Firstly, the user has the opportunity to login to the system by accessing the XML files. This creates a stable way to accessing files in a structured way that can be viewed outside of the system for exporting purposes and diagnostic. Another advantage of the way log in is used is the fact it searches through a very easy to read file, which makes the efficiency of the program very fast and uses very little memory reading through the XML file. It iterates through as if it is a big array which allows it to run in the background and memory to be targeted where it needs to be.

A user can also be registered which makes the program into a slightly more user friendly design, as it allows direct input from the user into the system. This adds to the XML file directly which the login function reads from which shows the fact that the system is well integrated with each other. It stores static information and an admin can go through and change the permissions of a user which will open up more options in the system’s menu and allow higher privileges.

A user can book a flight that is valid at any time. This is done by reading a separate XML file with seat allocations into the system and allowing the customer to pick a seat depending on that. This then adds the user to the flight information XML document, which also adds to how integrated the system is. Bookings can be altered by the admins. This was a good example of implementation as it takes different types of information and displays it into the same file without causing issues and altering efficiency.

By having a time board in our software, it allows the implementation of our Quicksort and Queue algorithms that import times of flights, sorts them appropriately and then places them onto a Queue which would, in the real world, be outputted to a separate TV screen as you see in many airports. This is good as it uses a very complex sorting algorithm efficiency and then the queue would be as you’d see on a real life situation so shows the scope of our program. This could be better in some ways due to the fact Quicksort is one of the slowest and hardest to implement, so in moving the system around it could cause issues that we hadn’t foreseen.

The plane data that the software receives is also kept inside of an XML file that incorporates the different rows and columns of the plane and also has storage capacity information and model types. This is imported when an admin creates a flight using that plane data and it is then loaded in and creates a text file that holds seat information.