INTEGRATED DESIGN PROJECT

CSE-460

System Requirement Specification (SRS)

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. Introduction.

The name of our project is "Health Monitoring System of a Pilot Using IoT Wrist Band". Our project includes a digital wristband, an ATC base and a mobile app. In this project, data analysis will be done by taking reading from the smart wristband of the pilot. By analyzing the data, signal will be given based on the health condition of pilot flying the aircraft. There will be given standard readings comparing with which analysis will be done to generate necessary alert for the pilot. Physician sitting in the ATC base can also provide necessary health advice to the pilot using the default communication method for aircraft.

1.1 Purpose.

The purpose of this SRS is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's target audience, its user interface, hardware and software requirements. The target audience of the product is mainly pilots flying the aircrafts. It will provide the physical health status of that individual pilot to the ATC tower for analysis and dispatch of necessary actions.

1.2 Comparative Discussion.

Mental state and physical state of a human being is inter related. Now-a-days different sensors are used for measuring different physical state factors of human body. And different smart watches are there for this purpose. So, we consider that the pilots are one of the valuable assets in human society and we need to have something that will help to lessen the risk of pilot, passenger and valuables. It became a motivation for us to apply that smart watch technology in our pilot society while on flight. Accidents of aircraft may take place due to discomfort faced by the pilots. They need proper monitoring, guideline and alert. Few scenarios of mishaps are described below:

a. Directorate of air operations need to know the condition of a particular pilot during flying. Because he needs to take actions for poor performance in flying in order to safe guard the valuables. He knows that the pilot is going to be grounded because of his poor performance in flying. But his instructors think that his poor flying is because of his over excitement during flying. They need to figure it out whether this happens for his overexcitement or not. Basing on this he will form a report where he has to mention whether the pilot is fit for flying in future or will be grounded. So, Directorate of air operations is seeking the reports of "Health Monitoring of a Pilot using IoT wristband" system to decide that the pilot's health is the only reason for his poor performance in flying or there is any other.

b. Flight Surgeon of 103 ATTU, Flight Lieutenant X checks pilot's health before flying. But he knows that during flying it may not remain safe. Human mental and physical state changes due to change in environment. So, if he could know the condition of a pilot's health during flying, he could save lives as well as aircraft. For continuous monitoring of a pilot's health during flying and take necessary actions, he is seeking the "Health monitoring system of a pilot using IoT wristband" and will providing necessary advise if any adversity occurs.

1.3 Scope.

Smart wrist band is a single platform application. The application will be interfaced with the software. When a pilot goes onto the flight, the wearable devices will collect data and transfer the data to the ATC base. Again the data will be sent to an android device from ATC base. Each pilot or user will have an entry of data and a registration process. By which in firebase where his previous and current physical state information will be stored. Moreover using this application, an authorized person will be able to keep track of the health state of a pilot to provide necessary alert. By acknowledging the alert and advices pilot can move himself to a stable state.

1.4 References.

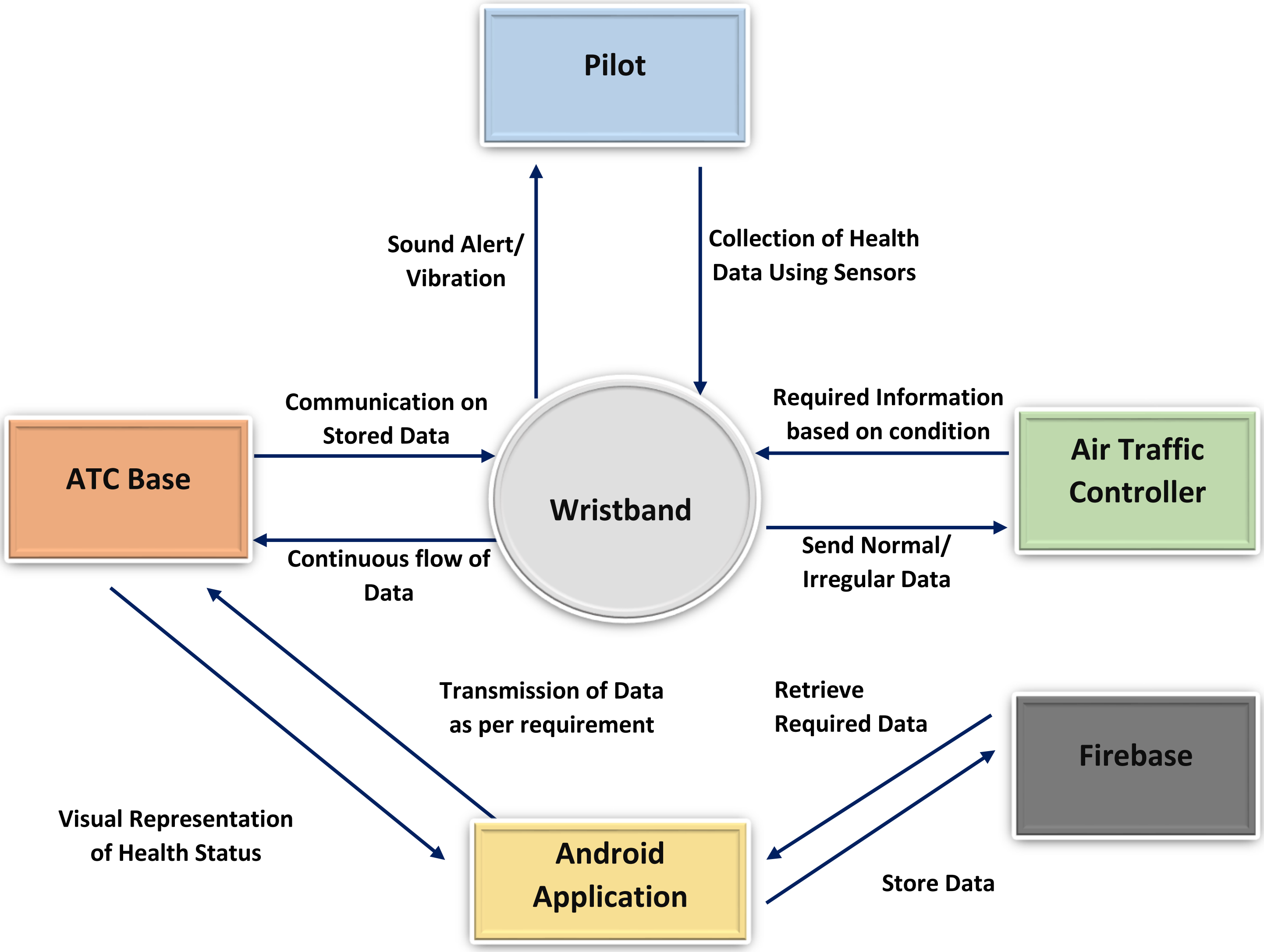
2. Overall Description.

This section with the system environment and the functional requirements of the system. The relationship between the stakeholders and the entire system is also briefly mentioned and some non-functional requirements are highlighted at the end.

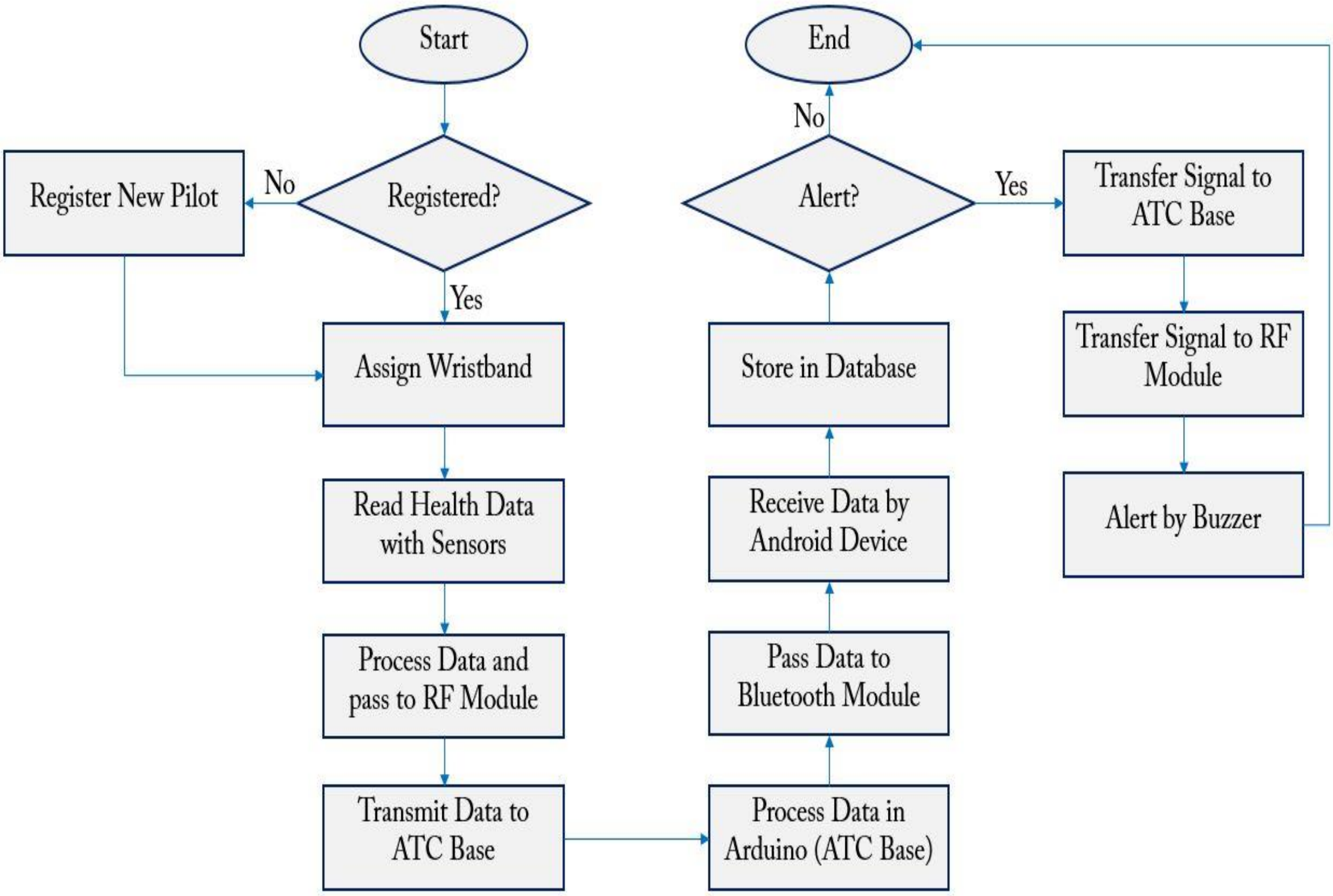
2.1 System Environment.

Here we include graphical system models showing the relationships between the system components and the system and its environment.

2.1.1 Context Diagram.


Pilot
Sound Alert/ Vibration
Collection of Health Data Using Sensors
ATC Base
Communication on Stored Data
Continuous flow of Data
Wristband
Required Information based on condition
Send Normal/ Irregular Data
Air Traffic Controller
Transmission of Data as per requirement
Visual Representation of Health Status
Retrieve Required Data
Store Data
Firebase
Android Application

2.1.2 Activity Diagram.


Start
Registered?
No
Register New Pilot
Yes
Assign Wristband
Read Health Data with Sensors
Process Data and pass to RF Module
Transmit Data to ATC Base
Process Data in Arduino (ATC Base)
Pass Data to Bluetooth Module
Receive Data by Android Device
Store in Database
Alert?
No
End
Yes
Transfer Signal to ATC Base
Transfer Signal to RF Module
Alert by Buzzer

2.2 Functional Requirements Specification.

Use case diagrams and sequence diagrams are used for interaction modeling. Use cases were developed originally to support requirements elicitation and each use case represents a discrete task that involves external interaction with a system. In this system we will see two use case scenarios. A sequence diagram shows the sequence of interactions that take place during a particular use case or use case instance. Here for the first use case we will see a sequence diagram in figures and second use case the sequence diagram is in figures. The objects and actors involved are listed along the top of the diagram, with a dotted line drawn vertically from these. Interactions between objects are indicated by annotated arrows.

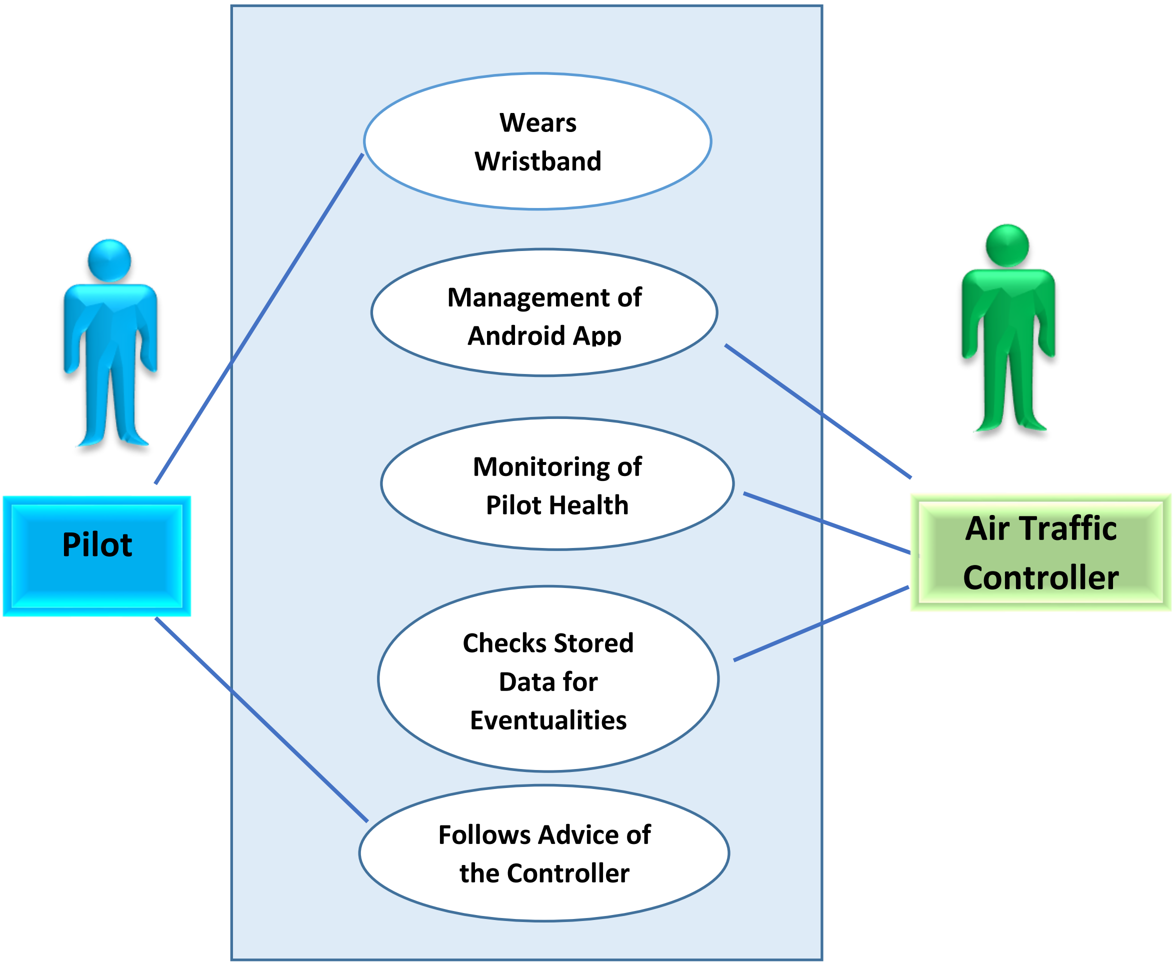
2.2.1 Feature List.

The list is given below:

a. To integrate wearable devices to the pilots.
b. To collect data from those wearable devices and send those to ATC base.
c. Display data in ATC base.
d. Send Data to android device via Bluetooth module.
e. Data analysis and store data to firebase.
f. Data retrieval from firebase.
g. Generate alert for pilot by buzzer in wrist watch.

2.2.2 User Case Diagram.

a. Normal and Critical Scenario.

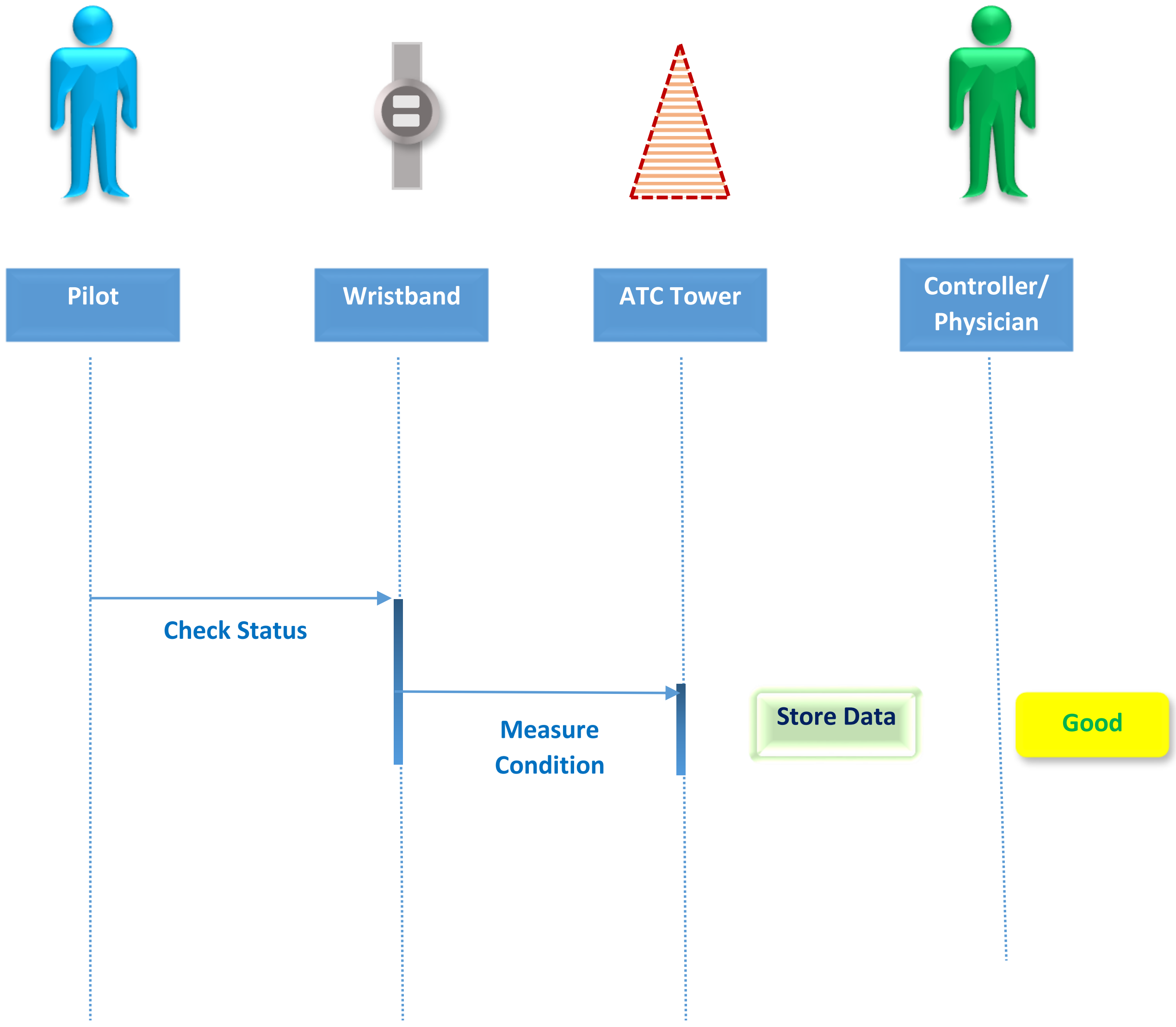

Wears Wristband
Management of Android App
Monitoring of Pilot Health
Checks Stored Data for Eventualities
Follows Advice of the Controller
Pilot
Air Traffic Controller

Use Case	**Normal Scenario**
Actors	Pilot, Air Traffic Controller
Description	A pilot wears the wristband and the readings of the sensors are seen in the android app by the air traffic controller. From the standard readings, controller can tell whether the condition of the pilot is fit for flying or not. The controller gives necessary advice by monitoring pilot's health.
Data	Readings from the sensors
Stimulus	Data collection, command issued by Controller
Response	Do nothing
Comments	a. The ATC tower should have a good communication with pilot. ii. The android app should have accurate data.

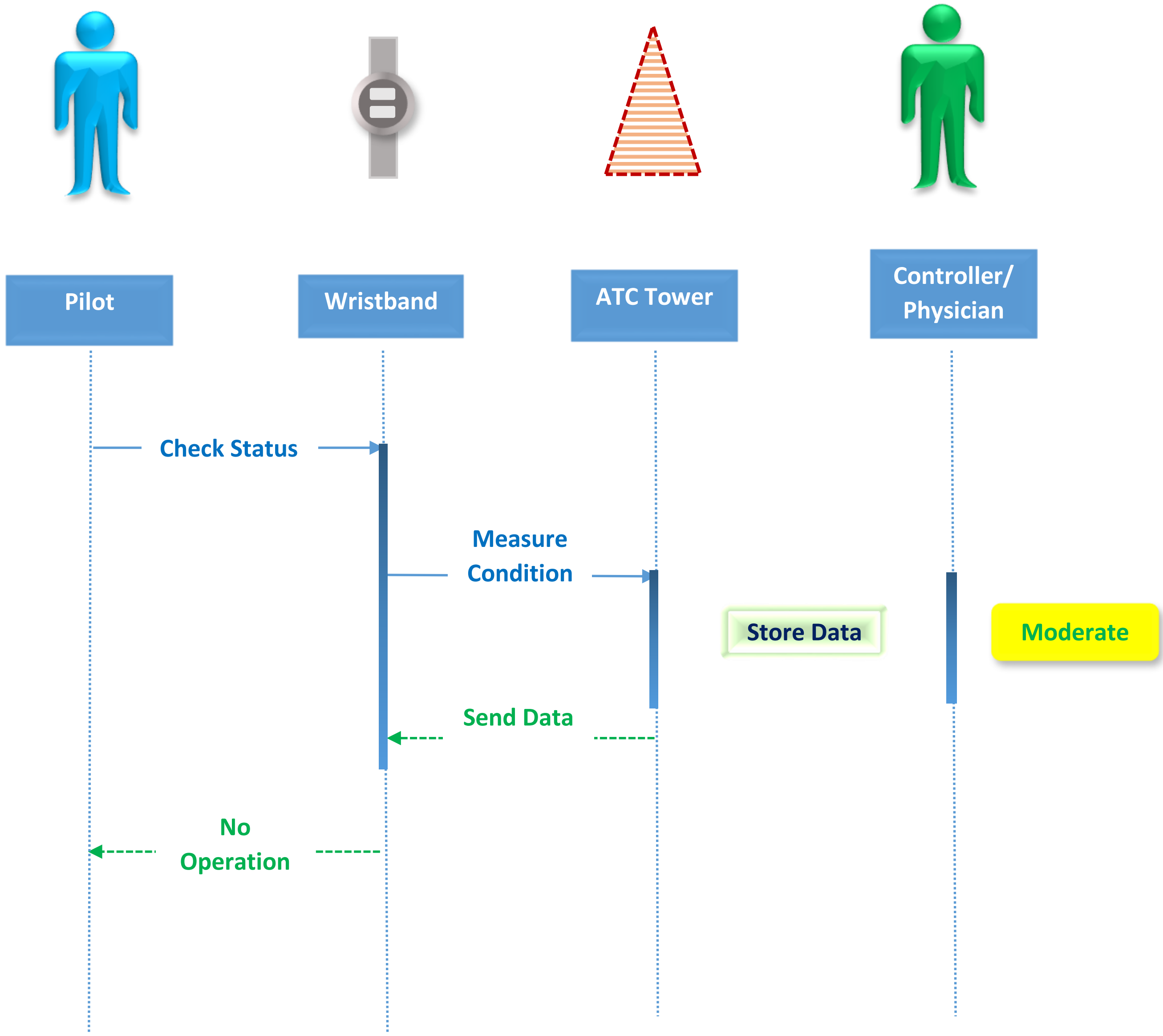
Use Case	**Critical Scenario**
Actors	Pilot, Air Traffic Controller
Description	A pilot wears the wristband and the readings of the sensors are seen in the android app by the air traffic controller. From the standard readings, controller can tell whether the condition of the pilot is fit for flying or not. The controller gives necessary advice by monitoring pilot's health.
Data	Readings from the sensors
Stimulus	Data collection, command issued by Controller
Response	Alert, Eject/ Medication advice
Comments	i. The ATC tower should have a good communication with pilot. ii. The android app should have accurate data.

2.2.3 Sequence Diagram.

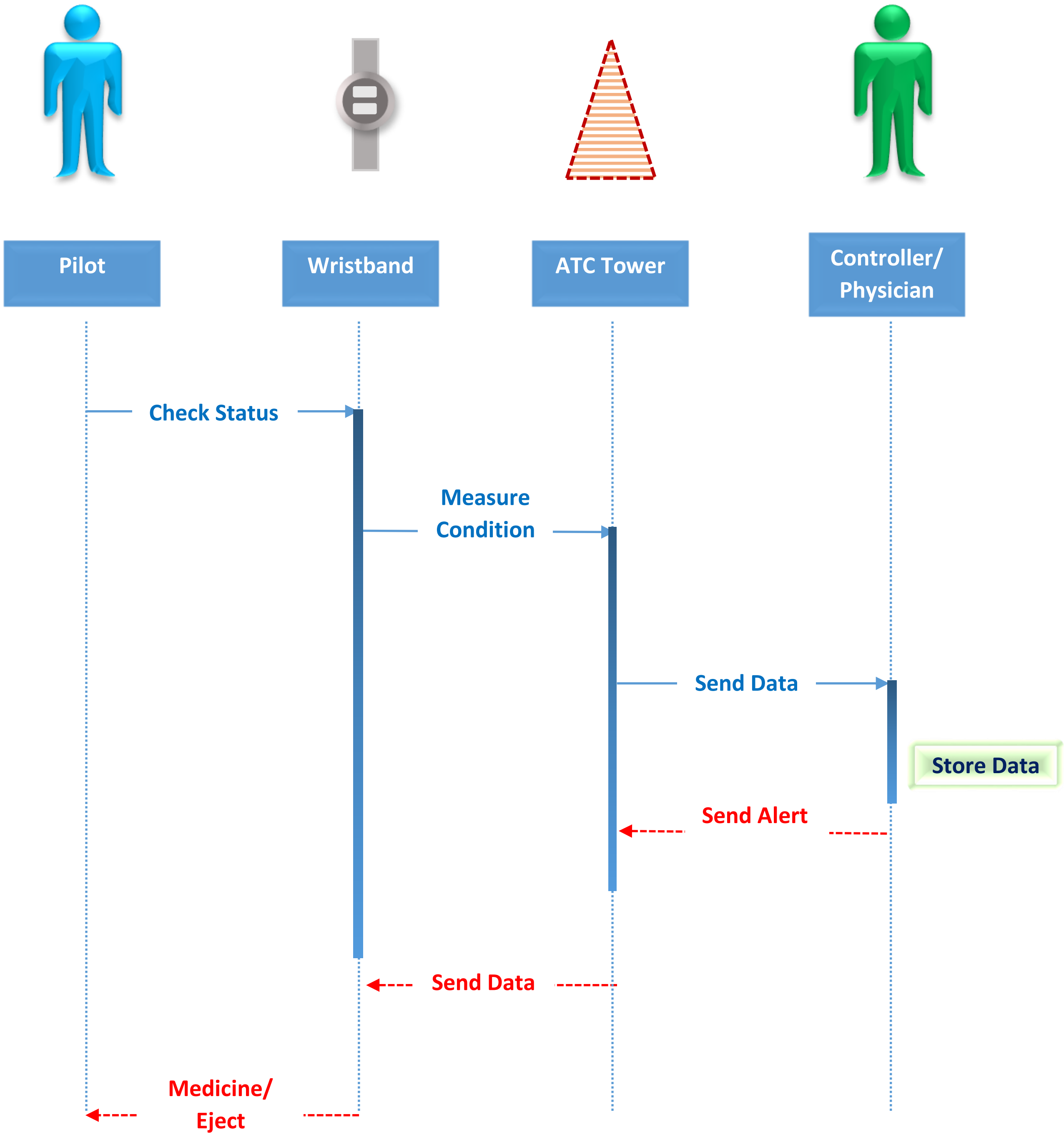
a. Normal Scenario.



b. Moderate Scenario.



c. Critical Scenerio.



2.3 User Characteristics.

a. Admin: Admins are responsible persons sitting at ATC Base to monitor the pilot health.

They provide the user ID to all the users. They need to enter all data to the system for automated system flow.

b. Pilot: Pilots just wear the wrist watch and work according to the instruction from ATC base when in adverse condition.

2.4 Non-functional Requirements.

2.4.1 Performance Requirements.

2.4.2 Safety Requirements.

2.4.3 Security Requirements.

2.4.4. Software Quality Attributes.

3. Specific Requirements.

3.1 External Interfaces.

3.1.1 Human Interface.

User characteristics define two types of user for human interactions. Everyone from each panel will be provided different types of user IDs and passwords from us which can be changed later. Admin password and ID will be fixed and only the ATC base authorized person can act as admin. There is a common home page for every user. After entering the correct id and password they are taken to their corresponding respective UI.

3.1.2 Software Interface.

The main purpose of front-end development is to make the whole system easily accessible and usable for all the stakeholders. There are dashboards of the app for two types of users (Admin and Pilot).

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:

a. 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.

b. 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.

a. 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.

b. 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. 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. Despite its low-level capabilities, the language was designed to encourage cross-platform programming. A standards-compliant C program written with portability in mind can be compiled for a wide variety of computer platforms and operating systems with few changes to its source code.

3.1.3 Hardware interface.

Sensors are used to send data to Arduino Uno of wrist watch then it is transferred through RF Module (NRF24L01) to the antenna of ATC base tower. Then the data are displayed after being processed by Arduino mega on a serial display. Then the data are again transferred to android device by Bluetooth module (HC05). Then the data is stored in firebase.

3.1.4 Communication Interface.

We need internet connection to store data and retrieve data from firebase. We’ve used a RF and a Bluetooth module to transfer data in the whole system.

a. System Architecture.

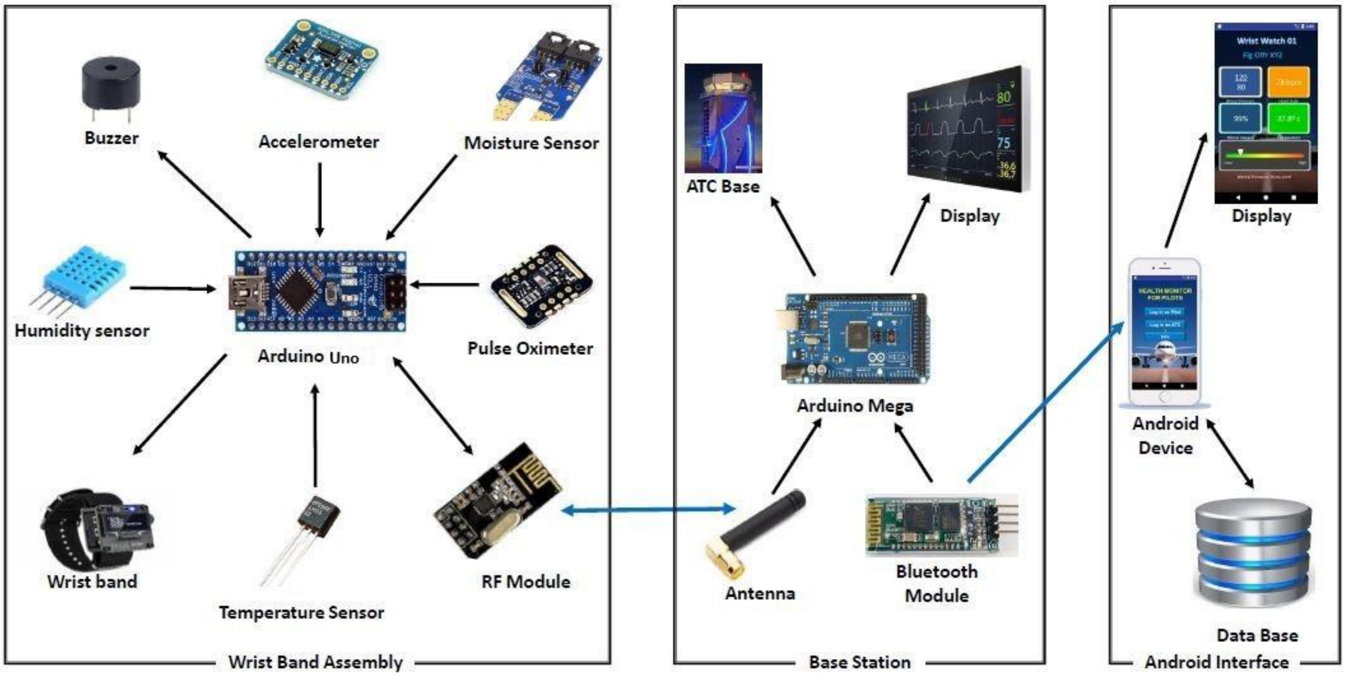


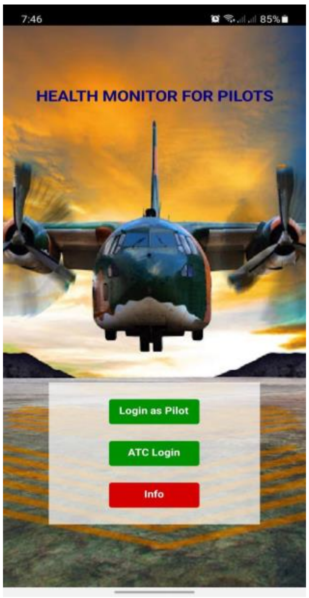
Fig: System Architecture

4. <u>UI Design</u>.

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

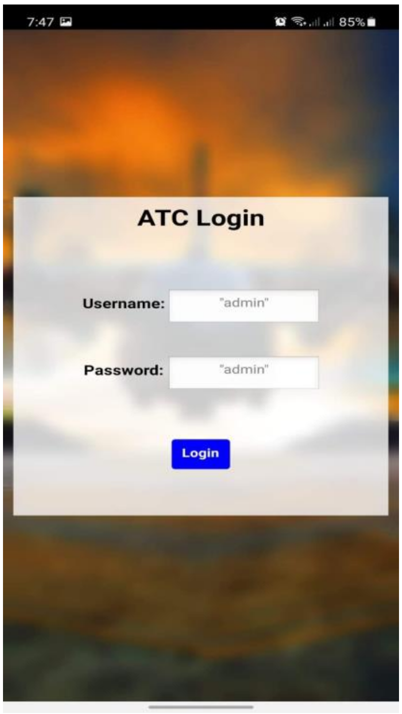
4.1 **Home Page**.

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.



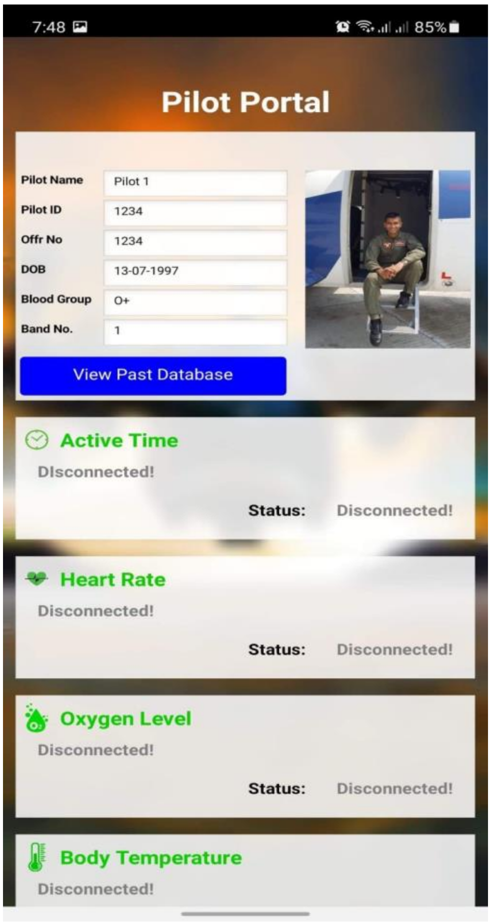
4.2 **Admin Page**.

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



4.2.1 Check Status.

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



Pilot Portal

Pilot Name: Pilot 1
Pilot ID: 1234
Offr No: 1234
DOB: 13-07-1997
Blood Group: O+
Band No.: 1

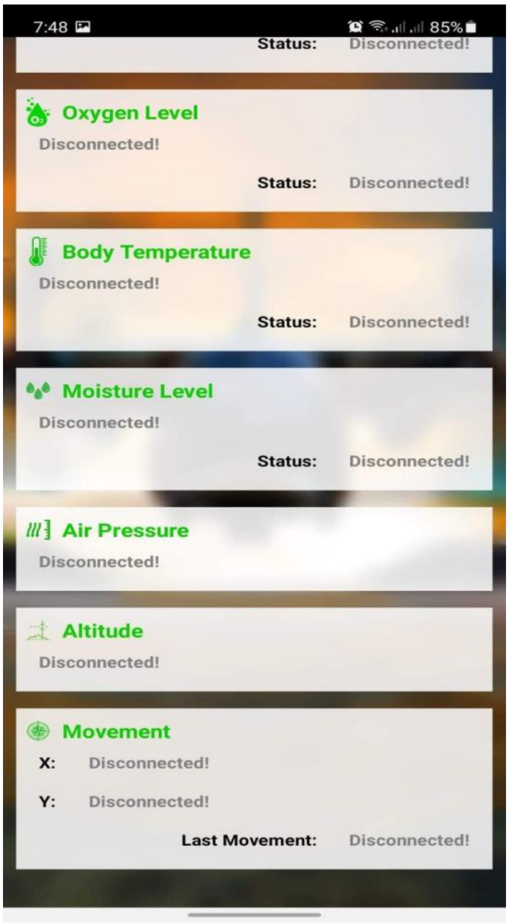
View Past Database

Active Time
Disconnected!
Status: Disconnected!

Heart Rate
Disconnected!
Status: Disconnected!

Oxygen Level
Disconnected!
Status: Disconnected!

Body Temperature
Disconnected!



Status: Disconnected!

Oxygen Level
Disconnected!
Status: Disconnected!

Body Temperature
Disconnected!
Status: Disconnected!

Moisture Level
Disconnected!
Status: Disconnected!

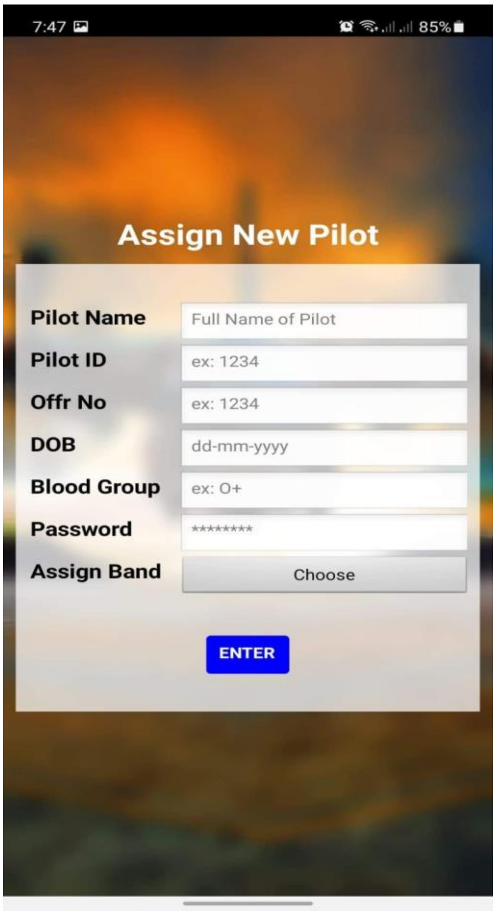
Air Pressure
Disconnected!

Altitude
Disconnected!

Movement
X: Disconnected!
Y: Disconnected!
Last Movement: Disconnected!

4.3 **Registration.**

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



Assign New Pilot

Pilot Name	Full Name of Pilot
Pilot ID	ex: 1234
Offr No	ex: 1234
DOB	dd-mm-yyyy
Blood Group	ex: O+
Password	********
Assign Band	Choose

ENTER