The Bluebrook Airport system utilises an object-oriented approach in its code design. The entire concept of the program revolves around objects interacting with one another, such as users, flights and planes, making object-orientated programming ideal and efficient. Additionally, in a system like an airport booking system, maintainability is something that should always be thought about as the travel industry evolves and object-oriented design lends itself very well to this – adding functionality to objects is a much easier process than going through a procedural based program.

One useful feature of object-oriented programming is inheritance, which is something that has been utilised with the User class. There are several methods that all users should be able to perform, such as update their details, but additionally each different type of user will have different methods that they want to invoke. For example, regular accounts exclusively would want to view their flight details and their frequent flier points whereas staff accounts would want to be able to print off flight chart details and manually override flight details if required. None of these things should be accessed by the other class, which means we can make use of inheritance to make two subclasses, UserRegular and UserStaff. This is flexible and helps with maintenance, as we can make specific user subclasses for each staff role to allow them to do perform their own methods without having to rewrite all the shared methods inherited from the User class individually.

Another feature of an object-oriented system is that allows for easy maintainability is that it is much easier to model the system through UML (Unified Modelling Language). This not only has benefits in the design process and initial building of a system, but a well-designed and maintained model makes it a lot easier to sustain and further develop already existing systems because of how easily read these models are and how clearly they show all the relations between classes.