# Dreamline Project Requirement Summary

**Group 26- Cole Pearson, Nausherwan Tirmizi, Brian Kopec, Harshal Patel**

Dreamline as a product has many requirements that must be fulfilled before it can be released to the public. These requirements will ensure that the product that is released provides the client with exactly what they want from the product. These requirements will give the developers of the product a very detailed idea of what the product should look like and what requirements the product should follow. These requirements are drawn from use cases that the product may come across. To determine these, we have a couple of external actors that will do some action that will initiate the use cases. Some of these actors include the airlines, users, smart devices, the Dreamline calculator and the website.

The use cases start with a user entering in an itinerary, which results in the Dreamline calculator to find the cheapest and most comfortable flight. This is calculated into a Dreamline score and the best scores and flight information is displayed on the website or mobile application. The user can then purchase the ticket which will trigger the system to enter payment information and the transaction completes the purchase by buying the ticket from the airline and gives the user a confirmation.

On the day of their scheduled flight that the user bought through Dreamline, the mobile application will start collecting data starting at take off time as well and finish collecting data a few hours after the flight lands. Sensors on smart devices such as smartwatches and phones, collects data and looks out for any signs of discomfort, coming from sensors such as the accelerometer, oximeter, sleep tracker, and heart rate monitor.

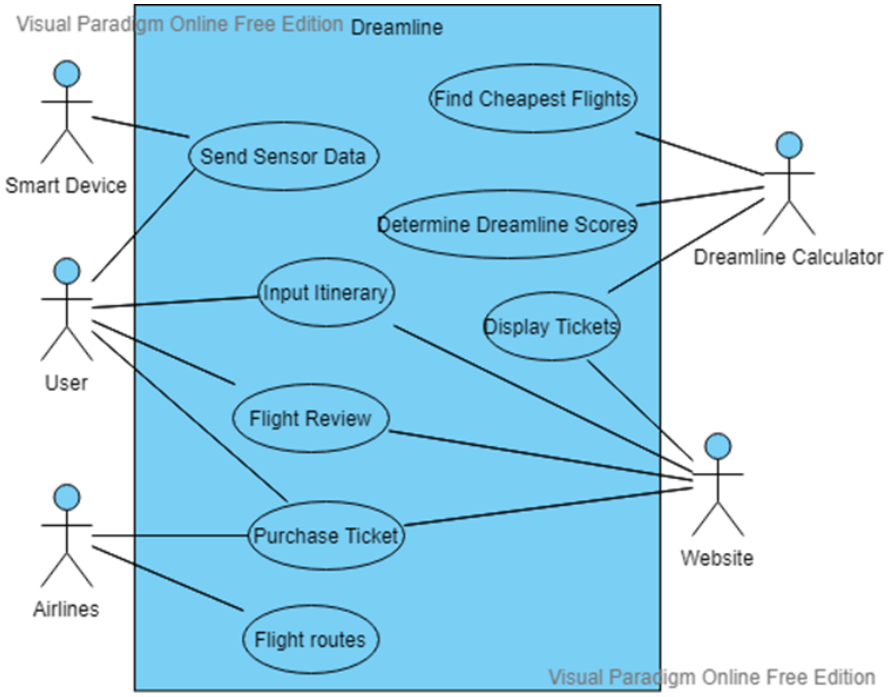
Once the flight is over and the data finishes collecting data, the data is uploaded to the cloud server. Before the data is sent, the take off time and landing time is verified through flight API's and data outside of that time frame is cut out. The user can also rate the different aspects of their flight. This is compared with the sensor data, to determine whether the flight was uncomfortable or not and the Dreamline score is updated for other users to see.

These are some of the use cases that help us determine what requirement must be set in place before the developers can create the system. Attached below is a Use case diagram that illustrates some of these use cases.

The requirements that are outlined in the report includes, Functional, Data, Performance, Dependability, Maintenance, Supportability, Security, Usability, Humanity, Look and Feel, Operational, Environmental, Cultural, Political, and Legal. It is important to note that there were not any applicable requirements for Cultural, Political and Audit requirements.

The functional requirements mainly go into detail about how the specific use cases must be accomplished. For example, FUNC-2(Ticket Lookup) states that the system must be able to make API calls for ticket availability. These functional requirements specify any requirements and constraints that must be followed in terms of the functionality of the system. There are also requirements about the data and how it should be collected and stored. For example,

DATA-1(Sensor Data) is a requirement stating that the system must be able to store sensor data from smart devices.




There are also many non functional requirements that must be taken into consideration. In terms of performance many of the requirements essentially state that the use of the system must be smooth and lag free. The data that is collected must also be precise and accurate which is why we have a requirement stating that the data should be precise to 5 decimal places and collected every 10ms to provide accurate data (PAR-1 and PAR-2). There are some capacity requirements stating that the cloud should be able to store a large amount of data and the application itself must not be very large. There are also maintenance and supportability requirements that state there should be some form of tech support and bug evaluation. Importantly, there are also security requirements which state who has access to user data and that the data must be protected. Data must also be verified before it is stored, preventing any malicious intent sent to the system. Also, Usability and Humanity requirements must be taken into consideration, which determine how accessible and easy to use the application must be. For example, there should be little to no training required to use the application. Look and Feel requirements state that the product should look professional and allow for light and dark modes. Operational and environmental requirements state how compatible the systems must be. Finally the Product must comply with legal requirements such as data collection must be from consent of the user.

There are many other smaller requirements reported in this section and are equally important as the ones listed. These requirements will allow the developers to start creating a design for the system with a clear view of what the client wants.