OPL Performance Calculator for a L-39ZA Light Attack Aircraft

Jack Brubaker, Peyton Hobson, Alex Walther

Background

Operator Performance Laboratory (OPL)

- Part of the Iowa Technology Institute (ITI)
- Flight test laboratory specializing in:
  - Human-in-the-loop systems
  - Intelligent autonomous systems
- Projects center around Public Aircraft Operations (PAO) of OPL aircraft as well as their associated simulators

L-39ZA Light Attack Aircraft

- Manufactured in Czechoslovakia and designed to handle armed training and light attack missions
- Equipped with four wing pylons and a center mount that can be used for armament/noncombatant arrangements
- L-39ZA has a maximum flight weight of about 5580 kg (12,300 lbs)
- Performance Parameters:
  - Takeoff Speed and Distance, Accel-Stop Distance, Landing Reference Speed and Distance, Ground Run, Crosswind Component

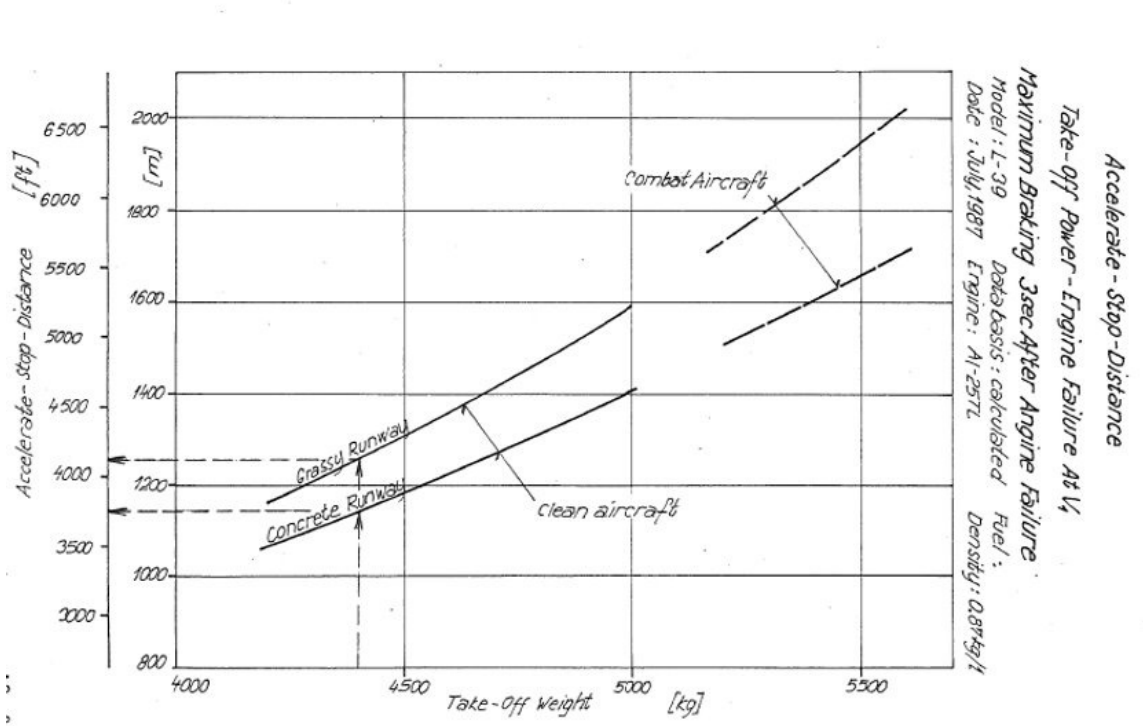
Problem Statement

Problem – Hand-held graphical lookup charts used in calculating the required performance parameters have several drawbacks:

1. Traditionally use Metric Units
2. Relatively Inaccurate
3. Tedious and Inaccessible
4. Prone to Human Error

Solution – A web-hosted, electronic performance calculator capable of taking in aircraft and environmental characteristics and producing their respective performance parameters will help to:

1. Improve Accuracy
2. Decrease Calculation Time
3. Reduce Human Error
4. Increase Accessibility for all OPL Members



Project Objectives

Primary Objective

- Goal: Streamline performance parameter calculation(s) by reducing interpretation time and calculation error while increasing calculation accuracy

Secondary Objective

- Goal: Host the performance calculator on an easily accessible web interface, whose design is readable, concise, and encourages usability

Project Phases

1. Analysis & Preliminary Work

2. Detail Design

3. Implementation

4. Test & Validation

Analysis & Preliminary Work

Capture Client Requirements

- Initial client meeting was used to introduce the team
- Introductory meeting helped to:
  - Define the problem and desired solution
  - Establish information resources
  - Determine project expectations

Capture Engineering Requirements

- Evaluated potential methods to digitize the graphical lookup curves utilized in the algorithm's calculations
- Determined the primary programming languages/frameworks used in the application's development
  - Python and Java – Back-End Functionality
  - Angular – Front-End Functionality

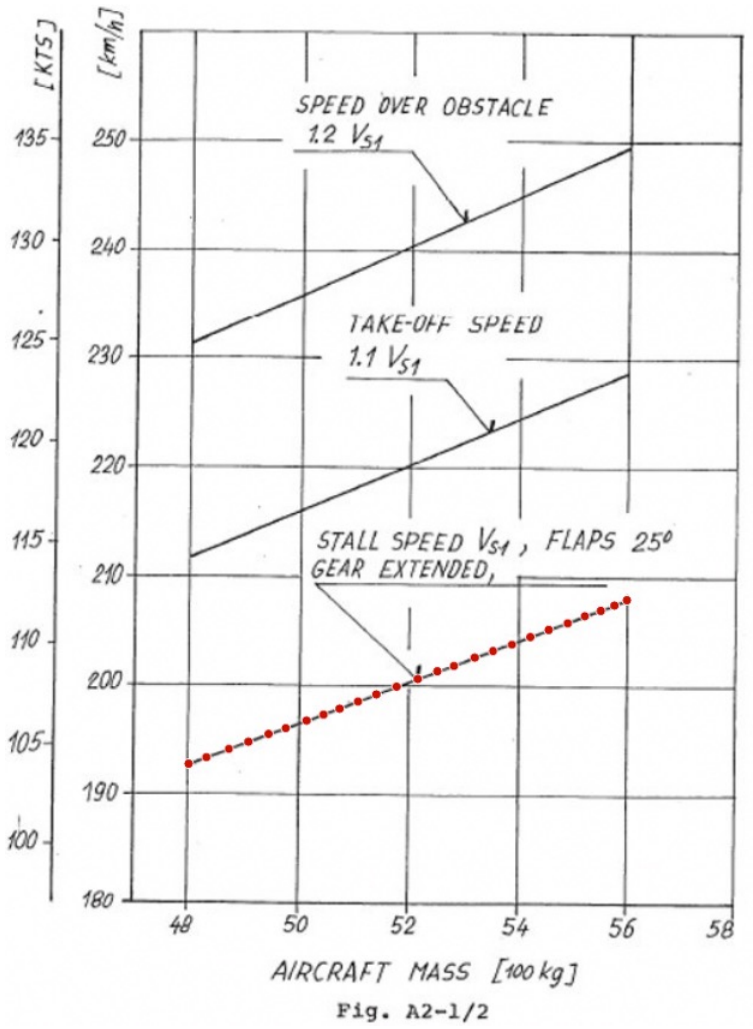


Fig. A2-1/2

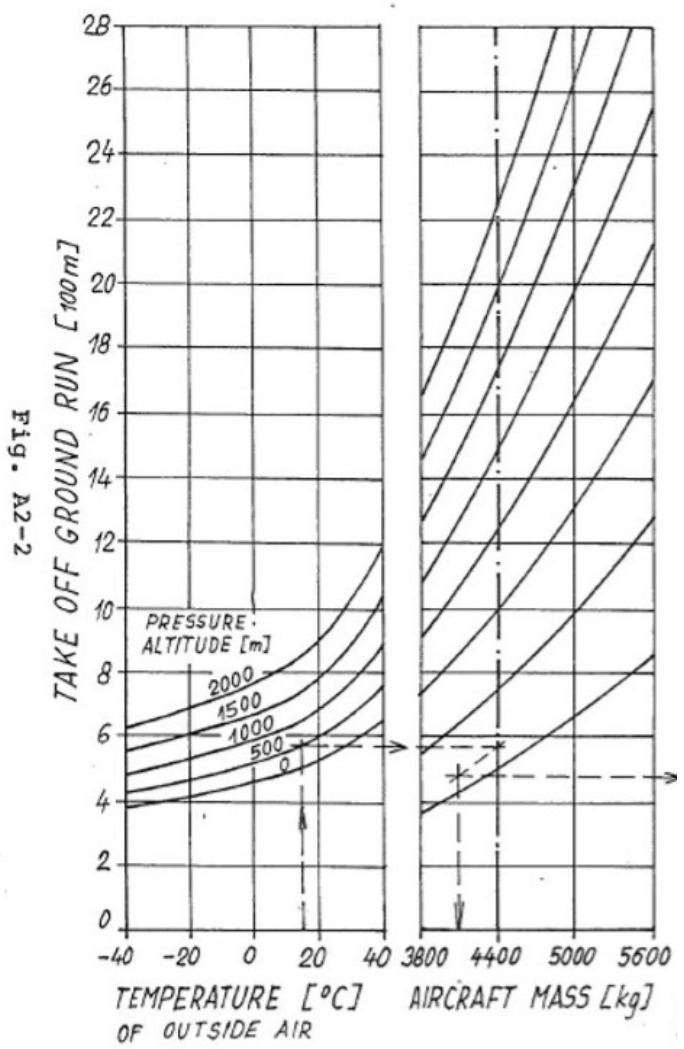
Analysis & Preliminary Work

Design Overall Architecture & Functions

- Plot tracing and polynomial interpolation were selected to calculate the necessary performance parameters due to its efficiency
- Token authentication, activity tracking, and secure login functionalities were also developed to uphold OPL's request for a private application

Describe Concept of Operations (CONOPS)

- User interviews with members of the OPL were used to better understand the process in which mission configurations were chosen
  - (Ex: a plane loadout in the cold winter might not be able to safely fly in the summer)



Detail Design

Create & Design User Interface

- Designed UI mock-ups that helped to form the basis for finalized performance calculator UI

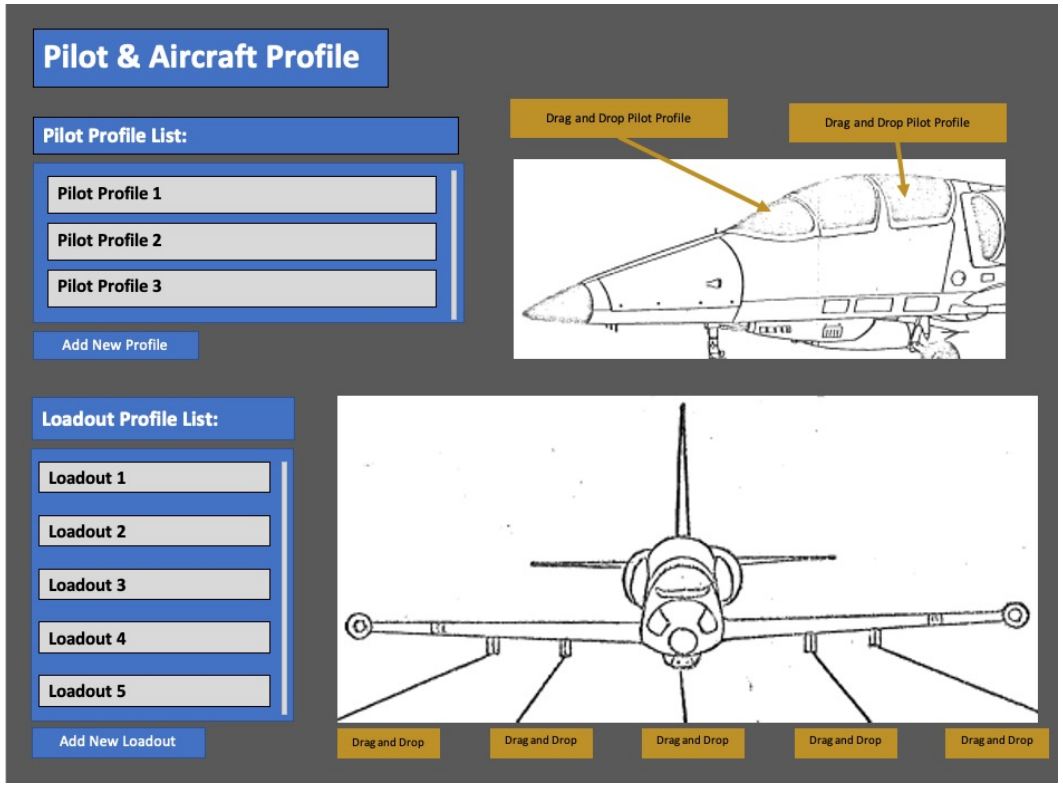
Design Data Storage & Framework

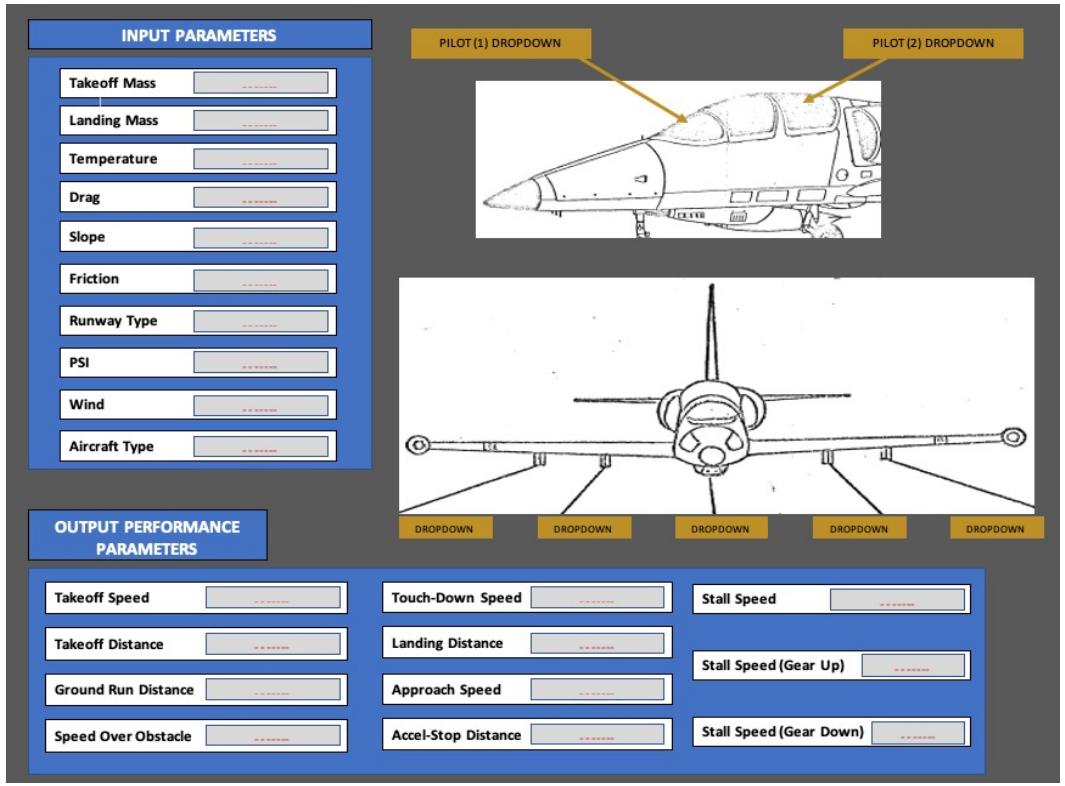
- Java used extensively to provide the back-end with its functionality
- The Angular Framework, CSS, and HTML are used to produce the UI display
  - Angular uses GET and POST requests to retrieve information from the server before processing and displaying it

Design Functions

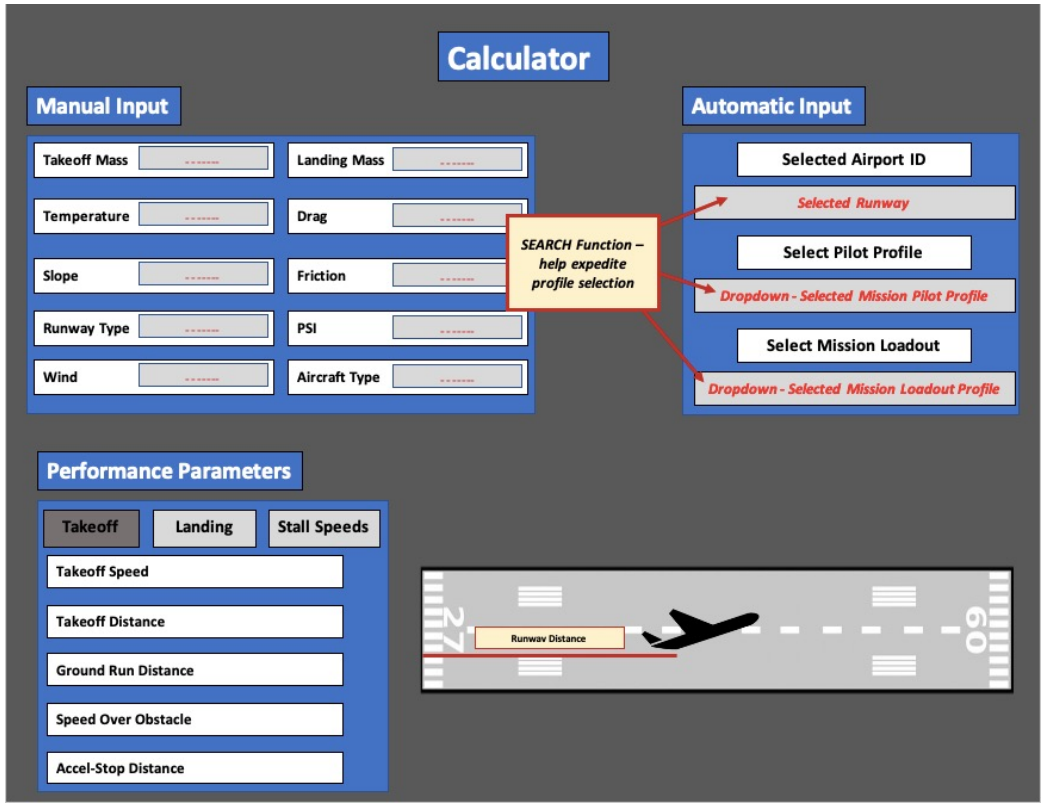
- Developed functions to create and authenticate new and existing users using tokens
- Pre-saved profiles for pilots and aircrafts were also created
- Print Functionality for the outputted parameters was also included

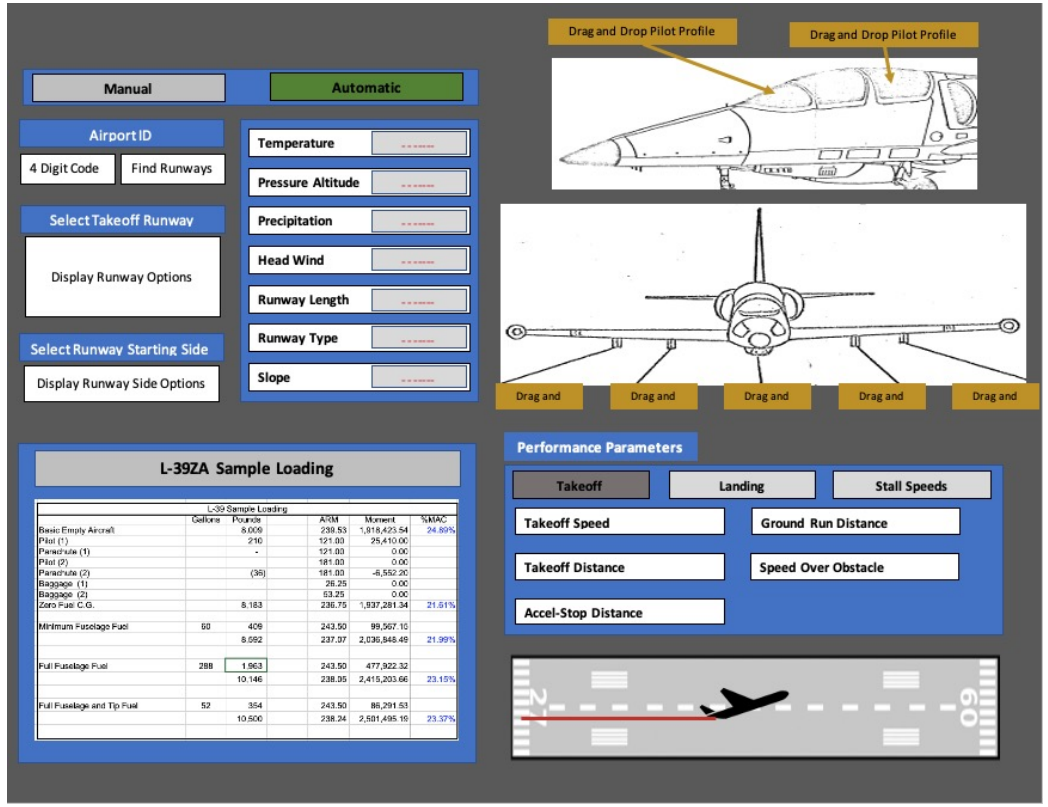
Detail Design – UI Mockups





Detail Design – UI Mockups





Implementation

Implementation

1. Build Out User Functionality

- Since all design decisions had been made, team could now start building out functionality

2.Module Level Test

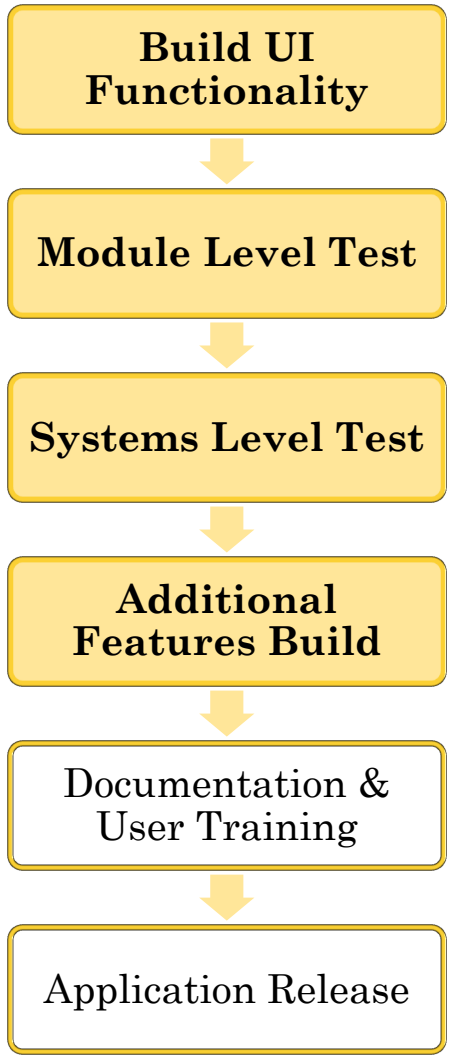
- Test cases were used to compare calculator outputs to those of known graphical lookup chart results
- Ensured back-end functionality was operating correcting by uncovering bugs and inconsistencies in the calculator's algorithm

3. Systems Level Test

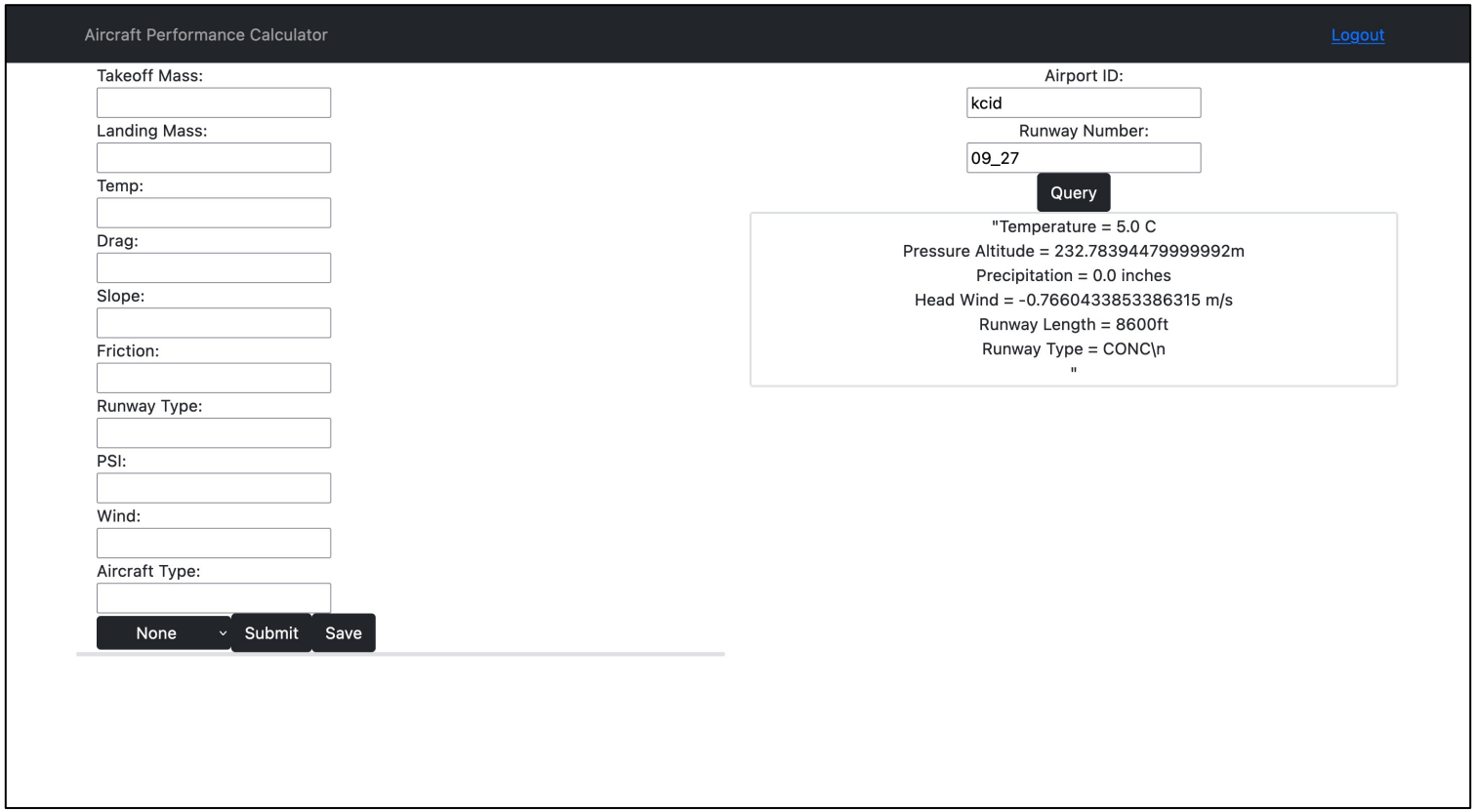
- RESTful API used to connect the back-end with the front-end
- Confirmation tests using HTTP requests were used to verify front-end and back-end connection was working appropriately

4. Inclusion Build with Additional Features

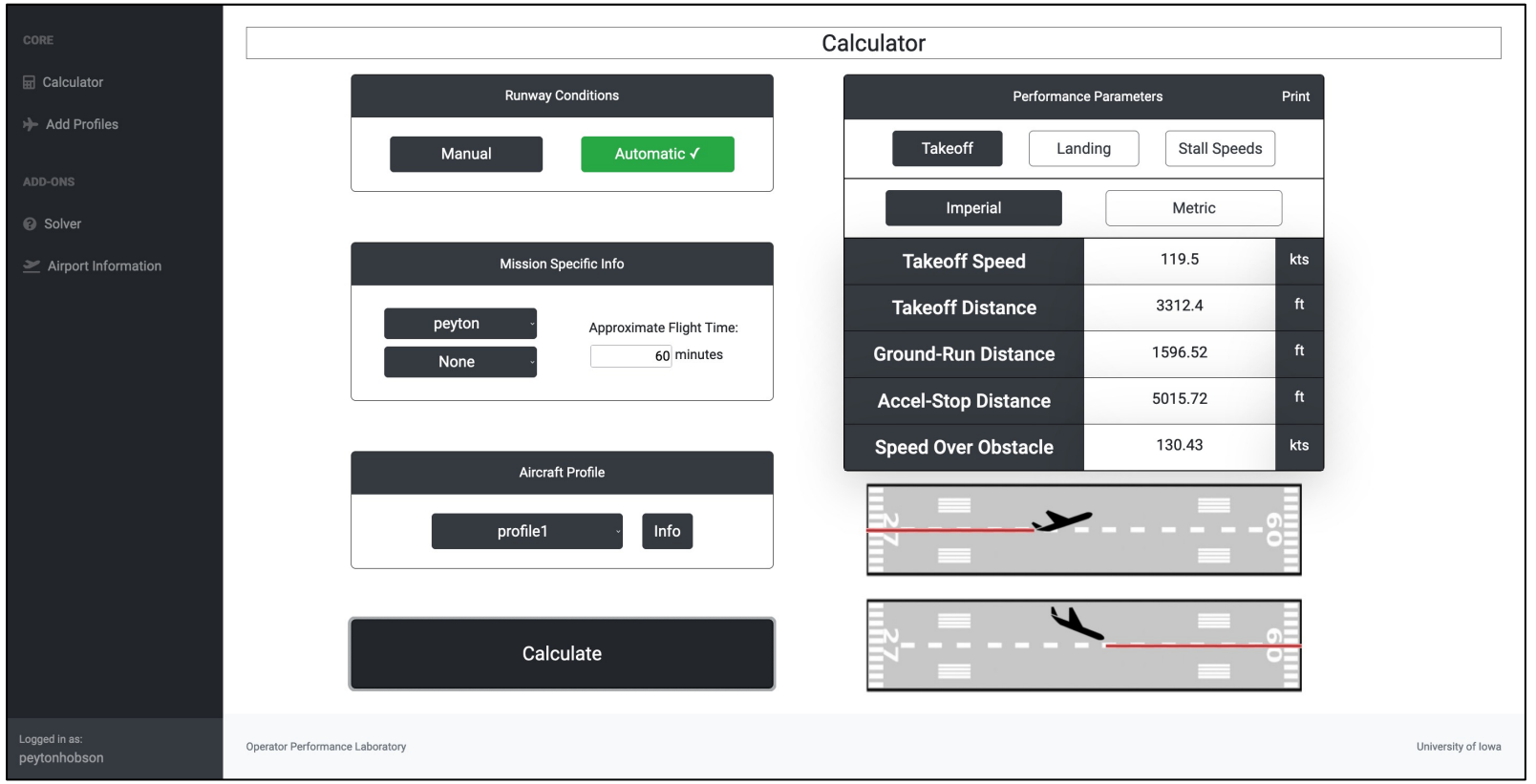
- Features Added:
  - Pilot Profiles
  - Common Aircraft Store Loadouts
  - Performance Parameter Print Functionality
  - Automatic Entry of Airport Weather Conditions
  - Visual Display for Takeoff and Landing Buffers



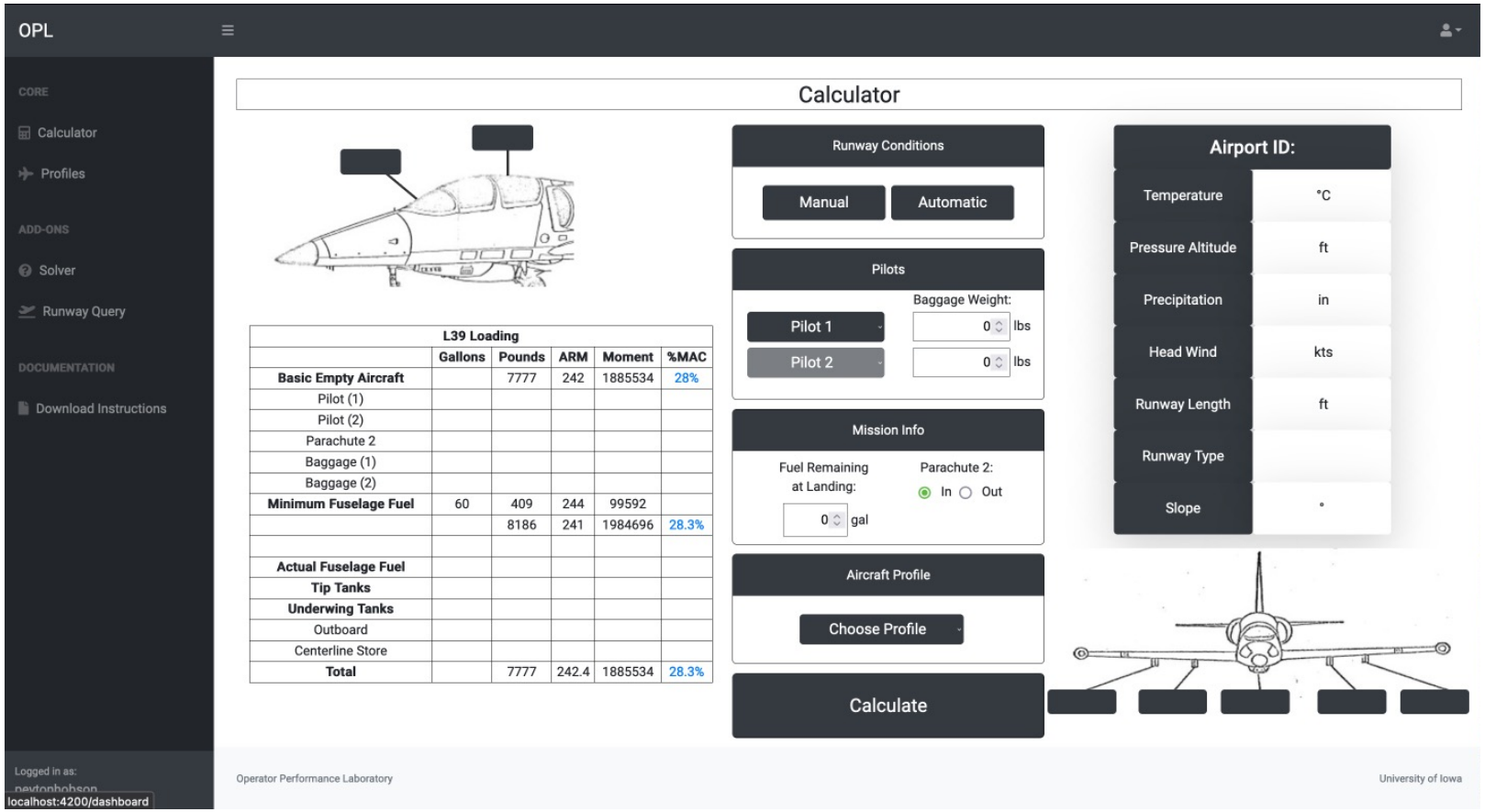
Detail Design – Original UI



Detail Design – First Full UI

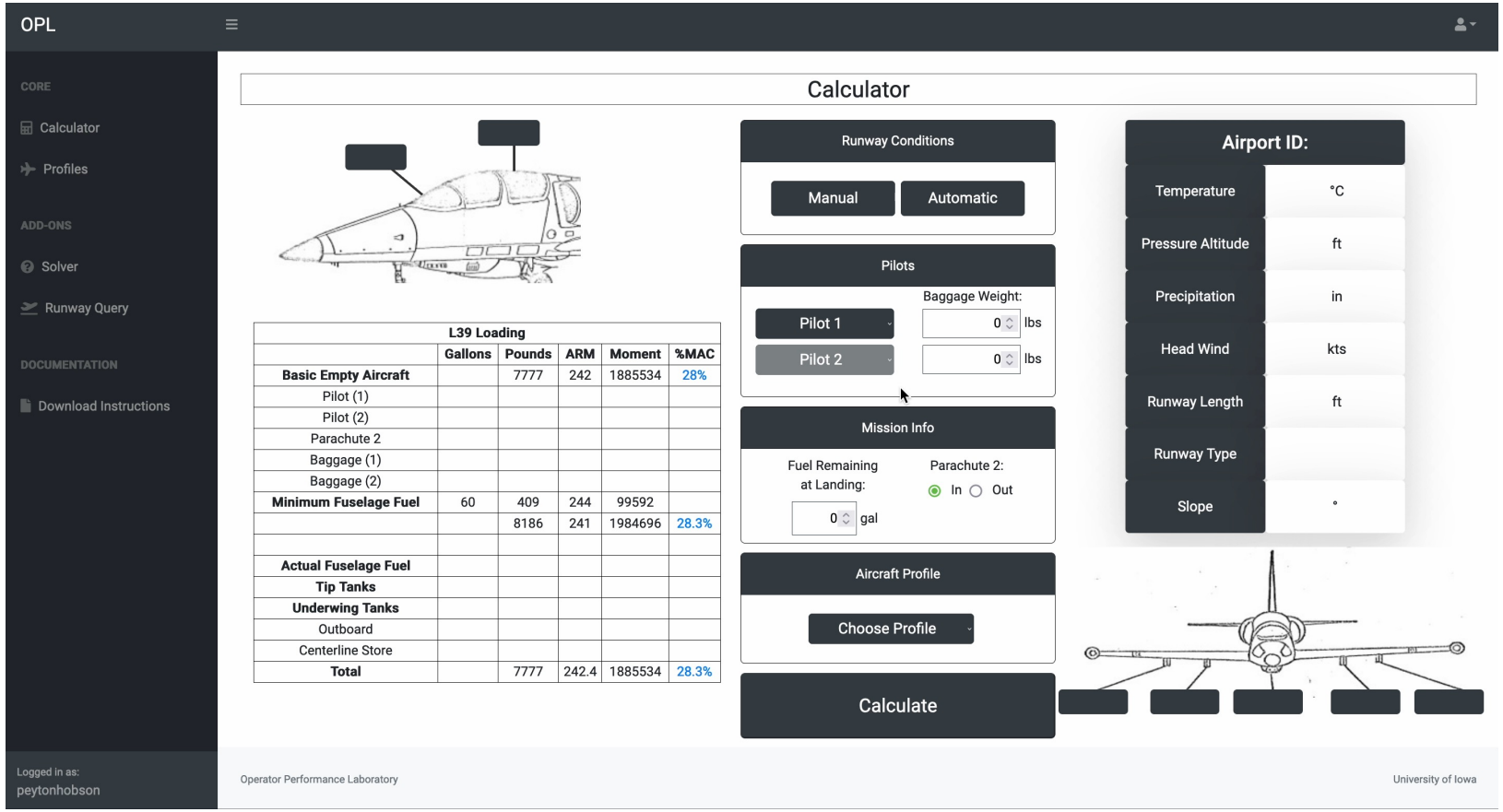


Detail Design – Final UI



L39 Loading					
	Gallons	Pounds	ARM	Moment	%MAC
Basic Empty Aircraft		7777	242	1885534	28%
Pilot (1)					
Pilot (2)					
Parachute 2					
Baggage (1)					
Baggage (2)					
Minimum Fuselage Fuel	60	409	244	99592	
		8186	241	1984696	28.3%
Actual Fuselage Fuel					
Tip Tanks					
Underwing Tanks					
Outboard					
Centerline Store					
Total		7777	242.4	1885534	28.3%

Detail Design – Demo



L39 Loading					
	Gallons	Pounds	ARM	Moment	%MAC
Basic Empty Aircraft		7777	242	1885534	28%
Pilot (1)					
Pilot (2)					
Parachute 2					
Baggage (1)					
Baggage (2)					
Minimum Fuselage Fuel	60	409	244	99592	
		8186	241	1984696	28.3%
Actual Fuselage Fuel					
Tip Tanks					
Underwing Tanks					
Outboard					
Centerline Store					
Total		7777	242.4	1885534	28.3%

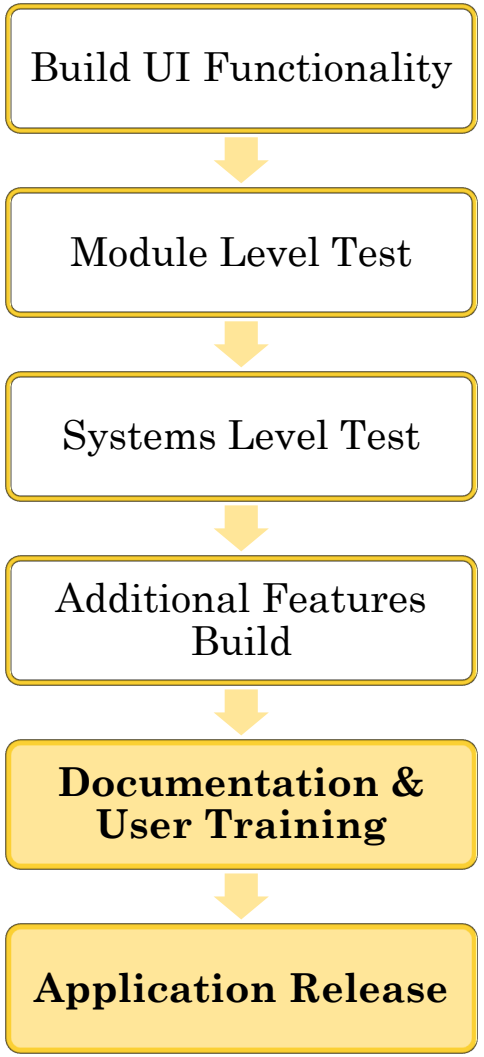
Implementation

5. Develop Documentation and User Training

- Due to the complexity of the application our team developed a User Training PowerPoint that includes:
  - Comprehensive overview of the calculator's full functionality
  - Written instructions detailing how to operate the calculator
  - Picture displays and embedded video demonstrations depicting how to enter required inputs and interpret outputted performance parameters
- PowerPoint was selected due to its ability to utilize all these media platforms to provide a concise yet complete summary of the calculator's features

6. Application Release

- Performance calculator released in three parts:
  1. Database server
  2. Back-end service
  3. Front-end service
- The completed application and its associated resources have been turned over to the OPL for future maintenance and modification



Test & Validation

- Initial validation done using OPL's Flight Simulator
  - The team utilized the X-Plane simulator to compare the simulator's performance parameters to the calculator's outputs
  - Most performance parameters were within ±3 Knots, however takeoff and landing parameters saw larger differences
- Possible explanations for the larger difference in Takeoff and Landing parameters
  - X-Plane tends to misrepresent the runway friction which would account for the discrepancy in parameters pertaining to the runway while not impeding air parameters such as stall speeds
  - The pilot of the simulator was not a licensed pilot and therefore could also account for the difference in parameter outputs

Test & Validation

- Real-world confirmatory flights are still looking to be conducted to test the fully-functioning performance calculator
  - Dr. Thomas Schnell would lead and pilot any and all confirmatory flights
- Our team is working with OPL's schedule to determine if confirmatory flights can be completed prior to the semester's end
  - In the case the team cannot find a suitable time with OPL's schedule, real-world confirmatory flights can still be conducted after the project's completion

Resources & Constraints

- **Resources**
  - Colton Thompson – Project Liaison
  - Dr. Thomas Schnell – Project Liaison
  - Jon Stoltz – Airport Querying Resource
  - Brad Parker – OPL Web-Hosted Server(s) Resource

- **Constraints**
  - Limited physical constraints due to software-based nature of the project

    Time Constraints – Roughly 15 weeks to develop application and integrate it with a functioning user interface

Colton Thompson

Dr. Thomas Schnell

Questions?