# Comprehensive Report for Task 4

## General Introduction

The objective of Task 4 is to develop a set of classes representing an airport system. This system includes an `Airport` class, a `Flight` class, and an interface `LegalEntity`. The purpose is to model real-world entities in a software context, allowing for operations such as adding and removing flights, and saving and loading flight information from a file.

## Functionality and Goal of Each Class

### Airport Class

\*\*Functionality:\*\*

- The `Airport` class is designed to manage a list of flights and handle file operations to save and load flight information.

- It implements the `LegalEntity` interface, which mandates methods for getting the address and VAT number of the airport.

\*\*Key Methods:\*\*

- `addFlight(Flight f)`: Adds a flight to the list of flights.

- `deleteFlight(Flight f)`: Removes a flight from the list of flights.

- `saveFlightsToFile()`: Saves the list of flights to a file using serialization.

- `loadFlightsFromFile()`: Loads the list of flights from a file using deserialization.

\*\*Attributes:\*\*

- `List<Flight> flights`: A list to store flight objects.

- `String address`: The address of the airport.

- `String vatNumber`: The VAT number of the airport.

### Flight Class

\*\*Functionality:\*\*

- The `Flight` class models a flight with attributes such as flight number, departure and arrival airports, and departure and arrival times.

\*\*Key Methods:\*\*

- Getters and setters for each attribute: `flightNumber`, `departureAirport`, `arrivalAirport`, `departureTime`, `arrivalTime`.

\*\*Attributes:\*\*

- `String flightNumber`: The flight number.

- `String departureAirport`: The airport from which the flight departs.

- `String arrivalAirport`: The airport at which the flight arrives.

- `LocalDateTime departureTime`: The departure time of the flight.

- `LocalDateTime arrivalTime`: The arrival time of the flight.

### LegalEntity Interface

\*\*Functionality:\*\*

- The `LegalEntity` interface defines two methods that must be implemented by any class that represents a legal entity.

\*\*Methods:\*\*

- `getAddress()`: Returns the address of the entity.

- `getVatNumber()`: Returns the VAT number of the entity.

## Conclusion

This report provides a comprehensive overview of the classes developed in Task 4. The `Airport` class manages flight operations and file storage, the `Flight` class models individual flights, and the `LegalEntity` interface ensures that legal entity details are provided. Together, these classes form the foundation of an airport system, capable of managing flight information efficiently.