Comfort Airlines
Operations Manual

Index

1 Introduction

Welcome to Comfort Airlines, the pinnacle of realism and excitement in airline simulation, where we offer an immersive journey into the intricacies of airline operation.

This Comfort Airlines simulation is developed and maintained by Team Foobar, an up-and-coming provider of simulation software for the aviation industry. Team Foobar is dedicated to delivering realistic and immersive simulation experiences tailored to the needs of airline professionals and enthusiasts alike.

As the operator of this Comfort Airlines simulation, Team Foobar is committed to providing continuous support, updates, and enhancements to ensure that users have access to the most advanced and realistic simulation environment possible.

Whether you're an airline CEO, a flight operations manager, or an aviation enthusiast, this simulation offers an unparalleled platform to explore the complexities of airline company management in a dynamic and engaging manner.

We at Team Foobar are thrilled to have you on board and look forward to accompanying you on your journey through the virtual skies with Comfort Airlines.

This manual is a walkthrough on how to use Team Foobar's Comfort Airlines simulation. It may also provide additional background information on it. From managing flight schedules to handling customer satisfaction, our goal is to ensure that you have all the tools and knowledge necessary to embark on your trip.

In this manual, you'll find detailed instructions on every aspect of our Comfort Airlines simulation, including:

1. **Getting Started:** Learn how to set up your simulator, configure your aircraft, and customize your experience to suit your preferences.
2. **Navigation:** Explore the vast world that Comfort Airlines offers and learn how to navigate it using a variety of tools.
3. **Flight Planning:** Plan your routes, file flight plans, and optimize your journey for efficiency and safety.
4. **Continuously Improve**: Utilize performance analytics and feedback mechanisms to identify areas for improvement, implement corrective measures, and strive for excellence in every aspect of Comfort Airline's operations.

Needless to say, Comfort Airlines offers an unparalleled experience that will keep you coming back for more.

Throughout this manual, you will encounter various terms and abbreviations specific to the airline industry and Comfort Airlines. To facilitate understanding, a glossary of key terms and a list of commonly used abbreviations are provided in the appendix.

If at any point the output you receive is different from what is outlined in this manual, see the FAQ at the bottom of this document. If those solutions still do not help your problem, please submit a support ticket so that we may further assist you.

As always, thank you for choosing Comfort Airlines!

2 Revision Status

This user manual for Team Foobar's Comfort Airlines Simulation is subject to periodic updates and revisions to ensure accuracy and relevance. The revision status of this manual is indicated by the version number and the date of the latest revision.

Amendment Process:

1. **Identification of Issues**: Users are encouraged to report any errors, inconsistencies, or suggestions for improvement regarding this manual to Team Foobar.
2. **Review and Analysis**: Reported issues are reviewed and analyzed by the Team Foobar team to determine their validity and potential impact on the manual's content.
3. **Revision and Update**: Upon validation, necessary revisions and updates are made to the manual to address reported issues and ensure accuracy and clarity.
4. **Verification and Approval**: Revised versions of the manual undergo verification and approval processes to confirm compliance with quality standards before publication.

Revision Status:

- **Version 1.0 (Effective Date: 4.14.2024)**: Initial release of the Team Foobar Comfort Airlines User Manual.

Users are advised to refer to the revision status page to ensure that they are consulting the most up-to-date version of the manual for accurate information.

3 Duties and Responsibilities of Management and Operational Personnel

1. Chief Executive Officer (CEO):

- Overall responsibility for the strategic direction and management of the airline company.
- Ensures compliance with regulatory requirements and industry standards.
- Oversees financial performance and budgetary decisions.
- Implements policies and procedures to enhance operational efficiency and safety.

2. Chief Operations Officer (COO):

- Responsible for the day-to-day operations of the airline.
- Oversees flight scheduling, crew management, and aircraft maintenance activities.
- Ensures compliance with operational regulations and safety standards.
- Manages operational resources and coordinates with various departments to achieve operational objectives.

3. Chief Pilot:

- Ensures that flight operations are conducted safely and in accordance with regulatory requirements.
- Provides guidance and support to flight crews regarding operational procedures and safety protocols.
- Collaborates with the Chief Operations Officer on fleet management and route planning.

4. Director of Maintenance:

- Responsible for the airworthiness and maintenance of the airline's fleet.
- Oversees maintenance scheduling, inspections, and repairs.
- Ensures compliance with maintenance regulations and safety standards.
- Manages maintenance personnel and facilities to ensure efficient operations.

5. Emergency Response Coordinator:

- Develops and implements emergency response plans (ERP) to address various crisis scenarios.

- Coordinates with relevant stakeholders, including regulatory agencies and emergency services, to ensure a timely and effective response to emergencies.
- Maintains communication systems and protocols for emergency coordination.

Emergency Response Plan (ERP):

In the event of an emergency, personnel are required to follow the established Emergency Response Plan (ERP) to ensure the safety and well-being of all stakeholders. The ERP includes:

- Clear procedures for reporting emergencies to designated authorities.
- Evacuation protocols for passengers and crew, including designated assembly points.
- Communication channels for coordinating emergency response efforts.
- Assignments of roles and responsibilities for key personnel during emergencies.
- Training and drills to familiarize personnel with emergency procedures and protocols.

4 Description of Safety Management System

The safety management system (SMS) of our airline company is a comprehensive framework designed to proactively identify, assess, and mitigate risks associated with our operations while ensuring compliance with relevant regulatory requirements. Our SMS is structured to encompass the following key components:

1. Safety Policy and Objectives:

- **Management Commitment and Responsibility:** Our management is committed to ensuring the highest standards of safety in all aspects of our operations. They provide leadership and resources to support the implementation of the SMS.
- **Safety Accountabilities:** Clear roles and responsibilities are defined for personnel at all levels to ensure accountability for safety-related tasks and decisions.
- **Appointment of Key Staff Members:** Key personnel are appointed to oversee the various components of the SMS, including the Safety Officer, Quality Manager, and Emergency Response Coordinator.

- **Emergency Response Planning:** We have established comprehensive emergency response plans (ERP) to effectively manage and respond to various crisis scenarios, ensuring the safety and well-being of passengers, crew, and assets.
- **SMS Documentation:** All aspects of our SMS, including policies, procedures, and risk assessments, are documented and regularly reviewed to ensure accuracy and relevance.

2. Safety Risk Management:

- **Hazard Identification:** We systematically identify and assess hazards associated with our operations through proactive measures such as safety audits, incident reporting, and risk assessments.
- **Risk Assessment and Mitigation:** Identified hazards are analyzed to determine their potential impact and likelihood, and appropriate mitigation measures are implemented to reduce or eliminate associated risks.

3. Safety Assurance:

- **Safety Performance Monitoring and Measurement:** We continuously monitor and measure our safety performance through key performance indicators, safety audits, and regular reviews to identify trends, areas for improvement, and compliance with regulatory requirements.
- **Management of Change:** Any changes to our operations, procedures, or equipment are carefully evaluated for potential safety implications, and appropriate measures are implemented to manage associated risks.
- **Continuous Improvement:** We are committed to ongoing improvement of our SMS through regular review, feedback mechanisms, and implementation of best practices to enhance safety performance.

4. Safety Promotion:

- **Safety Communication:** Open and transparent communication channels are established to facilitate the reporting of safety concerns, sharing of safety-related information, and promoting a culture of safety throughout the organization.

5 General/Basic

5.1 Getting Started

In order to run the simulation, you must have **Python 3.10** installed. Several Python packages are also required, detailed below. If you want to use the GUI, you will also need a web server, such as XAMPP or something equivalent.

Python Package Name	Version
haversine	2.8.0
pytest	7.4.4
structlog	24.1.0
sqlalchemy	2.0.25
pywebview	latest

These packages can be individually installed using Python's package manager Pip. To install a package, simply run the following in a terminal window:

```
$ pip3 install <package name> <version>
```

Or, you can install all of them at once after downloading the simulation using:

```
$ pip3 install -r github/requirements.txt
```

You can download the simulation in one of two ways. Both will be outlined.

1. Using git commands in a terminal window, or
2. Manually download as a .zip directly from Github.

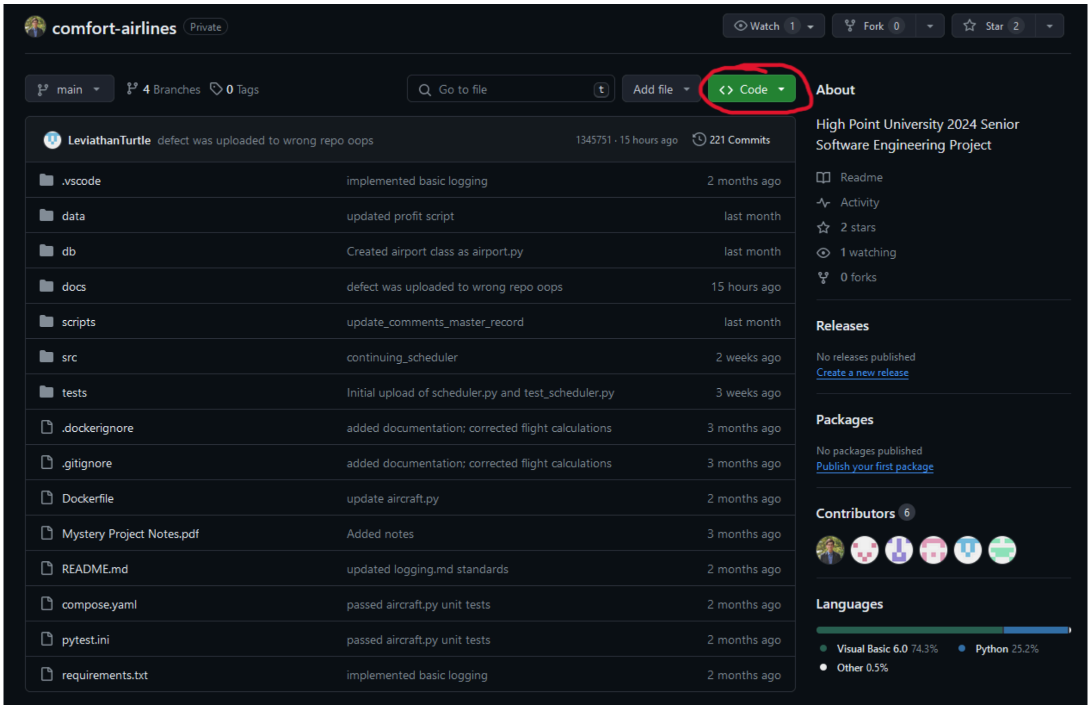
5.2 Downloading the Simulation

Using Git, you can download the simulation by running the following in a terminal window:

```
$ git clone https://github.com/EthanC2/comfort-airlines
```

That is the easiest way, but if you do not have Git, you can also manually download it.

1. Navigate to https://github.com/EthanC2/comfort-airlines.
2. Click the green `<> Code` button.



LeviathanTurtle defect was uploaded to wrong repo oops 1345751 · 15 hours ago 221 Commits

.vscode	implemented basic logging	2 months ago
data	updated profit script	last month
db	Created airport class as airport.py	last month
docs	defect was uploaded to wrong repo oops	15 hours ago
scripts	update_comments_master_record	last month
src	continuing_scheduler	2 weeks ago
tests	Initial upload of scheduler.py and test_scheduler.py	3 weeks ago
.dockerignore	added documentation; corrected flight calculations	3 months ago
.gitignore	added documentation; corrected flight calculations	3 months ago
Dockerfile	update aircraft.py	2 months ago
Mystery Project Notes.pdf	Added notes	3 months ago
README.md	updated logging.md standards	2 months ago
compose.yaml	passed aircraft.py unit tests	2 months ago
pytest.ini	passed aircraft.py unit tests	2 months ago
requirements.txt	implemented basic logging	2 months ago

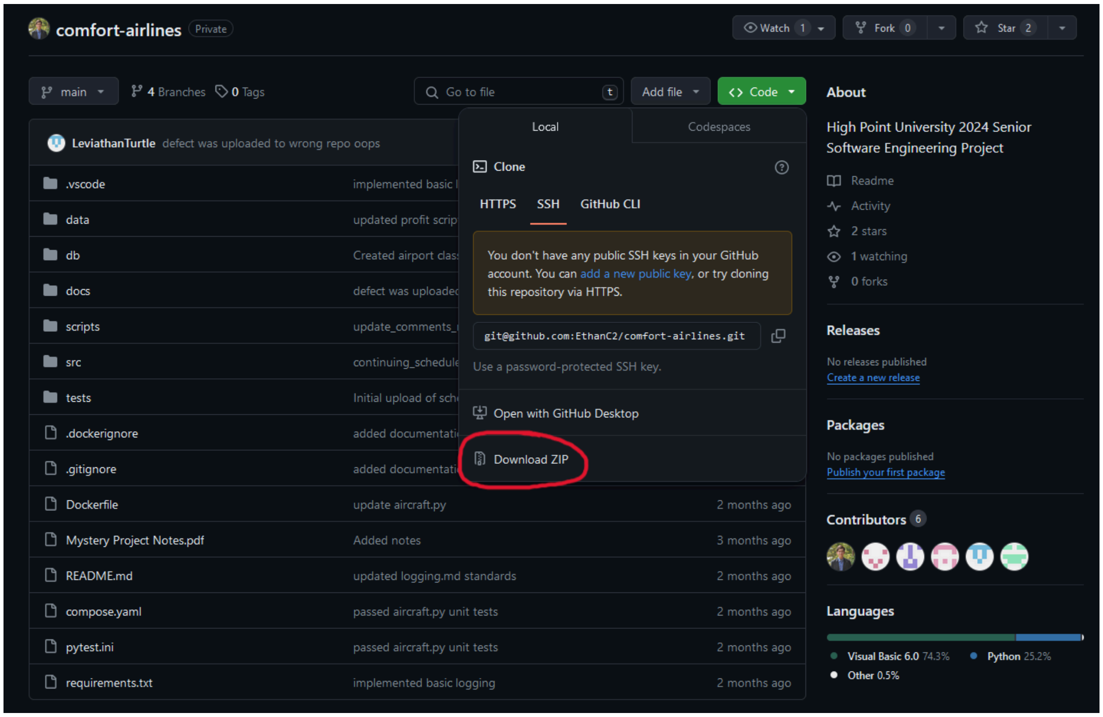
High Point University 2024 Senior Software Engineering Project

No releases published
Create a new release

No packages published
Publish your first package

Visual Basic 6.0 74.3% · Python 25.2% · Other 0.5%

3. Select `Download ZIP`.



You don't have any public SSH keys in your GitHub account. You can add a new public key, or try cloning this repository via HTTPS.

```
git@github.com:EthanC2/comfort-airlines.git
```

Use a password-protected SSH key.

LeviathanTurtle defect was uploaded to wrong repo oops

High Point University 2024 Senior Software Engineering Project

No releases published
Create a new release

No packages published
Publish your first package

Visual Basic 6.0 74.3% · Python 25.2% · Other 0.5%

4. Extract the archive to a directory (folder) of your choosing.

5.3 Simulation File Structure

The simulation has both a GUI element and the actual simulation. The top-level directory is all things pertaining to the GUI. `webdriver.py`, used to run the GUI, and the subdirectory `github` are the only important things of note. The subdirectory `github` is the simulation itself. It consists of the following:

File / Folder	Description
data	Contains static data related to the project, such as airport names, locations, and coordinates.
docs	Contains project and module documentation, including coding conventions and standards.
scripts	Contains one-time scripts, mostly for data analysis. these are not a part of the main application.
src	Contains the source code for the project, written in Python 3.10. **Ensure all requirements are installed first.**
tests	Contains unit, integration, and system tests that use the Python pytest testing framework.
.dockerignore	Contains a list of files for docker to ignore when copying applications to the application container.
.gitignore	Contains a list of files for Git to ignore (e.g when pushing or pulling code).
compose.yaml	Contains configurations for docker services (application and database containers).
Dockerfile	Contains docker commands for initializing the application container.
README.md	Contains a list of descriptions of all major files and folders.
requirements.txt	Contains all the packages and their versions.
run.ps1	Script to run everything in the simulation, including GUI.

Various directories may have an `archive` folder. These contain older versions of code or data and serve no purpose when running the simulation. They are for internal development purposes only and should not be considered a part of the project.

5.4 Running the Simulation

The simulation can be started by executing the following in a terminal window:

```
$ python3 github/src/main.py
```

That's it! The window that the simulation is running in will show the log entries as they are generated. Each entry contains the following:

- Aircraft tail number → tail number unique to an aircraft
- Source airport → airport the plane departed from
- Destination airport → airport the plane is flying to
- Event → what object is being logged
- Source file → the file that generated this log entry
- Source function → the function that generated this log entry
- Source line → the line where this log entry was generated
- Log ID → unique identifier for that log entry
- Simulation time (s) → how long the simulation has been running
- Level → type of log entry
- Real time → when this log entry was made in real time

These log files are generated in the `reports` directory. Everything pertaining to the aircrafts is stored in `aircrafts.csv`, everything pertaining to the airports is stored in `airports.csv`, and so on. The ledger is a record of every expense and profit made during the course of the simulation.

In total, there are four data files generated as output in CSV format, detailed below.

File Name	Structure
passenger	Unique passenger ID, location, source airport, destination airport, expected departure time, expected arrival time, actual departure time, actual arrival time, flights taken
aircraft	Tail number, flight number, source airport, destination airport, departure time, arrival time, number of passengers
ledger	Item, net profit, time, location

Our simulation also comes with a GUI element. To use it, first run:

```
$ powershell github/scripts/pipeline.ps1
```

Once this is finished, you can then run:

```
$ python3 webdriver.py
```

If you run `run.ps1`, then the above steps are not necessary.

5.5 Input Data

The table below outlines the core data files that the simulation uses as input.

Data File Name (.csv)	Description
aircraft	Contains the aircraft name, its max passenger capacity, max speed, max fuel capacity, max range, and miles per gallon
airports	Contains the airport name, its IATA code, the city and state it's in, the metropolitan area it serves, and its latitude and longitude coordinates
flights	Contains the source airport, destination airport, weighted distance (in km), number of passengers, and required fuel (in gallons)

The simulation is also set up to allow for adding and/or editing data. If you would like to change or add any values to the data files, simply open said file and add or edit the values you would like. The scripts used in the simulation are all type and value checked, so if any values are invalid or not of the right type, the simulation will exit and let you know what problematic value is and where it was triggered.

5.6 Flight Operations

In terms of passenger demand, serviced lights must meet certain criteria. At minimum, for a flight to be serviced, the passenger capacity must be greater than 30%. If that is the case, the flight is sent to a hub only if it arrives before the destination airport closes. If the passenger capacity exceeds 75%, then the flight is serviced directly to the destination.

In the simulation, flights are scheduled based on a greedy algorithm. The algorithm, located in scheduler.py, selects the most profitable flight, assuming an aircraft's operating hours does not exceed the minimum requirement for maintenance of 200 flight hours. Once the aircraft's operating hours meets or exceeds 200, the algorithm sets its next flight to the most profitable flight to a hub. The simulation's hubs are in Atlanta, Chicago, Dallas, and Denver.

The algorithm also actively aggregates passengers at hubs to maximize profits. For example, take a flight from Tampa to Charlotte. This flight only has 6 passengers, so servicing this flight is not worth servicing because the ticket sales do not cover the cost to service it. Instead, we introduce a layover and send the flight from Tampa to a hub. The hub chosen depends on the proximity to Tampa and the number of passengers at the hub that also want to go to Charlotte. A flight from Tampa to Charlotte would not go to the Denver hub.

At any point, an aircraft can have one of nine statuses, each with their own wait time:

- Available → no time
- In maintenance → 2160 minutes (36 hours)
- In maintenance queue → varies by airport
- Waiting on tarmac → NULL time
- Boarding without refueling → 25 minutes
- Boarding with refueling → 35 minutes
- Deboarding → 15 minutes
- In flight → depends on the flight
- Delayed → varies per delay

In the case of 'waiting on tarmac,' NULL time only means that it depends on the airport's available gates. If the airport has at least one gate available, the wait time is zero. However, if an airport has no gates available, the aircraft must wait until one is.

The simulation is set up to log every piece of information pertaining to an aircraft, passenger, and airport. This combined with knowing what an aircraft's status is helps the user know what the aircraft is doing at all times.

For example, a user can see that an aircraft departed from Los Angeles at 7:37 am, and is halfway to New York on a Boeing 747-600 with 100 passengers on board.

5.7 Flight and Weather Limitations

It should be noted that this module (`delay.py`) was implemented, but not tested and not included in the simulation. If you wanted to implement it into the simulation, instructions to do so are included in the module.

There are several days in which inclement weather could affect an aircraft's flight. Each day not mentioned progresses normally without trouble.

Day Number	Effect
Day 3	25% of all flights encounter bad weather and the flight time is extended for a random amount of time between 1 minute and 15% of the flight time
Day 5	20% of all flights originating above 40° N are delayed on the ground (not at the gate) due to icing for a random amount of time between 10 minutes and 45 minutes
Day 7	There is a strong jet stream and flights traveling due East have flight times extended by 12% flights traveling due West have flights shortened by 12%. All other flights have flight times impacted accordingly based on initial heading of the flight
Day 9	5% of flights are delayed at the gate by a random amount of time ranging from 5 minutes to 90 minutes
Day 11	You suffer an aircraft failure at one of the major Comfort Airlines hubs. The aircraft is taken out of commission for the entire day. The aircraft is towed away from the gate for unscheduled maintenance
Day 13	8% of all flights originating west of 103° W are canceled. Passengers must be put on other flights in order to reach their destination

5.8 Areas of Operation and Routes

The simulation uses the U.S.'s top thirty busiest airports (determined from Wikipedia[1]), including Paris's Charles de Gaulle airport and excluding Chicago's Midway International Airport. Serviced flights are only ones with a distance greater than 150 miles. There is only one flight to Paris, and it departs from New York's John F. Kennedy International Airport.

[1] https://en.wikipedia.org/wiki/List_of_the_busiest_airports_in_the_United_States

Airports are closed between 1 AM and 5 AM. If a flight lands between these times, a fee is given to the airline. To avoid this, the simulation does not service a flight if the arrival time is between that range.

6 General Aircraft Information

Comfort Airlines leases a total of 56 planes.

Aircraft Model	Amount Leased	Capacity	Max Speed (km/h)	Fuel Capacity (gallons)	Max Range (km)	Miles per Gallon
Boeing 737-600	15	132	1101	6875	5648	0.55
Boeing 737-800	15	189	1101	6875	5665	0.45
Airbus A200-100	12	135	1012	5790	5460	0.6
Airbus A220-300	13	160	1012	5790	5920	0.66
Boeing 747-400	1	416	1086	53765	11260	0.22

This chapter should contain general information on the aircraft used (e.g. manufacturer, type, number of aircraft, dimensions).

7 F.A.Q.

1. The simulation will not run.

A. Make sure you’ve installed both Python 3.10, as well as the Python required packages (in requirements.txt).

2. Is this simulation compatible with Windows and MacOS?

A. Yes.

3. Can I run the simulation on a mobile device?

A. No.

4. How do I update the simulation?

A. Simply repeat the steps in Section 5.2 to re-download it.

5. Does the simulation require an internet connection?

A. No, but you will need an internet connection to download Python and the Python packages. Everything else can be done without internet.

6. The provided user manual and documentation did not help my problem, what else can I do?

A. Submit a support ticket with Team Foobar.

7. The powershell script will not run on my Windows machine.

A. You need to enable remote execution in your system settings. In an elevated Powershell window (running as administrator), run `Set-ExecutionPolicy Bypass`, then you can run `powershell github/scripts/pipeline.ps1`.

Alternatively, you can go to Settings > For developers > Powershell > Toggle "Change execution policy to allow local powershell scripts to run without signing for remote scripts."

8 Glossary

A

available aircraft: an aircraft that is not in flight or undergoing maintenance (implies deboarded and waiting at a gate)

available flight: a flight with remaining demand that can be flown by the given aircraft type with sufficient maximum fuel, leaving within the operating hours of the source airport and arriving within the operating hours of the destination airport

available gate: a gate that is not currently being used by an aircraft nor scheduled to be used by an aircraft

aircraft needs maintenance: an aircraft with 200+ flight hours since last maintenance

H

hub: airport designated as central location with 11 gates. They are O'Hare, Denver Intl., Dallas/Fort Worth, Hartsfield-Jackson Atlanta Intl.

I

in flight: in the air, waiting on the tarmac, or at a gate preparing to arrive or leave

O

operating hours: 5 AM to 1 AM (exclusive)

Disclaimer

Team Foobar and this Comfort Airlines simulation are not to be held responsible for any damage or harm caused to a third party or any outside persons or entities from the use of this manual.