IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication April 25, 2022, date of current version April 25, 2022.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Analysis of Flight Disruption Data to Determine Problem Areas Utilizing Database Applications

FADJIMATA ANAROUA[1], (Student, ERAU), SEAN JENNINGS[2], (Student, ERAU), TYLER MARTIN[3], (Student, ERAU), AND JORDAN SANDERS.[4], (Student, ERAU)

[1]Embry-Riddle Aeronautical University, Daytona Beach, FL 32114 USA (issoufof@my.erau.edu)
[2]Embry-Riddle Aeronautical University, Daytona Beach, FL 32114 USA (e-mail: jennins3@my.erau.edu)
[3]Embry-Riddle Aeronautical University, Daytona Beach, FL 32114 USA (e-mail: martit52@my.erau.edu)
[4]Embry-Riddle Aeronautical University, Daytona Beach, FL 32114 USA (e-mail: sandej34@my.erau.edu)

Corresponding author: Khem Poudel (e-mail: poudelk@erau.edu).

This work was supported in part by the Embry-Riddle Aeronautical University.

ABSTRACT Flight delays represent a significant issue to airline profits and passenger satisfaction. Many factors can lead to a flight being delayed and/or cancelled. To compound the issue, COVID restrictions have created a shortage of skilled, willing labor and a disturbance in the supply chain. All aspects of air travel have been affected. For example, aircraft maintenance and new plane constructions have increased lead times due to parts unavailability. Mass layoffs or early retirements as a knee-jerk response to reduced travel in 2020 has exacerbated the problem. Pilot strikes protesting mandates led to mass flight cancellations in 2021. The group will evaluate flight delays, cancellations, and incident data with the goal of visualizing which airports, airlines, cities or states are experiencing the highest number of flight disruptions relative to others. In the era of new technological development, database applications are commonly used for data analysis to allow pattern recognition and large data distribution and organization. This study will mainly serve the purpose of flight data retrieval, the compilation of data and database design and finally output data visualizations. The outcome of the research will be presented in a user-friendly interface where the user can easily generate visualizations which can be used to analyze flight delay information.

INDEX TERMS Aviation, data, database, computers, information, technology, delays, cancellations, visualization, website, SQL, MySQL, HTML, CSS, Python

I. INTRODUCTION

"In 2007, the U.S government had endured 31–40 billion dollar downsides due to flight delays… in 2017, 76% of the flights arrived on time" [1] Air travel has led to a global interconnection of society. The desire to maximize the amount of time an aircraft is being flown, and therefore making a profit, drives airlines to decrease the amount of time in between flights. This greatly increases the chances of flight delays as there is a smaller time window in between arrivals and departures [2]. Any disturbance or cyber-attack can rapidly create global effects by causing monetary and reputational harm. Today, flight delay data has never been more relevant due to an increased focus on efficiency and improving passenger satisfaction. Our team believes it is urgent to address flight cancellation data and build robust systems capable of addressing and managing flight delay information on a global scale. "An accurate estimation of flight delay is critical for airlines because the results can be applied to increase customer satisfaction and incomes of airline agencies" [1].

A. DATASETS

The group has created a system to visualize flight delay data which can be used to recognize a pattern. Some of the data points the group has collected include: origin, origin city name, origin state abbreviation, destination, destination city name, destination state abbreviation, quarter, month, day of week, day of month, flight date, marketing unique carrier, departure time, departure delay, cancellations, and causes of delay [5] [6].

The data was retrieved from the Bureau of Transportation Statistics and uploaded to the database application using MySQL. MySQL was queried in the back-end to create the data visualization. A web interface was implemented using programming tools such as Python and HTML5/CSS framework. The website allows users to select an airport, airline, city or state. This input connects to the MySQL database through Python and uses data visualization techniques to show the user delay trends relative to that input in 2021.

B. LITERATURE REVIEW

People who have traveled by plane are familiar with one of the most inconvenient aspects of flying: delays. The plane may arrive late, there may be only one line for takeoff or landing, or severe weather may impose multiple hour delays (sometimes resulting in flight cancellation); regardless of the reason, flight delays are a major inconvenience for air travel passengers. As a result, with flight data from more than 300 thousand U.S. flights annually, we can acquire significant insights from this data to better understand flight delays and the associated causes.

Furthermore, because of the availability of large data sets for visualization, functional database systems may be used to help visualize these flight delays. This might be extremely useful for both travelers and corporations.

"In 2016, research for a post-flight data analysis using databases implied that in conventional systems, aircraft data was typically recorded in the form of files on a storage medium and handled by each flight sortie separately. As a result, as the number of flight sorties grows, so does the time spent searching for data for analysis. Instead of file-based flight data maintained by flight sortie, they proposed database-based data integration and management. This solution allows people to simply store and manage all of the real-time data in a database so that people may use it for visualization and in a variety of application programs. [7]"

Recent research has found it difficult to explain the principal reasons behind flight delays due to multiple factors being in play at a time.

Flight schedules can be subject to change. Because airline resources are be closely connected, delays could quickly spiral out of control if suitable recovery measures are not enacted. Despite the complexities, certain patterns of flight delays are consistent with the airline's schedule performance. The case study yielded some interesting outcomes [8].

"Air Carrier: The cause of the cancellation or delay was due to circumstances within the airline's control (e.g. maintenance or crew problems, aircraft cleaning, baggage loading, fuelling, etc.)

Extreme Weather: Significant meteorological conditions (actual or forecasted) that, in the judgment of the carrier, delays or prevents the operation of a flight such as tornado, blizzard or hurricane.

National Aviation System (NAS): Delays and cancellations attributable to NAS that refer to a broad set of conditions, such as non-extreme weather conditions, airport operations, heavy traffic volume and air traffic control.

Late-Arriving Aircraft: A previous flight with the same aircraft arrived late, causing the present flight to depart late.

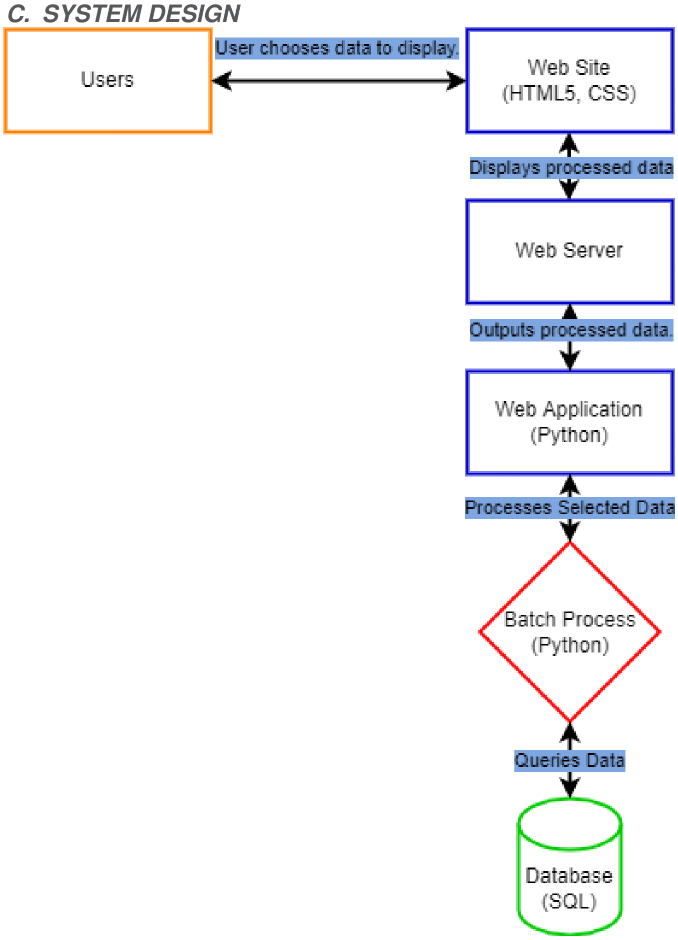
Security: Delays or cancellations caused by evacuation of a terminal or concourse, re-boarding of the aircraft because of a security breach, inoperative screening equipment and/or long lines in excess of 29 minutes at screening areas. [8]"

Our research uses the Relational Database Management System to give flight delay statistical information to the end user via a website. Moreover, this website allows the user to choose an airline, airport, or a geographical location. The website also shows the user a pie chart on the distribution of delays and a graph representing the delay based days of the week or airline.

A previous study by Wesonga, Nabugoomu, and Jehopio did an analysis of flight delays. They used a logistics model with twelve attributes and determined that " ... the number of freighter movements and non-commercial flights per day significantly influence both arrival and departure delays" [10].

Elmasri and Navathe stated that relational database management systems " ... provide flexibility to develop new queries quickly and to reorganize the database as requirements change." Additionally, relational database management systems are " ... the dominant type of database system for traditional database applications," [11].

C. SYSTEM DESIGN



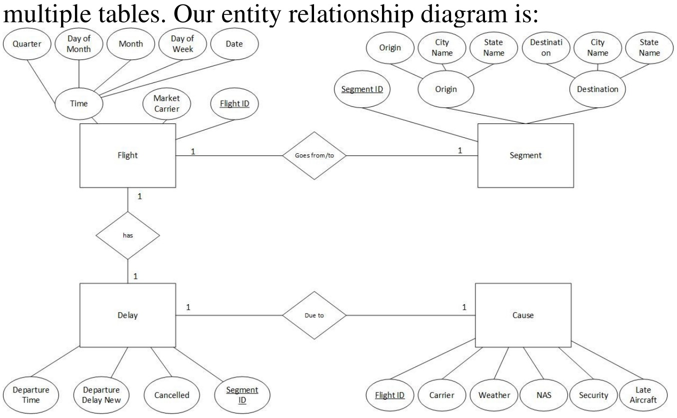
The above image shows the initial system design concept as well as the programming languages that were used to implement the system. The database is accessed when the user interacts with the web site by choosing the type of data he or she would like to interact with. That website was created with HTML5 and CSS.

Once the user makes a selection, the website requests information from the database server and then displays the data processed by the web application. Then, the web application uses a batch process to analyze the data from the database. Both the batch process and the web application used Python.

Finally, the database is accessed by the batch process to query the data. The database system language used for this project is SQL.

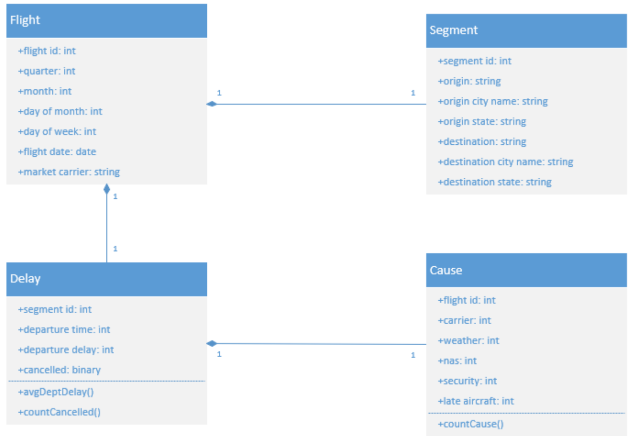
D. ENTITIES RELATIONSHIP (FLIGHTS_DATE, LOCATION, DELAYS, CAUSES)

An entity is a real-world structure or a specific data set that we want to try to mimic in our database. They are frequently identified as the system's primary nouns. For our study, we have attempted to design and establish a direct link between an entity and multiple tables of data; as in a relational database, data for a single entity may be kept in multiple tables. Our entity relationship diagram is:

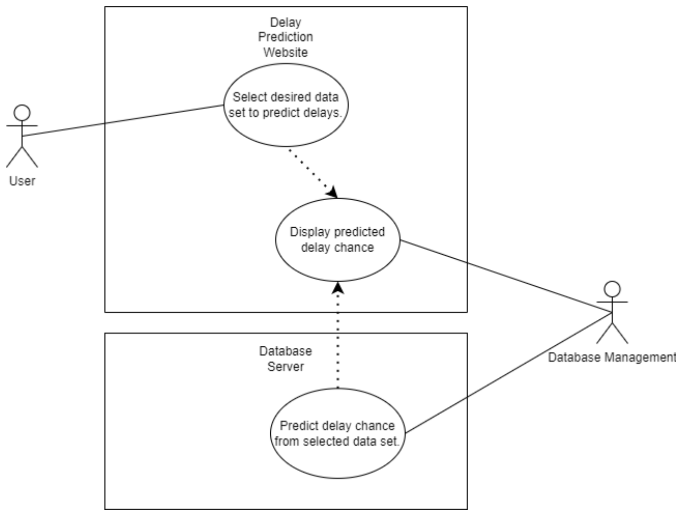


As can be seen above, each the Flight and Segment entities are related to each other in a one to one relationship called Goes to/from. The Flight entity has a primary key called Flight_id. Furthermore, the Segment entity has a primary key of Segment_id which is a foreign key to the Flight_id of Flight. The primary keys of Delay and Cause also are foreign keys to the Flight_id of Flight. The Flight entity also has an one to one relationship with the Delay entity called Has. Additionally, the Delay entity has a one-to-one relationship with the Cause entity called Due to.

The following image shows the UML(Unified Modeling Language) Diagram of the relational database.



The image below displays the use case diagram which illustrates how people interact with the system. The end user on the left side of the diagram selects his or her desired set of data on the delay prediction website from the front-end of the system. On the other side, the database management system (DBMS) interacts with the database server on the back-end of the system (i.e., creating the database).



The end user's desired set of data is analyzed to determine the specific 2021 delay information which is displayed to him or her. Once the data set has been selected, the DBMS accesses information stored in the database server in order to complete the task.

E. OTHER CONSIDERATIONS

Once a pattern has been established, the website will present its findings to the user. This web interface provides the user with data visualizations of 2021's delays, allowing the user to make their own predictions about whether a flight is worth taking. For example, a user could insert an airport and see that Fridays had a large average delay compared to the other days of the week. The user could then decide that it might be better to fly out the day before or the day after.

F. DATABASE DESIGN

Database design, at its most basic level, entails defining entities to represent various types of data and designing relationships between those entities. By "entities," we mean how the data sets are related to each other. The tables below illustrate this project's four entities: Flights, Segments, Delays, and Causes. Inside each entity, there are attributes and primary keys which were discussed in more detail in the Entities Relationship section.

Flights

flight_id	Quarter	month	day_of_month	day_of week	fl_date	Mkt_unk_carrier
1	1	1	23	6	01/23/2021	WN
2	1	1	7	4	01/07/2021	WN
3	1	1	7	4	01/07/2021	WN
4	1	1	7	4	01/07/2021	WN
5	1	1	8	5	01/08/2021	WN

segments

segment_id	origin	origin_city_name	origin_state_abr	dest	dest_city_name	dest_city_abr
1	DAL	Dallas, TX	TX	SAT	San Antonio, TX	TX
2	MCO	Orlando, FL	FL	MDW	Chicago, IL	IL
3	LAX	Los Angeles, CA	CA	STL	St. Louis, MO	MO
4	LAS	Las Vegas, NV	NV	MCI	Kansas City, MO	MO
5	BNA	Nashville, TN	TN	ATL	Atlanta, GA	GA

Delays

segment_id	dep_time	dep_delay_new	cancelled
1	928	0	0
2	923	23	0
3	816	16	0
4	1451	1	0
5	1436	113	0

Causes

flight_id	carrier_delay	weather_delay	nas_delay	security_delay	late_aircraft_delay
1	44	null	0	null	null
2	54	0	null	null	0
3	11	0	7	0	null
4	null	null	null	0	0
5	null	0	0	0	15

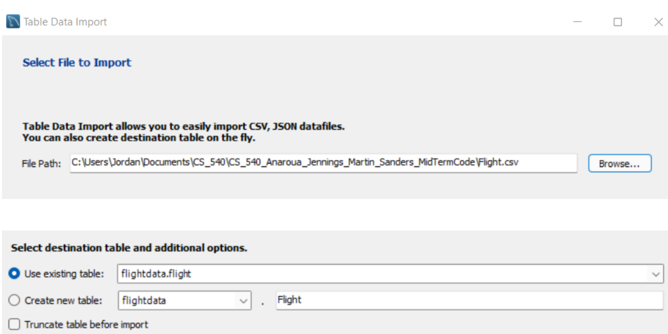
II. RESULTS AND DISCUSSIONS

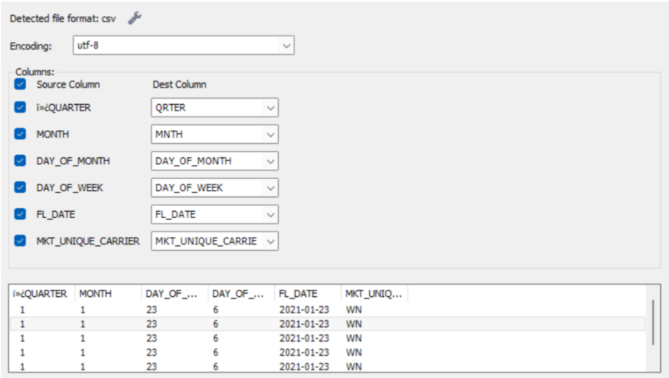
The relational database management system (RDBMS) was created via MySQL Workbench for the offline version of the program. The RDBMS consists of four tables: flight, cause, delay, and segment. These table structures can be seen in the Database Design section of this paper. The primary key of the flight table is flight_id. The other three tables have primary keys which are foreign keys to the flight table. Therefore, the flight table has the original unique id for each flight and each other table's id refers to the one created in the flight table. The code used to create this can be seen below.

```
CREATE TABLE FLIGHT

(   FLIGHT_ID   INT NOT NULL AUTO_INCREMENT,
        QRTER           INT     NOT NULL,
        MNTH            INT     NOT NULL,
        DAY_OF_MONTH INT NOT NULL,
    DAY_OF_WEEK  INT NOT NULL,
    FL_DATE      DATE ,
    MKT_UNIQUE_CARRIER CHAR(15),
    PRIMARY KEY(FLIGHT_ID)
);

CREATE TABLE SEGMENT
(   SEGMENT_ID   INT NOT NULL AUTO_INCREMENT,
        ORIGIN          CHAR(55)        NOT NULL,
        ORIGIN_STATE_ABR        CHAR(55)        NOT NULL,
        DEST CHAR(55) NOT NULL,
    DEST_CITY_NAME  CHAR(55) NOT NULL,
    DEST_STATE_ABR  CHAR(55) NOT NULL,
        PRIMARY KEY (SEGMENT_ID),
    FOREIGN KEY (SEGMENT_ID) REFERENCES FLIGHT(FLIGHT_ID)
);

CREATE TABLE DELAY

(   DELAY_ID INT NOT NULL AUTO_INCREMENT,
    DEP_TIME INT NOT NULL,
    DEP_DELAY INT NOT NULL,
    CANCELLED INT NOT NULL,
    PRIMARY KEY (DELAY_ID),
    FOREIGN KEY (DELAY_ID) REFERENCES FLIGHT(FLIGHT_ID)
);
CREATE TABLE CAUSE

(   CAUSE_ID INT NOT NULL AUTO_INCREMENT,
    CARRIER_DELAY INT NOT NULL,
        WEATHER_DELAY INT NOT NULL,
    NAS_DELAY INT NOT NULL,
    SECURITY_DELAY INT NOT NULL,
    LATE_AIRCRAFT_DELAY INT NOT NULL,
```

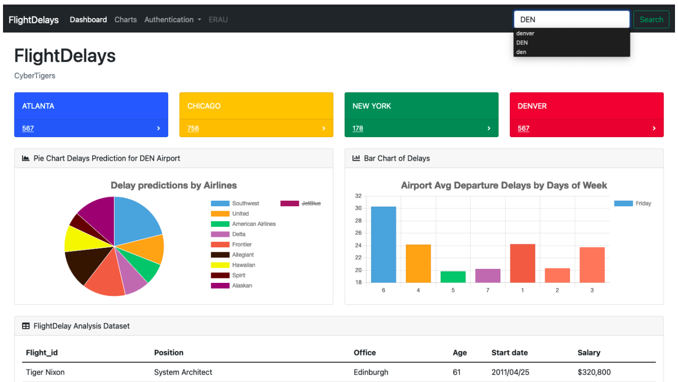
To load the data into these tables, the "import records from an external file" option was used. The csv files corresponding to each table were loaded using this option. The process of uploading the flight table data can be seen below.

```
SELECT * FROM FLIGHT
```





A website was created using HTML. It asked the user to search for an airport, city, state, or airline. The website then displayed data visualizations about delays associated with that particular airport, city, state, or airline. A static image of the website can be seen below.



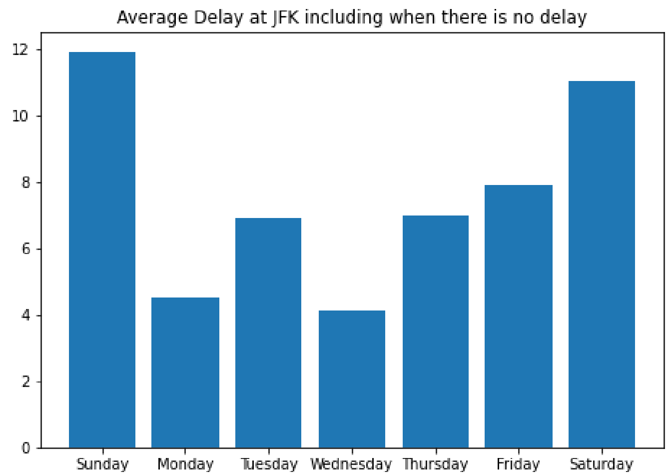
To connect the HTML website to the MYSQL RDBMS, the flask and flask_sqlalchemy Python libraries were used.

A. TESTING PROCESS

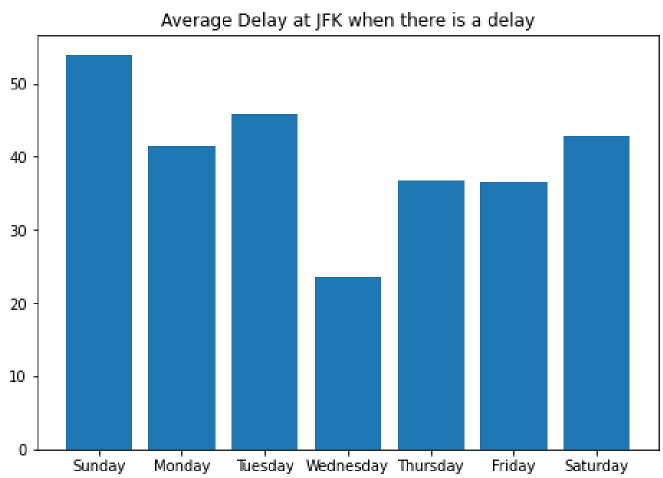
To test the queries needed and the data analysis to be shown to the user, some SQL queries were used. For example, delay by flight carriers. The delay_time column in the delay table is an integer value with a zero value indicating no delay. Using this, we can calculate the average time of delay in minutes by averaging all values that are greater than zero with a SQL statement such as: SELECT avg(delayḊEP_DELAY) FROM (delay INNER JOIN flight ON delayḊELAY_ID = flightḞLIGHT_ID) WHERE flightṀKT_UNIQUE_CARRIER = 'WN' AND delayḊEP_DELAY > 0.

The group had the idea to create an internet accessible version of the website that would pull data from an SQL database hosted online on a web server. The web server was to interface with the user via PHPMyAdmin which is a program that was installed onto the server via Softaculous. PHPMyAdmin was chosen for the web server to make running SQL queries on the database more user-friendly through an intuitive interface. The csv files were be converted to SQL statements and queries to be entered in via the Poudel52_flightdata "SQL" tab of PHPMyAdmin. Ultimately the web server was not used for the final version of the project.

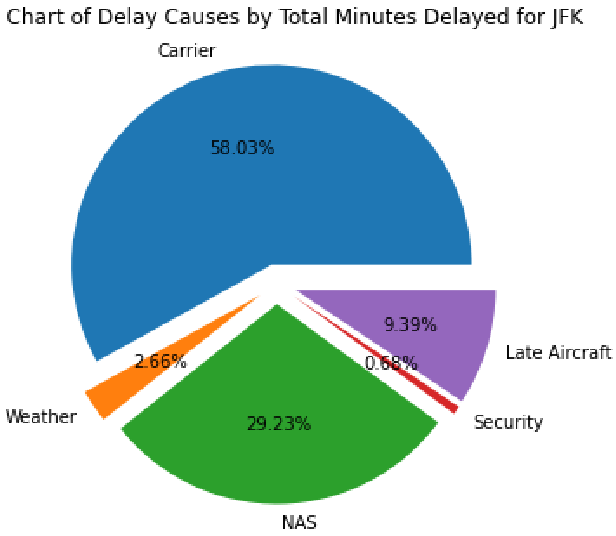
To test graphing and plotting techniques with the data, a Jupyter Notebook file was created. This file used the Matplotlib.pyplot library to implement the graphs. The Pandas library was also utilized to manipulate the data into the necessary slices to create the desired graphs. The data was sliced by Airport using an input to test one of the user input options in the website. After the user inputted the Airport Code, Pandas sliced the data in the different tables by that Airport Code. It was then further reduced by creating seven slices for each day of the week. The average delays for each of those days were plotted in a bar graph via Matplotlib.Pyplot as shown below.



This data was further sliced to only include instances that had a delay. Another bar graph was created to show the average delay of a flight at the user selected airport for each day of the week when there was a delay. An example of this graph is shown below for when the inputted airport is 'JFK'.



Furthermore, the Cause data set or table was sliced to only include instances from the inputted airport. Then, the total minutes delayed by all causes was found. The total minutes for each individual cause were calculated. Using the total minutes overall and the individual totals, percentages of each cause were found. This was then plotted via Matplotlib.Pyplot using the same input value of 'JFK'.



At one point our group hosted our website, https://flightdelays.poudel.tech/ using a web hosting service called HostGator (https://www.hostgator.com/) which is no longer in service. Files, such as index.html, were modified through cPanel. cPanel (https://cpanel.net/) is software installed under the server's operating system that provides a graphical user interface for interacting with a web server and modifying its files as an alternative to FTP (file transfer protocol).

A connection between MySQL and Python was created. This connection queried the flightDelay database and was utilized to plot the MySQL queries. The resulting charts displayed the averages and percentages of delays by choice of city, state, airlines, and airport within website search field.

The mysql.connector module was used to establish the Python connection to the MySQL database locally. Matplotlib, Pandas and NumPy were used in conjunction with mysql.connector to plot results of various delays and averages of delays within our data set in a form of python bar and pie chart.

The MySQL Database connection to Python allowed us to query average departure delays, percentages of delays by cause, and overall data set delays from our flightDelay database.

The selected queries were run through MySQL, and the result achieved allowed for displaying and graphically displaying various delays and problem areas within airports, airlines, cities, or states. This data set is comprised of ten (10) major airlines operating in the US from the BTS website. We included over 300 cities ranging from smaller to big cities with a higher amount of traffic, thus the chance of delay occurrence may be different dependent upon the location. Depending on the city, state, or specific airport, an overarching delay pattern was not determined for this study. Various results may appear with gaps, certainly due to the type of delay, the number of carriers' scheduled flights, and availability of data within the set.For the delay types from the overall data set, it was noticed that most delays were caused by carrier/airlines, NAS delay, and late_aircraft_delay. Furthermore, there were very few or no cancellations in most airports. Security_delay and weather delay data appear to be non-significant. These delays have less chance of occurrence as they are often unpredictable which makes them less likely to cause huge delays within airports, cities, or states.

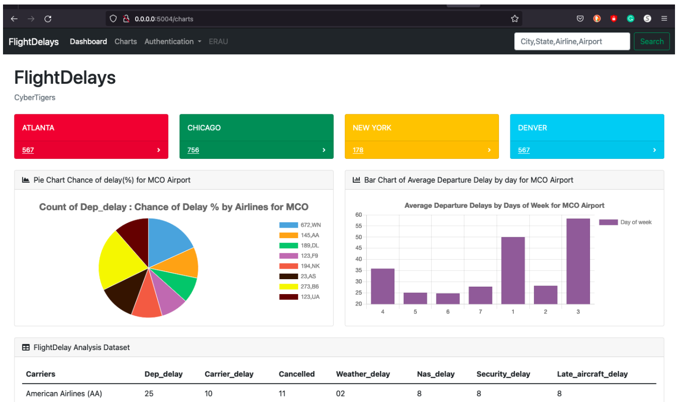
Some examples of queries and results are shown below.

USER INPUT	SQL Query to MySQL database within python file	Result
User inserts string as "City" name='Denver' in search field	SELECT mkt_unique_carrier, AVG(dep_delay) FROM flight JOIN (delay,segment) ON (flight.FLIGHT_ID = delay.DELAY_ID AND flight.flight_id = segment.Segment_id) WHERE (MKT_UNIQUE_CARRIER = "UA" OR MKT_UNIQUE_CARRIER = "WN") AND (delay.DEP_DELAY > 0) AND (origin_city LIKE "%Denver%") GROUP BY mkt_unique_carrier	Output: mkt_unique_carrier = ['WN'] averages = [Decimal('16.5953')] Comment: All Airlines within the database tables were selected and displayed with their average departures delay within the "city"
User inserts string as "State" name = 'FL' in search field	SELECT mkt_unique_carrier, AVG(dep_delay) FROM flight JOIN (delay,segment) ON (flight.FLIGHT_ID = delay.DELAY_ID AND flight.flight_id = segment.Segment_id) WHERE (MKT_UNIQUE_CARRIER = "UA" OR MKT_UNIQUE_CARRIER = "WN") AND (delay.DEP_DELAY > 0) AND (origin_state_abr LIKE "%FL%") GROUP BY mkt_unique_carrier	Output: mkt_unique_carrier = ['WN'] counts = [Decimal('19.9940')] Comment: All Airlines within the database tables were displayed with their average departures delay within the state including all cities in the state.
User inserts string as "Airline" name = 'WN' in search field	SELECT mkt_unique_carrier, sum(dep_delay)/ sum(dep_delay+cancelled+carrier_delay+weather_delay +security_delay+late_aircraft_delay)*100, sum(CANCELLED)/ sum(dep_delay+cancelled+carrier_delay+weather_delay+ security_delay+late_aircraft_delay)*100, sum(carrier_delay)/ sum(dep_delay+cancelled+carrier_delay+weather_delay+ security_delay+late_aircraft_delay)*100, sum(weather_delay) /sum(dep_delay+cancelled+carrier_delay+weather_delay+ security_delay+late_aircraft_delay)*100, sum(nas_delay)/sum(dep_delay+cancelled+carrier_delay+weather_delay+ security_delay+late_aircraft_delay)*100, sum(security_delay)/ sum(dep_delay+cancelled+carrier_delay+weather_delay+ security_delay+late_aircraft_delay)*100, sum(late_aircraft_delay)/ sum(dep_delay+cancelled+carrier_delay+weather_delay+ security_delay+late_aircraft_delay)*100 from flight join (DELAY,cause) on (flight.flight_id=cause.cause_id AND flight.flight_id=DELAY.DELAY_id) WHERE (MKT_UNIQUE_CARRIER = "WN") GROUP BY mkt_unique_carrier'	Output: Percentages = [Decimal ('67.9658'), Decimal('0.1926'), Decimal('15.8790'), Decimal('0.9653'), Decimal('17.8117'), Decimal('0.1092'), Decimal('14.8881')] Comment: The output result prints all percentages of delays for the airlines by delay type/cause of delays. Result was displayed in a form of pie chart.
User inserts string as "Airport" name = 'STL' in search field	SELECT day_of_week, avg(delay.DEP_DELAY) FROM FLIGHT JOIN (delay,SEGMENT) ON (flight.flight_ID = segment.SEGMENT_ID AND flight.flight_id = delay.delay_id) WHERE segment.ORIGIN like "%STL%" AND delay.DEP_DELAY > 0 AND day_of_week =1 OR day_of_week =2 or day_of_week=3 or day_of_week=4 or day_of_week=5 or day_of_week=6 or day_of_week=7 group by day_of_week	Output: day = [4, 5, 6, 7, 1] delay = [Decimal('1.9619'), Decimal('3.2200'), Decimal('3.2050'), Decimal('7.7681'), Decimal('14.9200')] Comment: The output/result prints average departure delays (all airlines in our dataset located at STL) by day of the week. Result was displayed in a form of pie chart.
User view of overall delays by cause (excluding departure delay) from our current dataset.	SELECT sum(carrier_delay), sum(weather_delay), sum(nas_delay), sum(security_delay), sum(late_aircraft_delay) FROM cause	Result: Each result from the set was divided by (100*total of allvalue from the set) to get a percentage of the delay type. *. Sum of result was originally set in minutes from the dataset

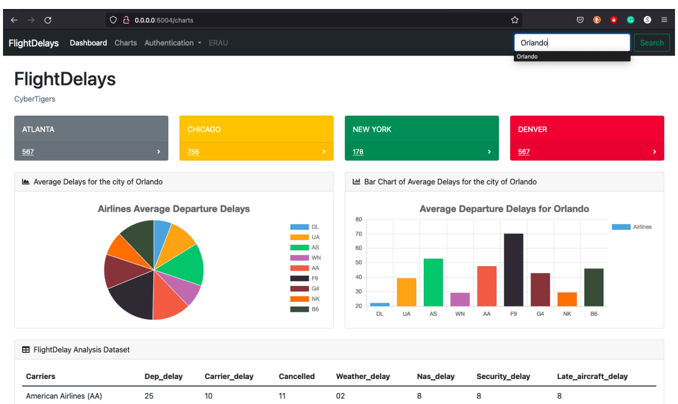
B. FINAL PRODUCT

The final version of the web interface had the following visualizations.

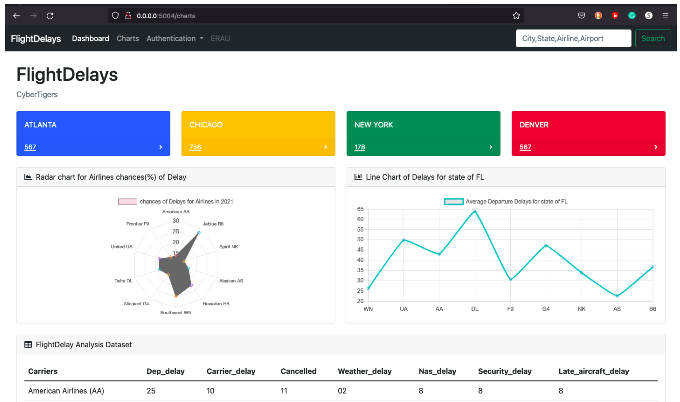
When the user entered an airport code, the chance of delay by airline at the airport was displayed as a pie chart. Additionally, a bar graph of the average minutes of delay by the day of the week was shown for that particular airport. The following image shows the results for MCO (Orlando International Airport).



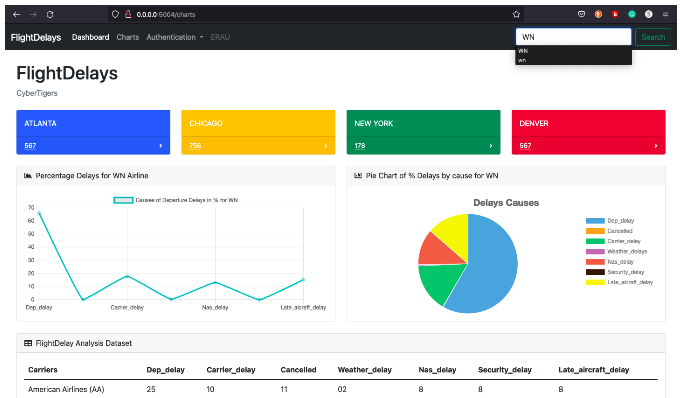
If the user inserted a city, then the chance of the airline delay within the city was displayed as a pie chart and the average delay in minutes for each airline in the city was shown. An example of this is shown below using Orlando as the city.



For state inputs, the website outputs a radar chart for the airline chances of delays. Furthermore, it displays a line chart of delays for the specified state. If the specified state was Florida, the web interface would output:



Lastly, the airline input results in two data visualizations: causes of departure delays as a line chart and chance of delays as a pie chart. An example of the airline input is shown below using Southwest (WN) as an example.



III. CONCLUSION

There are few modern day inconveniences less anticipated than being stranded at the airport. Being able to make predictions and prudent choices based on historical flight delay information can save a significant amount of time, trouble, and tears. The group has created a system that directly interfaces with historical flight delay data in a database in order to visualize flight delay trends. An accessible, easy-to-understand graphical display of flight delay patterns from the past year could save travelers an untold amount of headache, stress, and trouble. Possible future applications of this concept could be delineated in the form of a phone app, an applet as part of an airline's website (for high-performing airlines), an informational service as part of a booking website, or simply a private independent resource for travelers and interested parties. Accessible information in a digestible format can be a powerful decision-making tool for a wide-reaching audience.

A. POTENTIAL MACHINE LEARNING FUTURE WORKS

This study could be furthered if more data is implemented into the DBMS. If there is more than just one year's worth of data, then certain machine learning algorithms could be applied. For example, logistics regression could be implemented to predict the likelihood of delay at a specified time and date. This would require attributes like time of day, time of week, month, and more. Another potential machine learning algorithm that could be implemented could be a form of regression (potentially linear or Gaussian-process regression) to predict the delay in minutes of a flight, which would require similar attributes. For any machine learning algorithm, more research would need to be done. It is important to note that the aviation industry is very volatile, meaning the slightest change in the economy or the general societal concern for safety can greatly affect the airline industry. For example, in 2001 and 2002, many people were scared to fly due to the event of 911. In 2008 and 2009, people preferred not to travel by plane due to the tight budget from the 2008 Financial Crisis. Lastly, the SARS-2-CoV-2 (COVID-19) Pandemic greatly affected the air travel during the 2020, and 2021 years. This means that machine learning may not be extremely predictive due to instability of the data set to the current events of the world. Due to this fact (along with the current amount of data), this group decided it would be more beneficial to give the users the means to decide on their own what impact last year's delays makes on their flight choices today.

REFERENCES

[1] M. F. Yazdi, S. R. Kamel, S. J. M. Chabok, and M. Kheirabadi, (2020). Flight delay prediction based on deep learning andhttps://www.overleaf.com/project/61f8892572a7d2ad2bee0b91 Levenberg-Marquart algorithm. Journal of Big Data, 7(1). https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00380-z.

[2] J. J. Rebollo and H. Balakrishnan, (2014). Characterization and prediction of air traffic delays. Transportation Research Part C: Emerging Technologies, 44, 231-241. https://www.sciencedirect.com/science/article/pii/S0968090X14001041

[3] AhmadBeygi, S., Cohn, A., Guan, Y., Belobaba, P. (2008). Analysis of the potential for delay propagation in passenger airline networks. Journal of Air Transport Management, 14(5), 221–236. https://www.sciencedirect.com/science/article/pii/S0969699708000550

[4] Flightconnections - "all flights worldwide on a map!" (n.d.). Retrieved February 1, 2022, from https://www.flightconnections.com/

[5] Bureau of Transportation Statistics (2021). Air Carriers : T-100 Domestic Segment (U.S. Carriers) (51276413) [Data set]. Bureau of Transportation Statistics. https://www.transtats.bts.gov/DL_SelectFields.asp?gnoyr_VQ=FIMQO _fu146_anzr=Nv4%20Pn44vr45

[6] Bureau of Transportation Statistics (2021). On-Time : Marketing Carrier On-Time Performance (Beginning January 2018) (51276413) [Data set]. Bureau of Transportation Statistics. https://www.transtats.bts.gov/DL_SelectFields.asp?gnoyr_VQ=FGKQO_ fu146_anzr=b0-gvzr

[7] Jong-Ik, S., Gyu-Se J. (2016). Flight Data Visualization and Post-test Flight Data Analysis System by Using Database. International Journal of Mechanical Mechatronics Engineering IJMME-IJENS Vol:16 No:02. https://ijens.org/Vol_16_I_02/164802-5757-IJMME-IJENS.pdf

[8] Anderson, A. B. A., Kumar, A. J. S., Christopher, A. B. A. (2019). Analysis of flight delays in aviation system using different classification algorithms and feature selection methods. The Aeronautical Journal, 123(1267), 1415-1436. doi:http://dx.doi.org/10.1017/aer.2019.72

[9] Wang, Y., Yakun, C., Chenping, Z., Wu, F., Hu, M., Vu, D., . . . Eugene, S. H. (2020). Universal patterns in passenger flight departure delays. Scientific Reports (Nature Publisher Group), 10(1) doi: http://dx.doi.org/10.1038/s41598-020-62871-6 L.K. Hassan, B.F. Santos, J. Vink, Airline disruption management: A literature review and practical challenges, Computers Operations Research, Volume 127, 2021, 105137,SSN 0305-0548,https://doi.org/10.1016/j.cor.2020.105137.

[10] Ronald Wesonga, Fabian Nabugoomu, Peter Jehopio, Parameterized framework for the analysis of probabilities of aircraft delay at an airport, Journal of Air Transport Management, Volume 23, 2012, Pages 1-4, ISSN 0969-6997, http://www.sciencedirect.com/science/article/pii/S09 https://doi.org/10.1016/j.jairtraman.2012.02.001.

[11] Elmasri, R. Navathe, S. B. (2015). Fundamentals of Database Systems. Pearson. https://iran-lms.com/images/images/Books/PDF/Fundamentals-of-Database-Systems-Pearson-2015-Ramez-Elmasri-Shamkant-B.-Navathe.pdf