INTEGRATED DESIGN PROJECT

CSE-460

System Development

Group E (Sec A)

Serial	ID	Name	Email
1	201814002	Maj Md Rezoanul Hafiz Chandan	mrhchandan@gmail.com
2	201714111	Lt Md Anindya Mostofa	anindya.mostofa@gmail.com
3	201814011	Flg Offr Nafiun Nahar Polok	nafiunpolok@gmail.com
4	201814010	Offr Cdt Asif Shahriar Arnob	arnobsmail@gmail.com
5	201814012	Offr Cdt Raiyan Khan	ryankhankayoger@gmail.com

Table of Contents

1.   Scenarios of the Usecase.

1.1   Scenario 1.

Directorate of Air Operations need to know the condition of a particular pilot during his flight. Flying Officer ABC has been showing poor performance in his test flights. The particular officer has had some family issues but there is no way to find out the linkage between this to his poor flying. He knows that the pilot is going to be grounded because of his poor performance in flying. The instructors assumes that his poor flying is because of his over excitement during flying. They need to figure out whether it happens for his overexcitement or other issues. Basing on his mental status they can initiate a report to determine his future assignments. So, Directorate of Air Operations is seeking the reports of "Health Monitoring of a Pilot using IoT wristband" system. It will assist him to decide the linkage between the pilot's health issues and his flying performance.
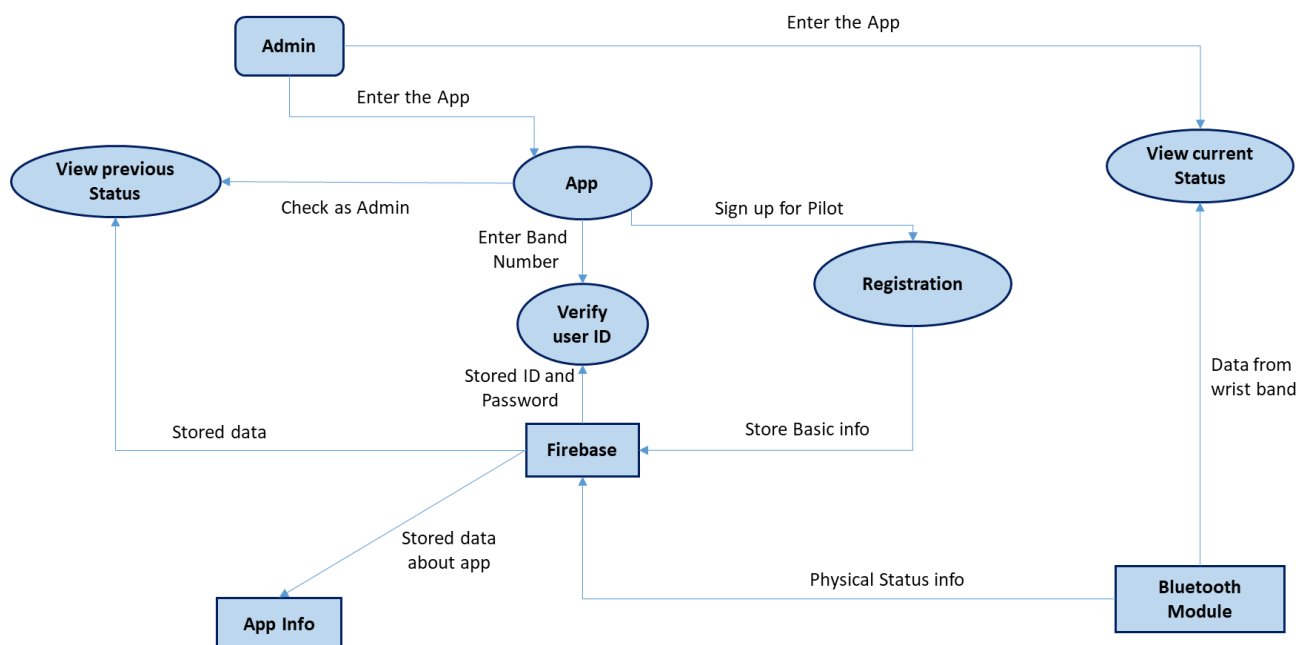
Figure-1: Data flow diagram for Scenario 1.

1.1 Scenario 2.

Flight Surgeon of 103 ATTU, Squadron Leader XYZ is assigned to check the health condition of a pilot before his/her flight. It is a standing operating procedure to check the health status before a flight. Yet the accidents happen. Few of this are related to aircraft failure and few are with pilots. A couple of months ago FO CSE had undergone a fatal aircraft crash and died on the spot. SL XYZ checked his vitals to be okay before flight but he could not monitor him during his flight. Human mental and physical state changes due to change in environment. So, if he wished if he could know the condition of a pilot's health during a flight, he could save lives as well as aircrafts. For continuous monitoring of a pilot's health, he is seeking the help of "Health monitoring system of a pilot using IoT wristband" to provide necessary advise if any adversity occurs. The pilot can also check his health status stored in the database previously.
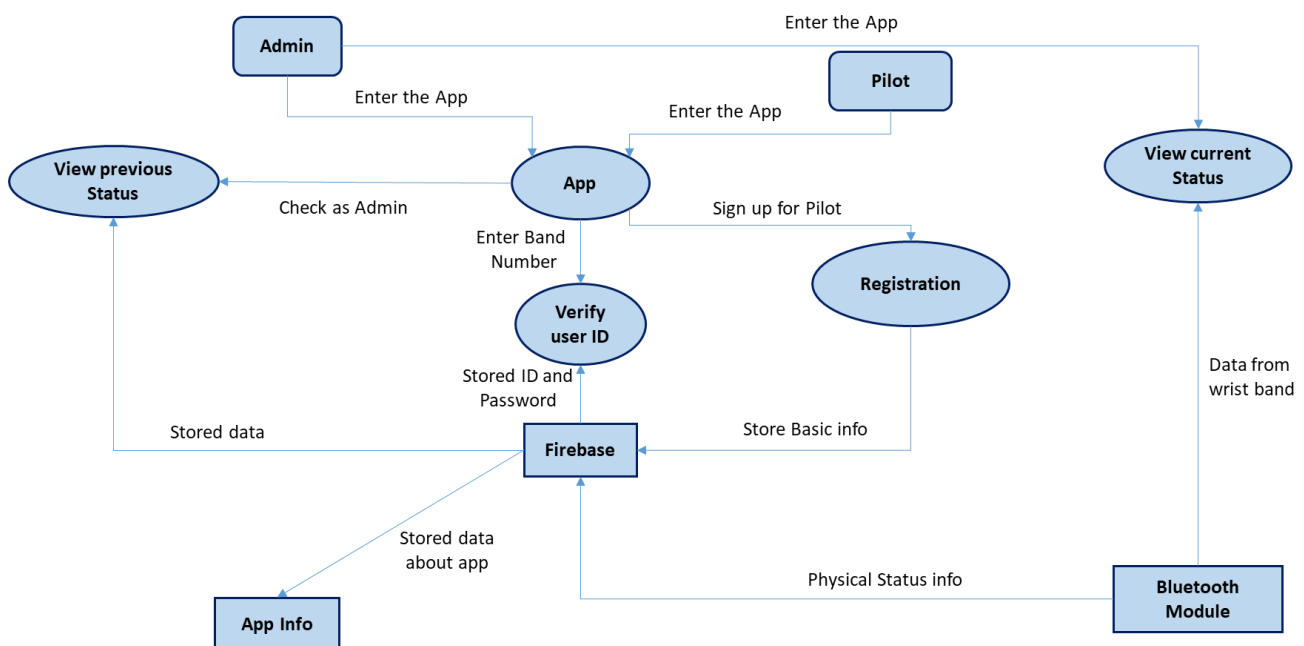

Figure-1: Data flow diagram for Scenario 2.
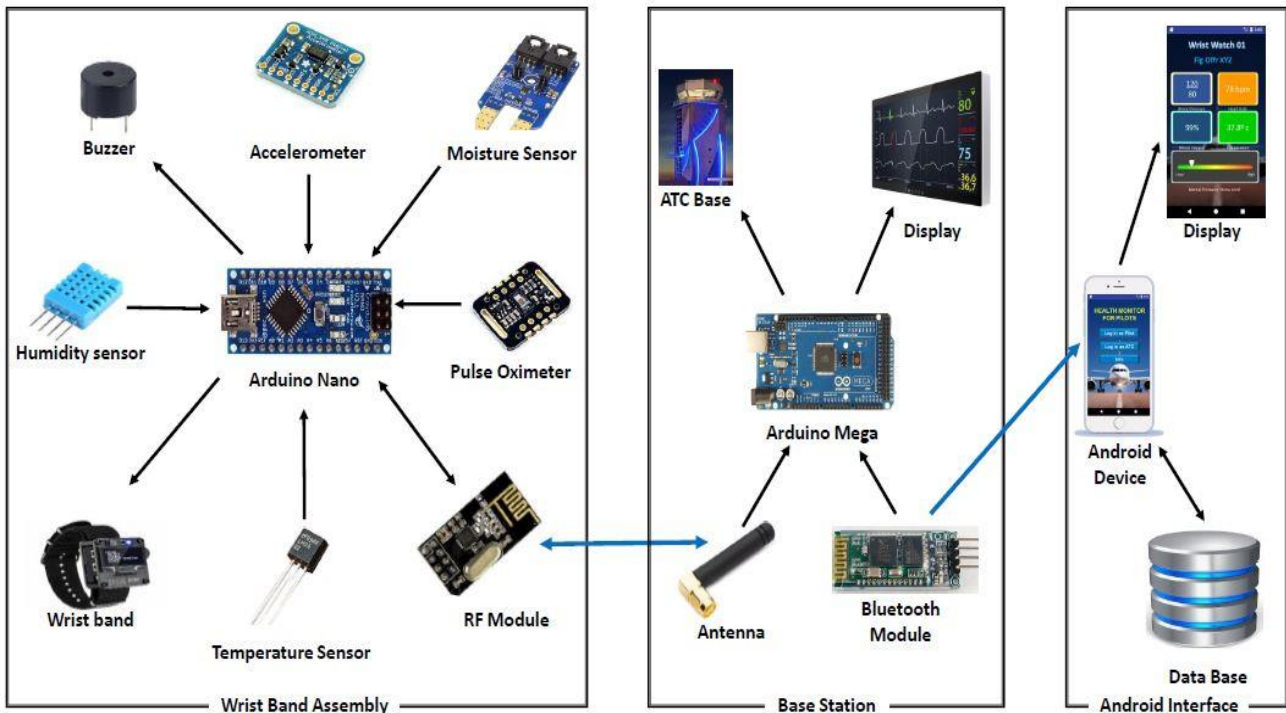
1.3   System Architecture.

Fig: System Architecture

2.   Platform Used in the System.

The platform of the whole project is a cross platform. Both web and mobile applications can be used for data integration and processing the whole data through the algorithm and generate report. Thus we have generated a applications based till now, but modifications can be done through which, it can be done through website platforms.

Front End

The **front-end** is what a user sees and interacts with (user interface). The app is built on MIT App Inventor. It uses Google Blockley and a graphical user interface (GUI) which allows to drag and drop visual objects to create an application. JavaScript is used in the frontend of the app. A brief description regarding these are described below:

Name	Description
	MIT App Inventor: MIT App Inventor is a free, cloud-based service that allows us to make our own mobile apps using a blocks based programming language. We access App Inventor using a web browser (Chrome, Firefox, Safari etc.). App Inventor lets us develop applications for Android phones using a web browser and either a connected phone or emulator. The App Inventor servers store our work and help us keep track of our projects.
	JavaScript: JavaScript is a text-based programming language used both on the client-side and server-side that allows us to make web pages interactive. JavaScript is an event-based imperative programming language that is used to transform a static HTML page into a dynamic interface. JavaScript code can use the Document Object Model (DOM), provided by the HTML standard, to manipulate a web page in response to events, like user input.

Back End

The back-end is all of the technology required to process the incoming request and generate and send the response to the client. This typically includes three major parts:

a.   The server. This is the computer that receives requests.

b.   The app. This is the application running on the server that listens for requests, retrieves information from the database, and sends a response.

c.   The database. Databases are used to organize and persist data.

For backend data storage we are using firebase. Java is used in the backend of the development of the app. C programming language is used in the backend of Arduino.

Name	Description
	Firebase: Firebase is a platform developed by Google for creating mobile and web applications. Firebase is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps, grow their user base, and earn profit. It is built on Google's infrastructure. Firebase is categorized as a NoSQL database program, which stores data in JSON-like documents.
	Java: Java is a high-level programming language developed by Sun Microsystems. It was originally designed for developing programs for set-top boxes and handheld devices, but later became a popular choice for creating web applications. The Java syntax is similar to C++, but is strictly an object-oriented programming language. Java is also known for being stricter than C++, meaning variables and functions must be explicitly defined. This means Java source code may produce errors or "exceptions" more easily than other languages, but it also limits other types of errors that may be caused by undefined variables or unassigned types. Java programs are multiplatform and can run on different platforms.
	C Programming Language: C is a general-purpose procedural computer programming language supporting structured programming. By design, C provides constructs that map efficiently to typical machine instructions. It has found lasting use in applications previously coded in assembly language. C is an imperative procedural language. It was designed to be compiled to provide low-level access to memory and language constructs that map efficiently to machine instructions, all with minimal runtime support.

3.   UI Screenshots.

The UI design is made as user friendly as possible to make it more comfortable for the users.
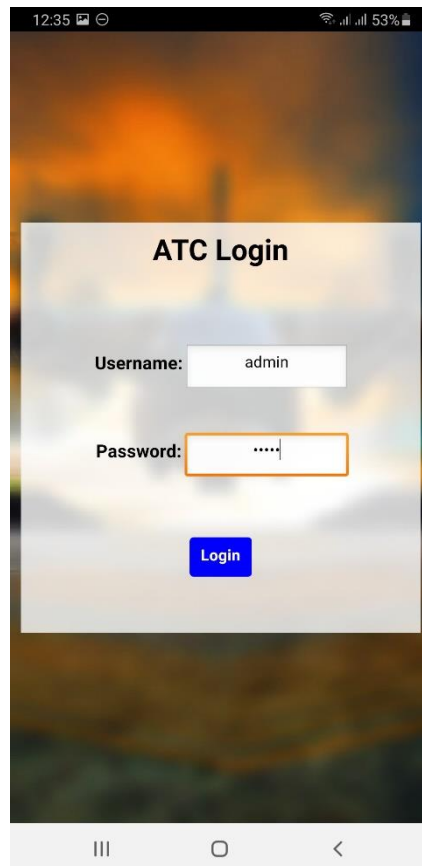
3.1   Home Page.

From home page we can have the facility to login as pilot and admin   as well as we can have access to the information of the app.
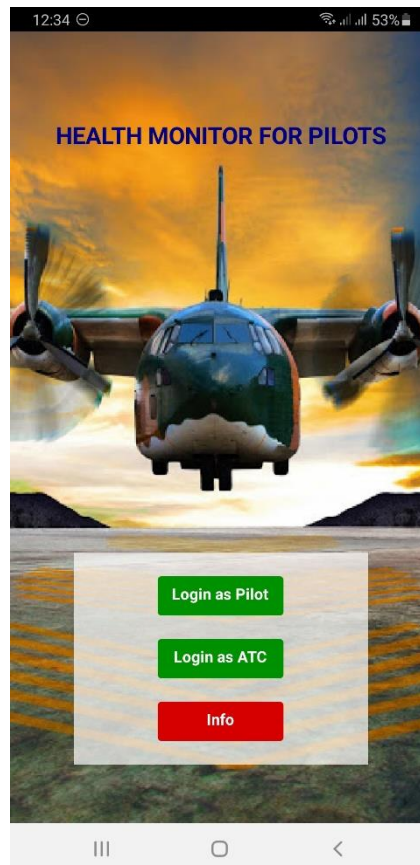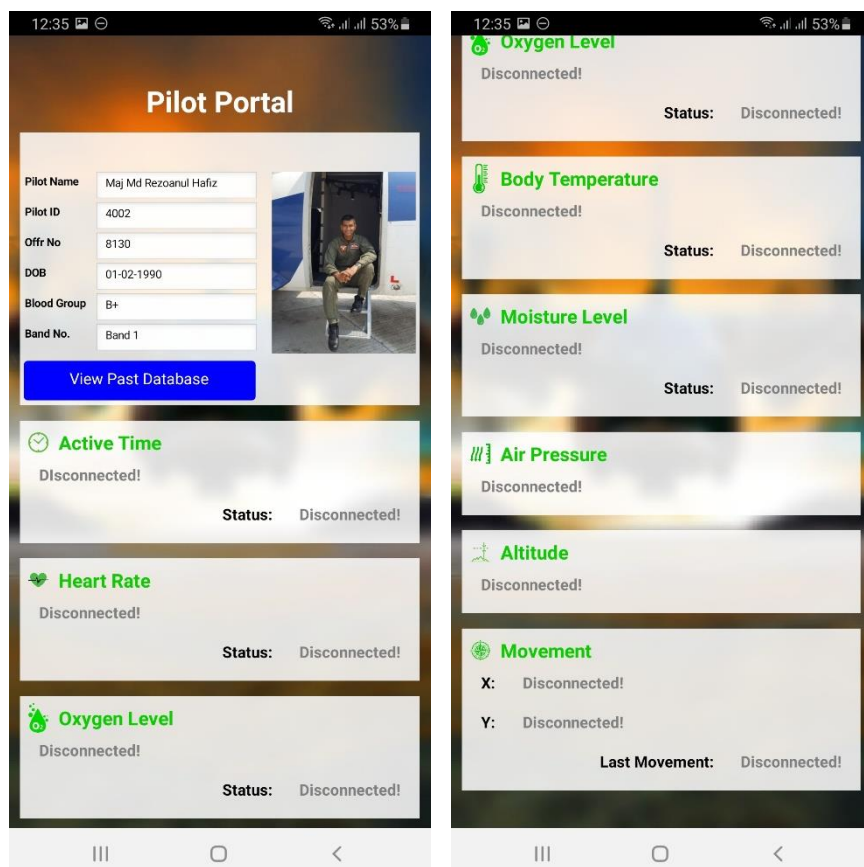

3.2   Admin Page.

In this page we need to put the authorized ID and password for an authorized log in as admin.

3.2.1 Check Status.
The ATC can access to monitor the current and previous state of the
pilot.

3.2.2 Database.

The ATC can access to monitor the current and previous state of the pilot.

3.3 Registration.

Only the authorized pilot can have the registration access. During registration the pilot gives necessary information which we store in firebase.
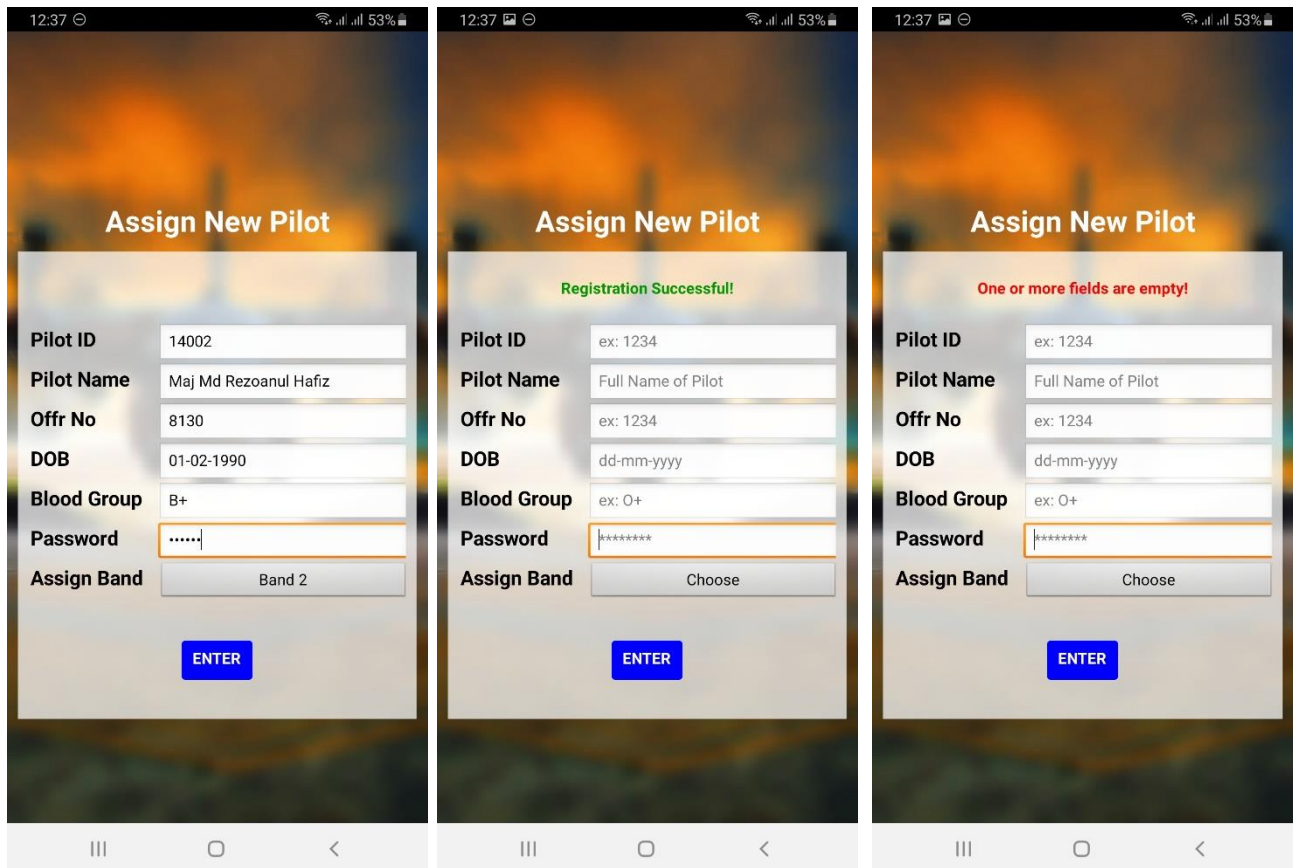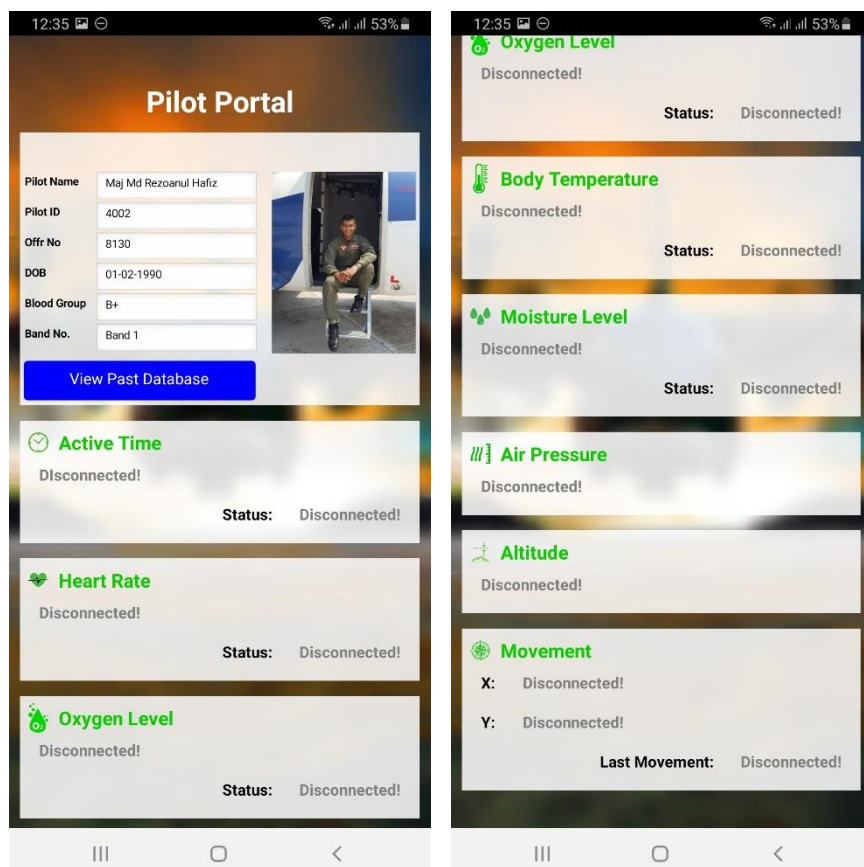
3.4 Pilot Page.

In this page we need to put the authorized ID and password for an authorized log in as pilot.
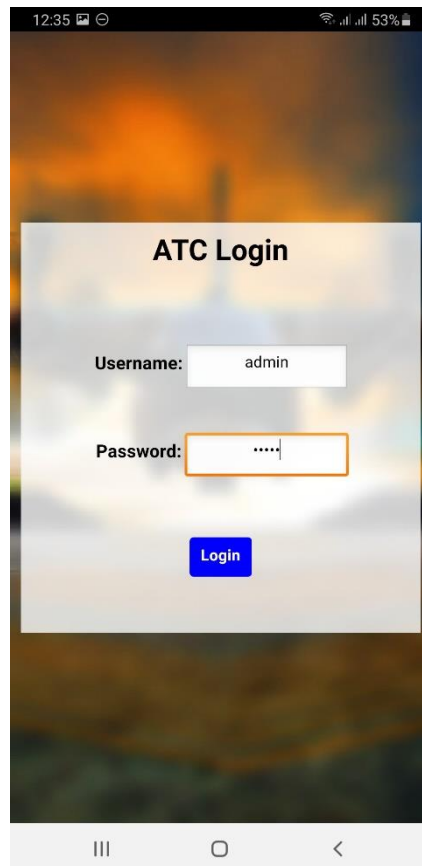
3.4.1 Check Status.
The Pilot can his health status when connected to internet.

3.4.2 Database.

The ATC can see his health status during previous flights.