Response:

```
$ curl -i http://127.0.0.1:8000/medical/ \
-X POST \
-H 'Content-Type: application/json' \
-d '{"pilot": 1,"medical_class": 1, "medical_date": "2022-07-17","examiner_name": "Dr Smith","examiner_phone": "865-555-1212","notes": "ecg due next visit"}' \
-w '\n'
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100    305    100    152    100    10157   10224  --:--:-- --:--:-- --:--:-- 21785HTTP/1.1 201 Created
Date: Sun, 26 Mar 2023 14:41:00 GMT
Server: WSGIServer/0.2 CPython/3.10.10
Content-Type: application/json
Vary: Accept, Cookie
Allow: GET, POST, HEAD, OPTIONS
X-Frame-Options: DENY
Content-Length: 152
X-Content-Type-Options: nosniff
Referer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin

{"pilot":1,"id":4,"medical_class":1,"medical_date":"2022-07-17","examiner_name":"Dr Smith","examiner_phone":"865-555-1212","notes":"ecg due next visit"}
```

Sample Error Response:

```
{
  "status" : 400,
  "response" : "Missing required fields. Please fill out all information."
}
```

Diagram:



## 10. Update Medical Record

Method: PUT

URL: <https://logbookmanager.com/logbook/medical/<medical-id>/>

Purpose: When the user is logged into the application and has navigated to the medical page, they will see a list of their medical history and be able to click update. They will get a form populated with the current data, after making changes to their data they can click save and this endpoint will be called. Their updated medical data will then be rendered on the page.

Example Request:

`curl -i http://logbookmanager.com/logbook/medical/4/ \`

`-X PUT \`

`-H 'Content-Type: application/json' \`

`-d '{"pilot": 1,"medical_class": 1,"medical_date": "2022-07-15","examiner_name": "Dr Smith","examiner_phone": "865-555-1212","notes": "ecg due next visit"}' \`

`-w '\n'`

Sample Success Response:

```

$ curl -i http://127.0.0.1:8000/medical/4/ \
-X PUT \
-H 'Content-Type: application/json' \
-d '{"pilot": 1,"medical_class": 1,"medical_date": "2022-07-15","examiner_name": "Dr Smith","examiner_phone": "865-555-1212","notes": "ecg due next visit"}' \
-w '\n'
100 304 100 152 100 152 10645 10645 --:--:-- --:--:-- --:--:-- 23384HTTP/1.1 200 OK
Date: Sun, 26 Mar 2023 14:47:24 GMT
Server: WSGIServer/0.2 CPython/3.10.10
Content-Type: application/json
Vary: Accept, Cookie
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
X-Frame-Options: DENY
Content-Length: 152
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
{"pilot":1,"id":4,"medical_class":1,"medical_date":"2022-07-15","examiner_name":"Dr Smith","examiner_phone":"865-555-1212","notes":"ecg due next visit"}
  
```

Sample Error Response:

```
{
  "status" : 400,
  "response" : "Missing required fields. Please fill out all information."
}
```

Diagram:



## 11.Delete Medical Record

Method: DELETE

URL: <https://logbookmanager.com/logbook/medical/<medical-id>/>

Purpose: When the user is logged into the application and has navigated to the medical page, they will see their medical history and be able to click delete for a medical they would like to remove. They will get a message to confirm the delete and if they click yes, this endpoint will be called. Their updated medical data will then be rendered on the page.

Example request:

```
curl -X DELETE http://logbookmanager.com/logbook/medical/4/
```

### Sample Success Response:

```

$ curl -i -X DELETE http://127.0.0.1:8000/medical/4/
% Total    % Received % Xferd  Average Speed   Time    Time     Time
0         0         0    0         0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
Date: Sun, 26 Mar 2023 14:51:54 GMT
Server: WSGIServer/0.2 CPython/3.10.10
Vary: Accept, Cookie
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
X-Frame-Options: DENY
Content-Length: 0
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin

```

### Sample Error Response:

 $\{$

```
"status" : 404,  
"response" : "That medical record was not found."  
}
```

## Reports

### 12. Get Reports

Method: GET

URL: <https://logbookmanager.com/logbook/reports/<report-type>>

Purpose: When the user is logged into the application, they will be able to click on the link for reports to go to the reports page. The user will select the type of report they want to view and click submit, and this endpoint will be called. A reports service will perform the business logic on the data and return the report information to the page.

Example requests:

```
curl --request GET --url https://logbookmanager.com/logbook/reports/hours
```

Sample Success Response:

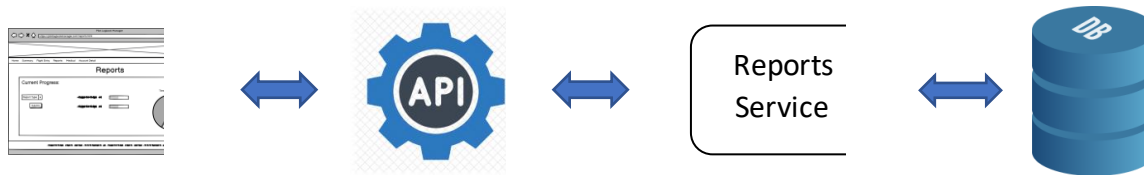
```
{  
  "status" : 200,  
  "data" : "[{"total_hours": 800, "night_hours": 251}]"  
}
```

Sample Error Response:

```
{  
  "status" : 404,
```

"response" : "Flight data must be entered to generate report."  
}

Reports Page:



## Pilot Certificates Held

### 13. Get Pilot Certificates Held

Method: GET

URL: <https://logbookmanager.com/logbook/pilot-certificates-held/>

Purpose: When the user is logged into the application, they will be able to click on the link for their account detail. On this page there will be a list of which certificates the user currently holds.

Example requests:

```
curl --request GET --url https://logbookmanager.com/logbook/pilot-certificates-held/
```

Sample Success Response:

```
$ curl -i http://127.0.0.1:8000/certificatesheld/ -w '\n'
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %         0         0             0      0 --:--:-- --:--:-- --:--:--   0
100  113  100  113    0    0    5127    0 --:--:-- --:--:-- --:--:--  5380HTTP/1.1 200 OK
Date: Sun, 26 Mar 2023 17:43:08 GMT
Server: WSGIServer/0.2 CPython/3.10.10
Content-Type: application/json
Vary: Accept, Cookie
Allow: GET, POST, HEAD, OPTIONS
X-Frame-Options: DENY
Content-Length: 113
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
[{"pilot":1,"certificate":1},{ "pilot":2,"certificate":1},{ "pilot":3,"certificate":3},{ "pilot":3,"certificate":3}]
```

Sample Error Response:

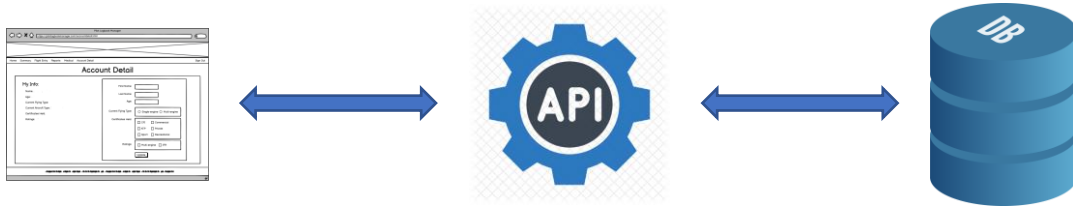
{

"status" : 404,

“response” : “No certificates entered for this account yet.”

}

Diagram:



14. Add a new pilot certificate:

Method: POST

URL: <https://logbookmanager.com/logbook/pilot-certificates-held/>

Purpose: When the user is logged into the application and has navigated to the account detail page, they will be able to click on update certificates and fill out a form to check off which certificates they hold, when they click submit this endpoint will be called. Their certificates data will then be rendered on the page.

Example requests:

```
curl -i https://logbookmanager.com/logbook/pilot-certificates-held/ \
```

```
-X POST \
```

```
-H 'Content-Type: application/json' \
```

```
-d '{"pilot": 1, "certificate": 3}' \
```

```
-w '\n'
```

Sample Success Response:



```

$ curl -i http://127.0.0.1:8000/certificatesheld/ \
-X POST \
-H 'Content-Type: application/json' \
-d '{"pilot": 1, "certificate": 3}' \
-w '\n'
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
100    58    100    27    100    31    1937    2224   --:--:-- --:--:-- --:--:--  4142
HTTP/1.1 201 Created
Date: Sun, 26 Mar 2023 17:53:21 GMT
Server: WSGIServer/0.2 CPython/3.10.10
Content-Type: application/json
Vary: Accept, Cookie
Allow: GET, POST, HEAD, OPTIONS
X-Frame-Options: DENY
Content-Length: 27
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
{"pilot":1,"certificate":3}

```

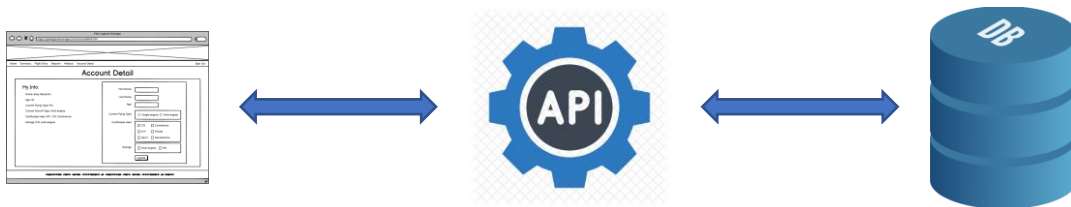
Sample Error Response:

```

{
  "status" : 400,
  "response" : "Missing required fields. Please fill out all information."
}

```

Diagram:



## 15.Delete Certificate Held

Method: DELETE

URL: <https://logbookmanager.com/logbook/pilot-certificates-held/>

<certificate-id>/

Purpose: When the user is logged into the application and has navigated to the account detail page, they will be able to click delete on a certificate. They will receive a message asking to confirm, and if they click yes, this endpoint will be called. Their updated certificates held data will then be rendered on the page.

Example Request:

```
curl -i -X DELETE http://logbookmanager.com/logbook/pilot-certificates-held/2/ \
```

Sample Success Response:

```
$ curl -i -X DELETE http://127.0.0.1:8000/certificatesheld/2/
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %         0         0             0             0             0             0
Date: Sun, 26 Mar 2023 18:02:17 GMT
Server: WSGIServer/0.2 CPython/3.10.10
Vary: Accept, Cookie
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
X-Frame-Options: DENY
Content-Length: 0
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
HTTP/1.1 204 No Content
```

Sample Error Response:

```
{
  "status": 404,
  "response": "The certificate record was not found."
}
```

Diagram:



## Pilot Ratings Held

### 16. Get Pilot Ratings Held

Method: GET

URL: <https://logbookmanager.com/logbook/pilot-ratings-held/>

Purpose: When the user is logged into the application, they will be able to click on the link for their account detail. On this page there will be a list of which type ratings the user currently holds.

Example requests:

```
curl --request GET --url https://logbookmanager.com/logbook/pilot-ratings-held/
```

Sample Success Response:

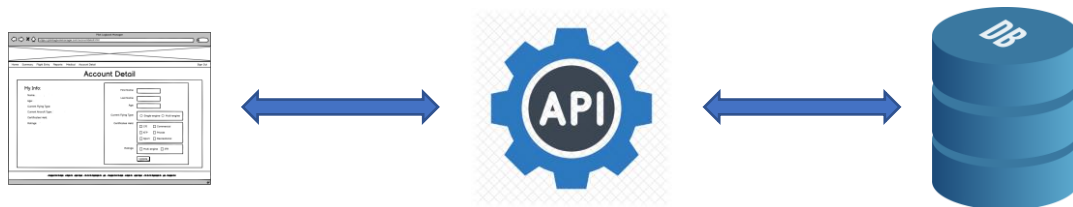
```
$ curl -i http://127.0.0.1:8000/ratingsheld/ -w '\n'
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100    47    100    47    0    0    7264    0 --:--:-- --:--:-- --:--:--  9400HTTP/1.1 200 OK
Date: Sun, 26 Mar 2023 18:08:45 GMT
Server: WSGIServer/0.2 CPython/3.10.10
Content-Type: application/json
Vary: Accept, Cookie
Allow: GET, POST, HEAD, OPTIONS
X-Frame-Options: DENY
Content-Length: 47
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin

[{"pilot":1,"rating":1},{ "pilot":1,"rating":2}]
```

Sample Error Response:

```
{
  "status" : 404,
  "response" : "No ratings entered for this account yet."
}
```

Diagram:



17. Add a new rating:

Method: POST

URL: <https://logbookmanager.com/logbook/ratings-held/>

Purpose: When the user is logged into the application and has navigated to the account detail page, they will be able click on update ratings and fill out a

form to check off which ratings they hold, when they click submit this endpoint will be called. Their ratings data will then be rendered on the page.

Example requests:

```
curl -i https://logbookmanager.com/logbook/ratings-held/ \
```

```
-X POST \
```

```
-H 'Content-Type: application/json' \
```

```
-d '{"pilot": 1, "rating": 3}' \
```

```
-w '\n'
```

Sample Success Response:

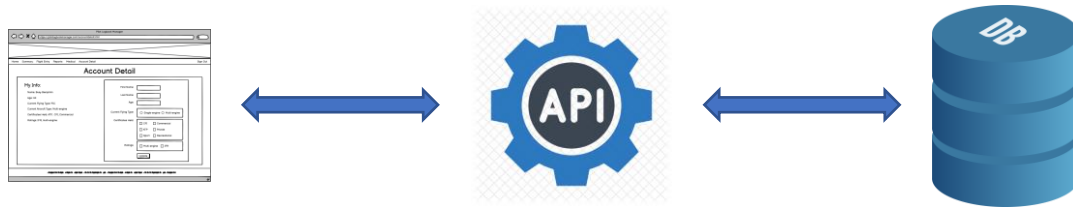
```
$ curl -i http://127.0.0.1:8000/ratingsheld/ \
-X POST \
-H 'Content-Type: application/json' \
-d '{"pilot": 2, "rating": 2}' \
-w '\n'
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
100  48 100    22 100    26    1419    1678  --:--:-- --:--:-- --:--:--   3200HTTP/1.1 201 Created
Date: Sun, 26 Mar 2023 18:12:39 GMT
Server: WSGIServer/0.2 CPython/3.10.10
Content-Type: application/json
Vary: Accept, Cookie
Allow: GET, POST, HEAD, OPTIONS
X-Frame-Options: DENY
Content-Length: 22
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin

{"pilot":2,"rating":2}
```

Sample Error Response:

```
{
  "status": 400,
  "response": "Missing required fields. Please fill out all information."
}
```

Diagram:



## 18.Delete Rating Held

Method: DELETE

URL: <https://logbookmanager.com/logbook/ratings-held/>

<rating-id>/

Purpose: When the user is logged into the application and has navigated to the account detail page, they will be able to click delete on a certificate. They will receive a message asking to confirm, and if they click yes, this endpoint will be called. Their updated certificates held data will then be rendered on the page.

Example Request:

```
curl -i -X DELETE http://logbookmanager.com/logbook/ratings-held/2/ \
```

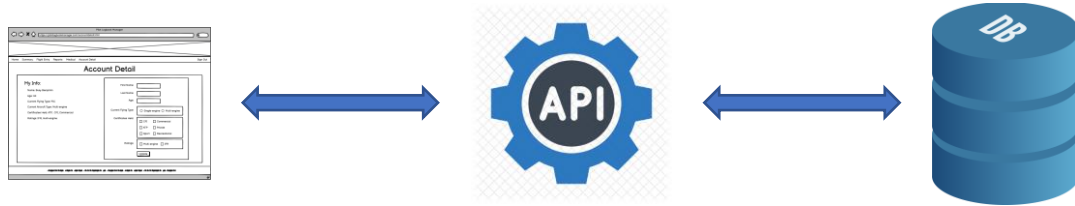
Sample Success Response:

```
$ curl -i -X DELETE http://127.0.0.1:8000/ratingsheld/2/
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %         0         0             0      0      0      0      0
0   0   0    0     0     0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
Date: Sun, 26 Mar 2023 18:14:41 GMT
Server: WSGIServer/0.2 CPython/3.10.10
Vary: Accept, Cookie
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
X-Frame-Options: DENY
Content-Length: 0
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
HTTP/1.1 204 No Content
```

Sample Error Response:

```
{
  "status": 404,
  "response": "The rating record was not found."
}
```

Diagram:



## Users

19. I will be using the Django auth app to authenticate users. The endpoints for login, register, and logout should be:

Method: POST

URL: <https://logbookmanager.com/logbook/users/login/>

Purpose: When a user wants to login from the home page, they will click on the link to sign in and will be taken to this endpoint. They can enter their user login information and click submit. They will be redirected to the user home page.

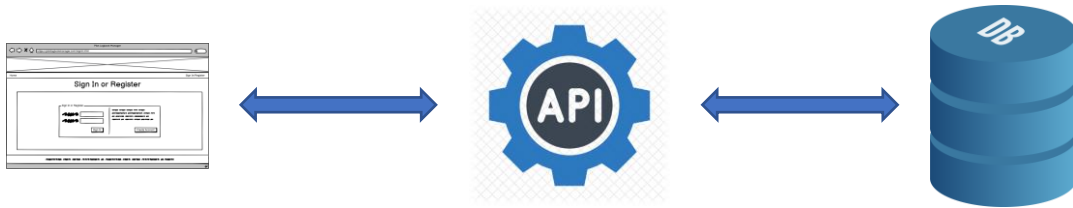
Diagram:



20. URL: <https://logbookmanager.com/logbook/users/signup/>

Purpose: When a user needs to create an account, they will click on the link to sign in/create account and be taken to the signin page. From there if they click on create account, they will fill out their account details and click submit, then this endpoint will be called.

Diagram:



21. URL: <https://logbookmanager.com/logbook/users/logout/>

Purpose: When a user wants to logout from the application, they will click on the logout link. They will be signed out and will be redirected to the home page.

Diagram:

