```
#include <iostream>
#include <string>
#include <vector>
#include <stdexcept>

using namespace std;

// Base class for Person
class Person {
protected:
    string name;
    int age;

public:
    Person(string n, int a) : name(n), age(a) {}
    virtual void displayInfo() {
        cout << "Name: " << name << ", Age: " << age << endl;
    }
};

// Derived class for Passenger
class Passenger : public Person {
private:
    string passportNumber;

public:
    Passenger(string n, int a, string pNum) : Person(n, a), passportNumber(pNum) {}
    void displayInfo() override {
        cout << "Passenger Info - Name: " << name << ", Age: " << age << ", Passport Number: "
<< passportNumber << endl;
    }
};

// Flight class
class Flight {
private:
    string flightNumber;
    string destination;
    vector<Passenger> passengers;

public:
    Flight(string fNum, string dest) : flightNumber(fNum), destination(dest) {}

    void addPassenger(Passenger p) {
```

```
        passengers.push_back(p);
    }

    void displayFlightInfo() {
        cout << "Flight Number: " << flightNumber << ", Destination: " << destination << endl;
        for (auto &p : passengers) {
            p.displayInfo();
        }
    }

    string getFlightNumber() {
        return flightNumber;
    }
};

// Booking class with Exception Handling
class Booking {
private:
    vector<Flight> flights;

public:
    void addFlight(Flight f) {
        flights.push_back(f);
    }

    void bookSeat(string flightNumber, Passenger p) {
        for (auto &flight : flights) {
            if (flight.getFlightNumber() == flightNumber) {
                flight.addPassenger(p);
                return;
            }
        }
        throw invalid_argument("Flight not found");
    }

    void displayBookings() {
        for (auto &flight : flights) {
            flight.displayFlightInfo();
        }
    }
};

// Main function demonstrating the use of the system
int main() {
```

```
try {
    Booking bookingSystem;

    // Adding flights
    Flight flight1("BA101", "Kenya");
    Flight flight2("BA202", "London");
    Flight flight3("BA303", "USA");
    bookingSystem.addFlight(flight1);
    bookingSystem.addFlight(flight2);
    bookingSystem.addFlight(flight3);

    // User inputs
    string name;
    int age;
    string passportNumber;
    string flightNumber;
    cout << "______welcome to Frosty Airline services______" << endl;
    cout << "Enter your name: ";
    getline(cin, name);

    cout << "Enter your age: ";
    cin >> age;
    cin.ignore(); // To ignore the newline character after age input

    cout << "Enter your passport number: ";
    getline(cin, passportNumber);

    cout << "Available flights: " << endl;
    cout << "1. BA101 to Kenya" << endl;
    cout << "2. BA202 to London" << endl;
    cout << "3. BA303 to USA" << endl;
    cout << "Enter flight number to book: ";
    getline(cin, flightNumber);

    Passenger passenger(name, age, passportNumber);

    bookingSystem.bookSeat(flightNumber, passenger);

    // Displaying all bookings
    bookingSystem.displayBookings();
}
catch (const exception &e) {
    cerr << "Error: " << e.what() << endl;
}
```

```
    return 0;
}
```