AN ENHANCED CLOUD BASED MODEL FOR FLIGHT DATA RECORDER (FDR)

BOUKARI SOULEY,
Department of Mathematical Sciences (Computer Science), ATBU Bauchi, Nigeria
bsouley2001@yahoo.com

ARIH SARAH NKEMDILIM,
Computer Science Programme, National Mathematical Centre, Abuja, Nigeria
snjeakor@yahoo.com

***Abstract*—Over the years, there has been the question of improving the mode of collecting flight data record from an aircraft which has a traditional device called the Flight data recorder (FDR) that records both mechanical data, electrical data as well as cockpit voice records popularly known as the "blackbox". It has been a subject of study over the years in research to have flight data records transmitted to a cloud repository in order to have a better assured way of keeping flight data records in the case of an emergency situation where the data needs to be retrieved. Several research works have been carried out in this area to store flight data records to cloud repositories which grant a faster access to such data for analysis. Others also made attempts in the areas of data compression in order to optimize the collection of data to the cloud. The only issue here is that these data collected are primarily for investigative purposes in case of a crash or an after-flight analysis. None has been able to provide flight assistance using the cloud-based FDR while a flight is still air bound. In this work, we have developed a system that collects flight data to a cloud repository and checks the condition of the flight based on such data in order to provide some form of assistance even while the flight is still air bound. Our system cross checks the flight data collected and compares it with the active database created which have a set of rules for a normal flight condition and provides a clue to the condition of a flight that is air bound. All flights with an abnormal condition are flagged red and this allows flight control tower administrators to alert the aircraft concerned as well as alert some emergency units for a standby given the location of the flight.**

Keywords - Flight Data record, Cloud repository, Database,Active Database, Aircraft, Data

I. INTRODUCTION

A Flight Data Recorder (FDR) is a replaceable computer element used to record specific aircraft performance parameters in airplanes. It could either be digital or analog depending on the age of the aircraft. The purpose of an FDR is to collect and record data from a variety of aircraft sensors onto a medium designed to survive an accident. Its task is recording pilots' inputs, electronic inputs, sensor positions and information sent to any electronic systems on the airplane [1]. Flight Data Recorder is informally called "black box". It is designed to be quite small and carefully manufactured to withstand the influence of a high speed and the heat of an extreme temperature [2].

In fact, Federal Airline Authority (FAA) regulations stipulate that FDRs retain just the last two hours of recorded information. Thus, many of the recorders have capacity to store only 2 hours of data. From history, the two types of "flight recorder" carried on aircraft are the FDR and the cockpit voice recorder (CVR). Sometimes both types of recorder are fitted or combined into a single unit. The design of an aircraft is built in such a way that the Flight Data Recorder as well as the Cockpit Voice Recorder (CVR) is placed at the tail of the aircraft which has been said to be most suitable place for its survival in case of a crash according to designers.

Usually, a flight-data acquisition unit (FDAU) receives various discrete, analog and digital parameters from a number of sensors and avionics systems and then routes them to a flight data recorder (FDR) and, if installed, to a Quick Access Recorder (QAR). Information from the FDAU to the FDR is sent via specific data frames, which depend on the aircraft manufacturer. Integration of FDAU functions into software required by other aircraft system components is now being seen, as in the case of the Enhanced Airborne Flight Recorder (EAFR) installed on the Boeing 787.

In the area of storage management, FDR faces some challenges. A common difficulty is when Flight Data Recorders run out of memory space [3]. The memory space size of a typical Flight Data Recorders is too small. Though up to date high density FLASH memory devices have created which have facilitated the Solid-State Flight data Recorders (SSFDR) to be manufactured with a quite larger memory size. Many airplanes are now equipped with SSFDR and no longer make use of disk drives which were low in memory.

Additionally, in the past twenty-five years the density of memory chips has significantly multiplied and the capability to record thousands of parameters for hundreds of flight hours in flight data recorders or quick assess recorders is now possible. [4]

II. Literature Review

The work by [5] considered several architectural models from the many Enterprise frameworks available which share the same objectives but vary in focus, scope and intent. Some of these architectural models considered in their work are:

- Zachman's Framework as contained in [6]. This approach is widely used for the development of enterprise-wise information System architectures and is considered as a reference model against which other frameworks can map themselves [7].
- RM-ODP was presented in [8] which uses a well-understood object-modelling technique (OMT) and is developed by highly reputable agencies such as ISO and International Telecommunications Union.
- TOGAF as presented in [9]. TOGAF is a freely available, industry standard generic framework.
- C4ISR / DODAF - These frameworks were developed mainly for the use of the US Department of Defense [10] and [11].

III. Materials and Methods

A. *Proposed model*

The proposed model is basically set to achieve its aim and objectives in the following steps:

- Designing an active database with events, conditions and actions to be triggered from the control tower in case of an abnormal flight condition detected.
- Initiate a cloud-based service to record FDR from the aircraft
- Integrate the above-mentioned systems for flight monitoring and assistance.

The proposed model is hereby presented in the figure 1 below which is an improvement upon [5]. The model was adapted from their work while the improvement is on the inclusion of the active database section and its synchronization with the aircraft to achieve flight monitoring and assistance.

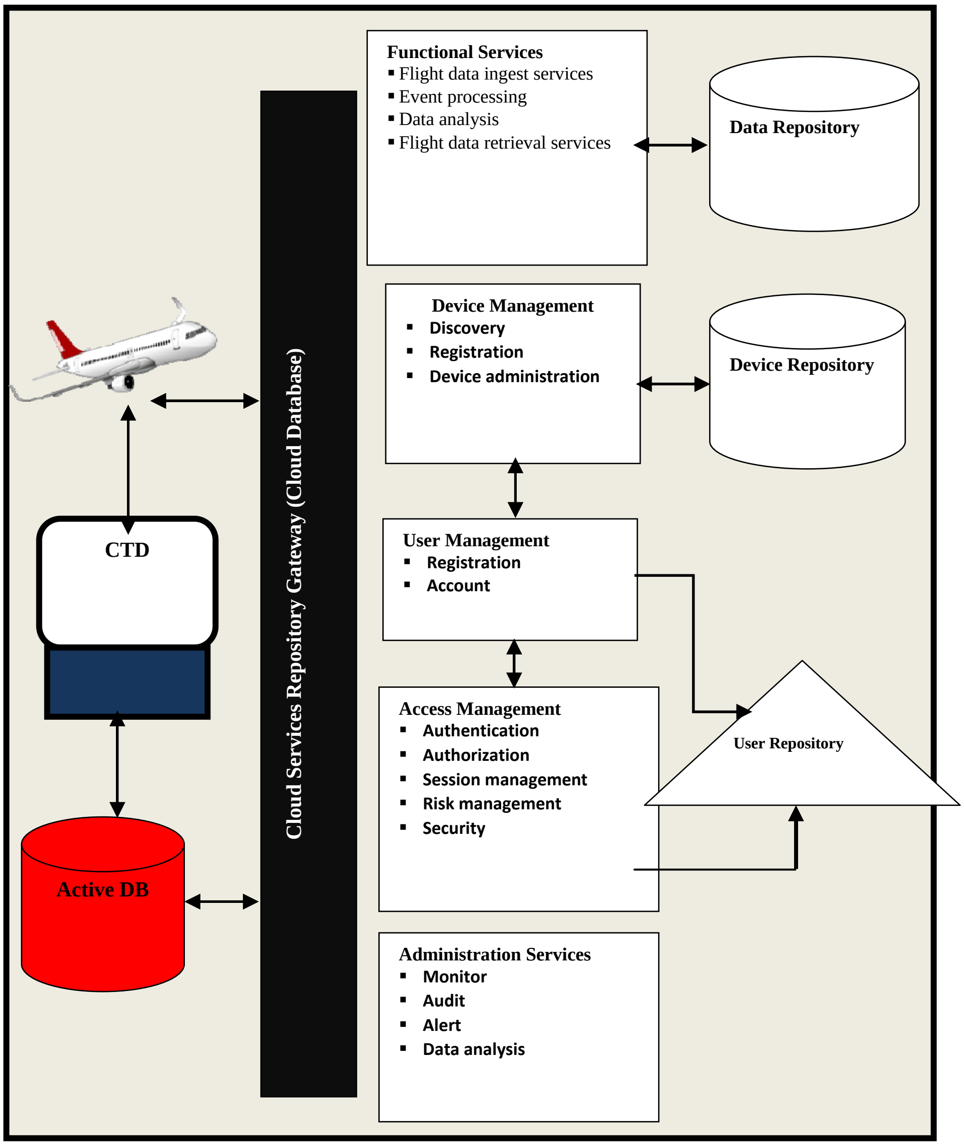

Figure 1:Proposed model

B. Data flow diagram for the proposed model

The proposed model aforementioned in this work can be seen in a summary form from the data flow diagram below. The diagram shows the real-time working principle and flow of data from one entity in the model to another. The data repository in the cloud collects the data generated by the aircraft which is stored on the FDR on board the aircraft while it is still air-bound and places the data in a ready form for analysis and further manipulation and control. The measures described above have been provided as an aid to the data manipulation, storage, control and general security. The active database which is embedded to the control tower system device interacts with the cloud repository to detect and indicate any abnormality in the flight. The control tower having been alerted then provides real time assistance to the aircraft. Here we understand how the flow of data happens in the proposed system.

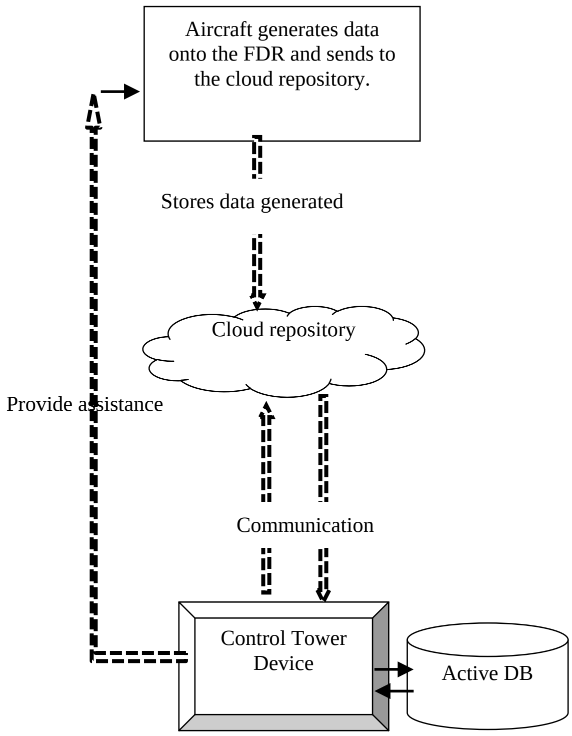

Figure 2: Working Principle of the proposed model

C. System analysis and design

The approach used in the design of the required system is presented here. We make integration between the use of databases and its tools while linking with the front-end application to provide the necessary assistance specified in this work. We first discuss the functional requirements and specifications of the system and then delve into the conceptual design methodology.

D. Functional requirements/specifications

The functional requirement of a system specifies explicitly what a system is supposed to achieve in its design and implementation. It summarizes or rather details on the objectives of the entire work and what the researcher seeks to achieve.

Building an Enhanced Flight Data Record system that provides flight assistance to flights that are even still air bound. It extends the power of transmitting flight data record to cloud to repositories for better access to flight data in case of an accident investigation and after flight analysis. The extension it seeks is to provide flight assistance where necessary in the case of identifying an abnormal flight condition from any of the aircrafts currently air bound.

Below are the specifications the proposed system seeks to achieve:

- Create an interface to allow administrators log in to the front end of the application
- Provide a mechanism for authenticating user log in
- Redirect to appropriate page in response to a log in attempt depending on the success of the log in process
- Provide an admin interface to select what operation to carry out
- An admin can choose to view air bound flight status
- An admin can choose to view a specific flight status
- An admin can choose to monitor a specific flight
- The interfaces communicate with the database and responds with appropriate results to queries

E. Application blueprint

Here we show the internal structure and communication flow of the entire application in a diagrammatic format for a proper understanding of the specifications aforementioned in the work. The blueprint maps the implementation course of the application that implements the methodology of approach to meet the problem stated earlier.

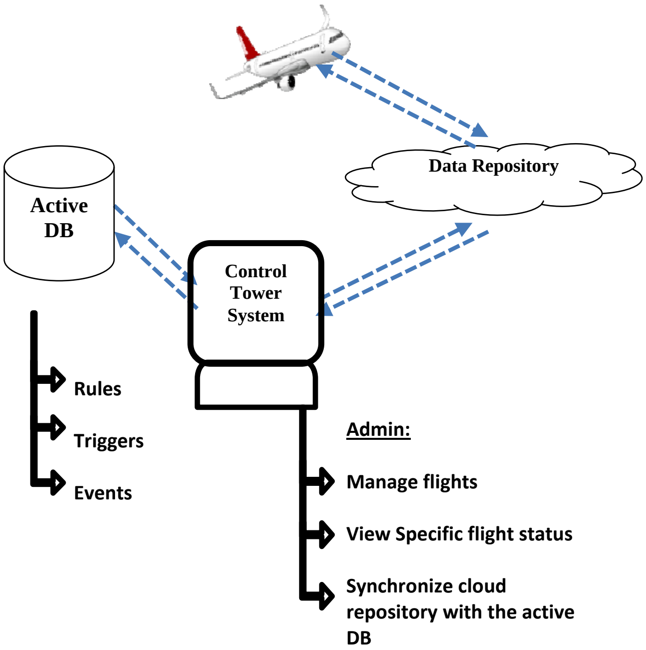

Figure 3: Application Blueprint

F. Active databases

There are traditional database implementations which are primarily for the purpose of data storage which may be needed for retrieval purposes which could either be updating the record in the database or other manipulation actions that are allowed on the records in the database. There are however, other forms of database implementations which are more robust in their approach in that they tend to make decisions with or without the supervision of an administrator. One of such implementation is known as an Active Database.

The approach of an Active Database implements certain rules by creating events and specifies some conditions and associated actions if triggered. The model of an Active Database's implementation compared with a traditional database implementation is presented in figure 4 below:

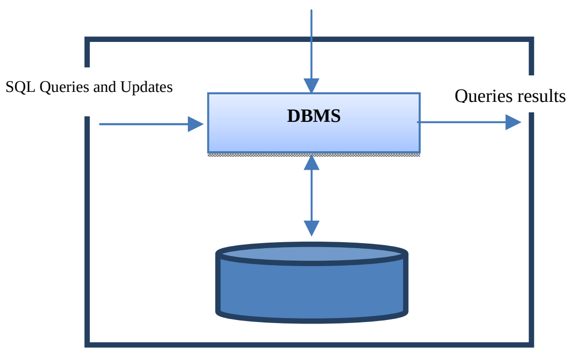

Figure 4: General Principles of Database Systems

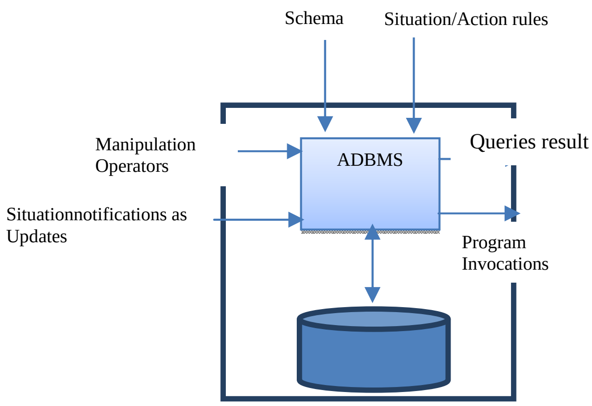

Figure 5: General Principles of an Active Database System

IV. RESULTS/ DISCUSSION

A. *Database interface*

The database contains the various tables that holds aircraft as well as flight details. Here in this work is the implementation of tables as specified earlier in the methodology of this work. The various tables aforementioned were implemented here with its ac-companying entities.

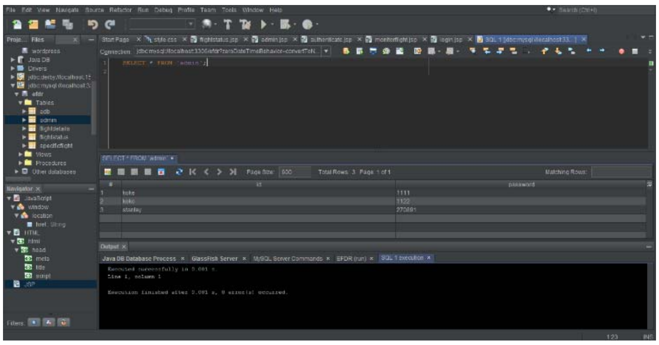

Figure 6: Admin Interface

B. *Admin interface*

The admin interface provides the platform for a user seeking access to the system to be authenticated and verified as a valid user before access can be granted. Implemented here is the security feature which prevents unauthorized persons from gaining access to the system which holds vital flight and aircraft data. Upon clicking the enter key on the keyboard the script checks for various levels of details and responds with an appropriate response. First it checks to make sure the User ID and pass-word fields are not empty then it checks for correctness by searching through the database for a match in both User ID and Password. In any case, an appropriate response is given as observed below.

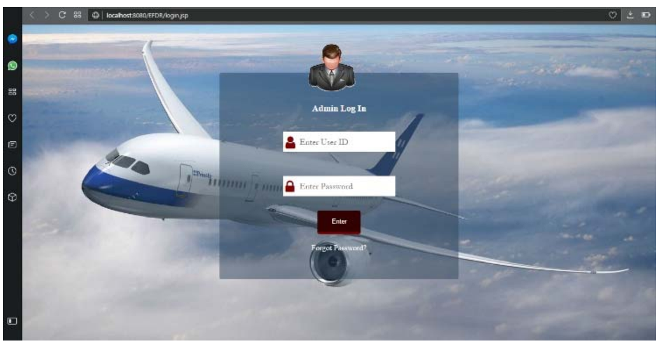

Figure 7: Admin Interface

C. *Incorrect User ID/Password*

When a user enters their details as can be seen on the screenshot, the processing script makes a search through the database for a user with matching de-tails provided. In the case where none is found the result is as shown below.

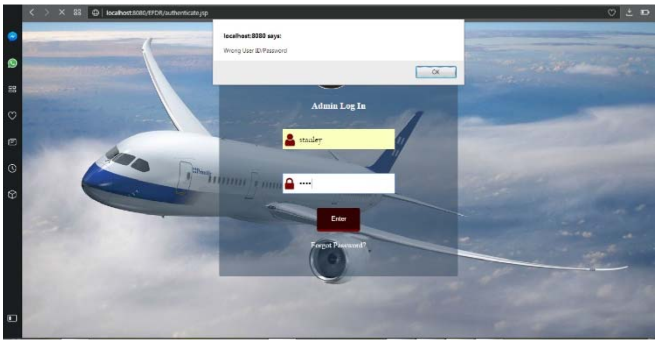

Figure 8: Incorrect User ID/Password verification

D. The dashboard area

Upon successful Logging in, the user is directed to this section of the application to choose what operation to carryout. The user can either view flight status of all air bound flights or monitor a particular flight by viewing its flight data analysis section or by viewing its specific aircraft details. This can occur in the case where a particular flight has been detected to have abnormal flight condition. This feature leads the admin to find out more details about that aircraft of interest.

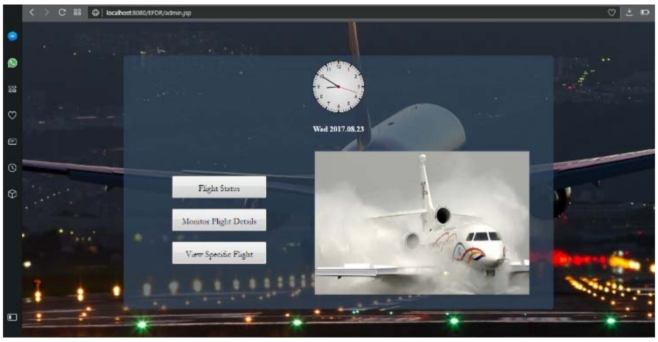

Figure 9: Admin Dashboard

E. Air-bound flights

This feature in the implementation displays all air-craft that are currently air-bound (that is, flights that are currently in progress). Here the attributes dis-played are only the flight number, the expected time of arrival as well as the total time spent so far as par the flight. This is calculated by subtracting the expected time of flight from the time at take-off. The status of the flight is also displayed here which tells whether a flight is of concern or not. The active database changes the status of the flight as it observes its predefined data in the database. Upon detection of abnormality, the status changes to either “warning” or “alert” depending on the severity.

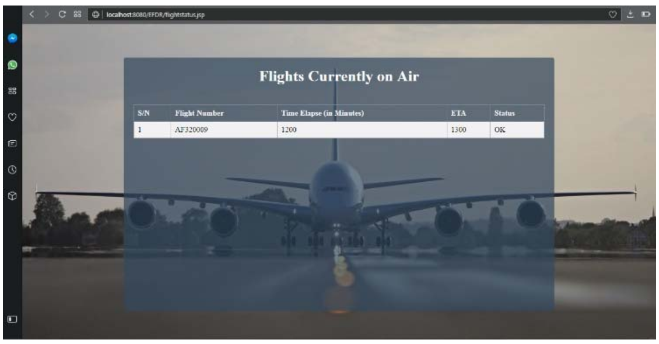

Figure 10: All air-bound flights

F. Monitoring a flight

An admin could, based on the current status observed of a flight, decide to find out more about a flight from its take off point to its current point. The implementation here simulates the movement of an aircraft from one heading towards another (destination). Here we see a flight with number as “AF320009” with all its considered flight data from its take-off to its current point.

The first column displays the timestamp as the flight progresses. The latitude and longitude coordinates are also given here to tell exactly what point the aircraft is at the moment. The altitude (height above sea level) is also given here alongside the speed of the aircraft. To implement this, some assumptions have been made. They are:

- The aircraft has a minimum altitude of 1300 meters and a maximum of 2000 meters. When it is detected to go outside this range then a warning signal is flagged with a yellow background.
- The aircraft has a minimum speed of 2100 kilometers per hour and maximum of 4000 kilometers per hour. When it is detected to go outside this range then a warning signal is flagged with a yellow background.

In the case where the aircraft is outside of its range on both the altitude as well as the speed then an "alert" signal is flagged with a red background indicating that its condition is critical.


Flight AF320009

Time Stamp	Latitude	Longitude	Altitude	Speed	Status
0	50	-130	1000	2000	Alert
1	49	-129	1000	2012	Alert
2	48	-128	1200	2314	Warning
3	47	-127	1400	3744	Ok
4	46	-126	1600	3736	Ok
5	45	-125	1800	2588	Ok
6	44	-124	2000	2666	Ok
7	43	-123	2000	2313	Ok
8	42	-122	1600	2949	Ok
9	41	-121	1400	2654	Ok
10	40	-120	1200	3273	Warning
11	39	-119	1000	2769	Warning
12	38	-118	1000	2788	Warning
13	37	-117	1000	2818	Warning

Figure 11: Monitoring a Flight

G. Flight with no alert signal

Here we see an aircraft with given flight data and only one warning signal almost after take-off. The implementation here is similar to the above but with the following assumptions made:

- The aircraft has a minimum altitude of 3100 meters and a maximum of 6500 meters. When it is detected to go outside this range then a warning signal is flagged with a yellow background.
- The aircraft has a minimum speed of 4500 kilometers per hour and maximum of 8500 kilometers per hour. When it is detected to go outside this range then a warning signal is flagged with a yellow background.

In the case where the aircraft is outside of its range on both the altitude as well as the speed then an "alert" signal is flagged with a red background indicating that its condition is critical.


Flight NG32000

Time Stamp	Latitude	Longitude	Altitude	Speed	Status
0	50	-130	3379	5014	Ok
1	49	-129	3038	5407	Warning
2	48	-128	4810	5421	Ok
3	47	-127	3194	5309	Ok
4	46	-126	3877	5294	Ok
5	45	-125	3747	5217	Ok
6	44	-124	3185	5440	Ok
7	43	-123	4650	4891	Ok
8	42	-122	3831	5253	Ok
9	41	-121	4959	4676	Ok
10	40	-120	4865	5053	Ok
11	39	-119	3616	5079	Ok
12	38	-118	3820	5379	Ok
13	37	-117	4167	4837	Ok

Figure 12: Flight NG32000

V. CONCLUSION

Enhancing flight data record has been the main aim of this work. This research work has focused on detecting an abnormal flight condition even without the Pilot checking in on that in order to provide assistance to air-bound flights.

So far, this research work has shown that flight data record collected to the cloud repository can be used for more than just accident investigation but can be employed to provide real time assistance. The implementation has been able to detect ab-normal flight conditions and flag them with appropriate colors where necessary for further investigation.

With the advancement in technology of aircrafts today, flight data recorded to cloud repositories can be enhanced to perform more than what it is usually used for (crash investigations). This can be of great assistance to Pilots who may have lost communication with the control tower when experiencing some abnormal conditions.

A. *Limitations of the study*

This research work has been implemented based on some assumptions and thus, some limitations. The limitations of this work include:

- Flight data collected are just limited to altitude, speed, latitude and longitude.
- Flight data were simulated and not real due to difficulty to access a live flight data.
- Database and the entire application were hosted locally not online.

B. *Areas for further research*

As a recommendation for further research, an ex-tension can be made on this work by improving on its available features in areas such as:

- Connecting with local authorities of a known flight location.
- Being able to place a direct call to authorities notifying them of the presence of an aircraft with some abnormal conditions within their jurisdiction.

References

[1] Li, J., Carlson, E. B., & Lacis, A. A. (2014). Application of spectral analysis techniques in the inter-comparison of aerosol data, Part 4: Synthesized analysis of multisensor satellite and ground-based AOD measurements using combined maximum covariance analysis. Atmos. Meas. Tech., 2531-2549.

[2] Endre, B., & Winterhalter, M. (2012). United States Patent No. 8121752.

[3] Pinchas,W., & Yair, W., (2013). Efficient Memory Control for Avionics and Embedded systems. International Journal of Embedded System,

[4] Wiseman, Y. (2016). Can Flight Data Recorder Memory Be Stored on the Cloud? Journal of Aviation Technology and Engineering , 16–24.

[5] Ayeni, B. K., & Yisah, S. O. (2015). Cloud Based Architecture Solution for Aircraft Flight. British Journal of Applied Science & Technology.

[6] Zachman, J. A. (1999). Framework for information Systems Architecture. IBM Systems Journal, 38(2/3):454–470.

[7] Zachman, J. A. (1987). The framework for information systems architecture. IBM Systems Journal, 26(3):276–292.

[8] Putman, J. (2001). Architecting with RM-ODP. Prentice Hall, Upper Saddle River.

[9] TOGAF. (2011). TOGAF. The Open Group: Version 9 Enterprise Edition. Retrieved from TOGAF: Available:https://www2.opengroup.org/ogsys/catalog/G116 Version 9.1 01

[10] C4ISR. (1997). Architecture Working Group: C4ISR Architecture Framework Version 2; Retrieved from C4ISR:Available:http://www.afcea.org/education/courses/archfwk2.pdf

[11] DODAF. (2010). DOD Architecture Framework Version 2.02.Retrieved from DODAF: http://dodcio.defense.gov/Portals/0/Documents/DODAF/DoDAF_v2-02_web.pdf