SkyWatch

https://test.flyuberjets.com/

https://github.com/GabrielVega19/SkyWatch

Gabriel Vega, Matthew Bedard, Andrew Farrell, Anthony Lawlor

April 26th, 2023
**Abstract**

As air travel continues to become more common, it is increasingly useful for hobbyists and frequent fliers to have access to flight data. Currently, a company called ADS-B Exchange offers data visualizations of flight information, however, this service will likely cease to be available in the near future. Anticipating this, we developed SkyWatch – a website that allows users to search for flight data and view results on an interactive map. With an intuitive and modern interface, our application aims to not only replace ADS-B Exchange upon their removal from the market but to actively compete as well. This document discusses the resulting project’s architecture, utilized technologies, final results, and more.

**Introduction**

Our team has created an intuitive and easy-to-use flight tracking software for modern web browsers. Our product SkyWatch allows users to be more informed about the flights which are happening all around them, 24/7, 365 days a year. The motivation for the project is to replace ADS-B Exchange (https://www.adsbexchange.com/), which is a software that allows users to either upload flight data from their own ADS-B transponders or view flight data from the community. An ADS-B transponder simply intercepts the data that planes broadcast to air traffic control within a certain radius. Data includes GPS location, altitude, ground speed, etc.

The kinds of people that would use this software are the hobbyists that own the transponders. They enjoy uploading, sharing, and then viewing flight data from all around the world. Our other target audience is members of the general public who have flights planned or are flying. This group would simply view the data for their flights.

The traditional source of flight data is provided by the government, however, they allow the wealthy owners of private jets to pay to make their flight data private, despite flight data being public
record. In response, ADS-B exchange created a community-driven source of this information to ensure that it remains public record. The reason we are trying to replace them is they were recently bought out, and the common belief is that ADS-B exchange is going to be taken down by the new owners. They are expected to be taken down because of the incentives from wealthy private jet owners to keep this information private.

Even if ADS-B Exchange remains on the market we plan to be a fierce competitor due to the problems with ADS-B Exchange. Their current interface is convoluted and difficult to navigate especially if you are part of the general market of flyers who are not familiar with industry jargon (see Fig. 1). We plan on being a competitor by bringing a modern and intuitive design that will attract both new users (the general public) and experienced users (the hobbyists).

Figure 1. A Screenshot of ADS-B Exchange interface
**Methodology**

For this project, we built a website with a homepage that will allow users to search for a certain tail number, and then navigate to an interactable map that will present the flight and aircraft data in a manner that is easy to digest.

When a user lands on the homepage, they will be presented with a simple page that contains a header, a text box, and a search button that leads the user to the map (see Fig. 2). To search for a flight a user will type the tail number into the search box and press the search button. While users are typing they are suggested tail numbers that are similar to their current search. This feature assists the user while searching for a flight and decreases the amount of time they spend navigating the site. We chose to keep it simple because one of the main pitfalls of the competition is convoluted UI. We believe that our simple design will be more intuitive for newer and inexperienced users while also decreasing the amount of time experienced users spend navigating through the website looking for their desired information.

![Original mockup for the homepage](image)

**Fig. 2** Original mockup for the homepage
When the user navigates to the map they will be presented with an interactable map that contains three markers. The first marker will be placed on the location of the aircraft which gets updated in real-time, and the other two markers will be placed on the location of the origin and destination airports. There will also be two collapsible overlays on the right side of the screen that contain other information about the flight and aircraft such as origin, destination, total miles, arrival time, aircraft make, and much more (see Fig. 3). This design is more readable and visually appealing when compared to ADS-B Exchange.

To structure our development of this project our team used the scrum framework. Additionally, we created a timeline with measurable milestones to keep us on track and we planned out the architectural design for our project beforehand, with the assistance of UML diagrams. Lastly, to aid in development, and ensure we were producing quality software, a continuous integration pipeline was implemented which automatically ran a host of unit tests, and automatically deployed our project. Using these strategies took some time to learn and set up, but we believe that it will save us lots of time in the long run, and keep us on track. We will provide our final thoughts on the process of creating this project in the work completed section.
Fig. 3 Original mockup of the map page

**Technology**

To bring the envisioned website to life, we engineered a custom full-stack application. The creation of a full-stack application requires the utilization of an extensive array of programming languages and tools. The key components of this application include a front end, a back end, a database, and a server to host everything. The front end consists of the two previously described pages, the backend will contain routes to service the generated requests, and lastly, the database will store all the raw data such as information about the different aircraft and its current flight path.

The front-end interface is created using a combination of HTML, CSS, and JavaScript, with the support of Tailwind, Jinja2, and Leaflet libraries. Tailwind enabled easier writing of CSS, Jinja2 enabled us to create reusable HTML templates, and Leaflet was utilized to develop our interactive map.
For the back end, we chose to use the Flask framework because of its quick and lightweight, prototyping capabilities. In addition, it runs on Python which is a user-friendly language that allows for quick development due to the numerous libraries written for the language such as GeoPy. For unit testing the backend we used PyTest which allows for quick and easy unit testing of all our code and is easily integrated into the CI/CD pipeline. We employed MariaDB for the database, as it is easy to integrate with Python, and performs better than other popular Database Management Systems such as MySQL.

The server hosting was executed on a Linux instance hosted on Amazon Web Services. The Linux instance employs Gunicorn to create the web server, NGINX to facilitate all network routing, CloudFlare to handle the DNS records, Daemons to ensure that the server stays up, and GitHub Actions to create the Continuous Integration/Continuous Deployment pipeline. The entire project used GitHub for version control since it allowed for easy collaboration on separate feature branches, which can be reviewed and tested before being merged with our production branch.

Figure 4 details how all of the mentioned technologies work together to create the final website. The front end takes in user input and then translates that into requests that get sent to the backend. The requests generated by the front end are more often than not asking for certain formatted data, that will then get presented to the user. The backend fulfills such requests and then sends them back to the front end. If the request entails fetching certain information, then the backend sends queries to the database to retrieve the raw data, which ends up getting formatted in whatever way the frontend requests.
Work Completed

We believe that by implementing the proposed architecture, with the technologies described in the section above, the main goals of this project will be met. Over the course of the semester, we were able to develop all the major components successfully, however, there were small compromises that had to be made in the interest of time. The compromises we made do not affect the overall functionality of the website, and will end up being implemented in the long term (see Future Work). Figures 5 and 6 contain screenshots of our completed interface.
Figure 5. Screenshot of our final Homepage

Figure 6. Screenshot of our final map
If you compare our initial mockups with our final interfaces, we stayed true to the desired look for our website. The homepage is simple yet effective because it provides users with all the functionality they need to find their desired flight, without adding any confusing jargon, and our map is a huge improvement over the competition. The map is more readable because it isn’t jam-packed with icons or other elements and the data is structured sensibly, it’s easier to use because of the simple flow of the website, and it’s more visually appealing than the competition because our styling utilizes modern techniques that most users have become accustomed to.

Looking back at the processes that were used to structure development, we believe they were very effective in keeping us on track, and allowed us to keep making constant forward progress. The timeline allowed us to measure where we were in the project and make adjustments if necessary, the scrum framework facilitated communication between the developers and gave us the flexibility for change if any problems arose, the unit tests showed us the areas where we broke our code saving us time while debugging, and lastly, the CI/CD pipeline saved us tons of time because it automated many tasks that would normally have to be performed by hand.

**Future Work**

Completely replacing all functionality from ADS-B Exchange was almost impossible within our time limit, so we focused on creating the most important components. There are more steps needed to completely replace ADS-B Exchange, such as creating a system for users to upload transponder data, linking the database up to a combination of the information from the government and the transponder data, and then marketing our product and bringing the ADS-B exchange community over.

Now that we have reached the end of the semester we have halted work on this project however we would like to work on it after graduation. The areas that we plan on working on are putting the finishing touches on the developed components such as allowing users to search for flights through
different methods, putting in more edge case preventions, and displaying the complete flight paths. Additionally, our current database has a couple of entries entered by hand, however, we would like to link the database up to the government data, and the UberJets databases so we can display live information. When this project is completed it will be a complete replacement for ADS-B Exchange. However, in the meantime, it will be a portal for tracking aircraft.

**Conclusion**

In conclusion, our team successfully created a functional and intuitive flight-tracking web application named SkyWatch. We utilized various programming languages and tools to create a custom full-stack application that allows users to search for a certain tail number and view real-time flight and aircraft data on an interactive map. In the long term, our project is meant to replace ADS-B Exchange and provide a better user experience to both hobbyists and members of the general public. The processes that we used to develop this application were found to be effective in keeping us on track and we will continue to use them in the future. In the end, we are satisfied with the final product and believe that it will be a strong competitor to ADS-B Exchange.

**References**


“Home - Serving the Flight Tracking Enthusiast - ADS-B Exchange.” ADS, 26 Jan. 2023,
https://www.adsbexchange.com/.


“Rapidly Build Modern Websites without Ever Leaving Your HTML.” Tailwind CSS,
https://tailwindcss.com/.

“Rising Use of Private Jets Sends CO2 Emissions Soaring.” Transport & Environment, 14 Sept. 2021,


“Welcome to Flask.” Welcome to Flask - Flask Documentation (2.2.x),
https://flask.palletsprojects.com/en/2.2.x/.


https://studycrumb.com/alphabetizer]