NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

PESHAWAR CAMPUS

SEMESTER PROJECT (OOP)

PROJECT REPORT

SUBMITTED TO: MAM SARA REHMAT

SUBMITTED BY: ALISHBA TARIQ

ROLL NO: 22P-9112

SECTION: BSE-2A

Project Name:

Airport Management System

Team Members:

1. Alishba Tariq (22P-9112)
2. Hasnain Saleem (22P-9123)

1. Purpose:

Saving passenger, employee, flight, and other records manually is very time-consuming, difficult, and exhausting because it takes a long time. We are developing a program that is menu-driven, user-friendly, and simple to record in order to solve this issue.

2. Objectives

The main objectives of the Airport Management System project are as follows:

- Show airport details to user including history and map
- Book tickets for passengers with unique passenger ids
- Show, store and update employee data stored in files
- Show details of international and domestic flights with their arrival and departure stored in separate files
- Get feedback from user and store it in a file
- Allow only administrator having the unique admin id to use the administrative features

3. System Design

The system follows an object-oriented design approach, utilizing various classes and their interactions.

- **Static_password**: static Constant to check the credibility of user to ensure secure before using administrative features.

Classes:

The key classes in the system are as follows:

- **Employee:** Class inherited into children contains basic data members like id, name etc .
- **Airline_Employee:** publicly inherited from Employee class.
- **Airline:** Consist of functions related to airline employees.
- **Airport_Employee:** publicly inherited from Employee class.
- **Airport:** Consist of functions related to airport employees.
- **Ticket_Booking:** class to book ticket for passengers .
- **Passenger:** contains object of Ticket_Booking class as data member (includes file handling).
- **Flight_Schedule:** stores basic data related to flight details such as date, time, city, flight code.
- **Flight_Type:** objects of Flight_Schedule created as data members to store data (involves file handling for storage).

Struct:

- **Airport_Details:** struct to print name, history and map of the airport.

Functions:

Display(): display for the airport details structure.

Feedback(): standalone function for the user to enter feedback regarding the airport.(uses file handling to store feedbacks).

4. Features and Functionalities

The Airport Management System provides the following features and functionalities:

- **Airport_Details:** show the name, history and map of the airport for user's ease.
- **Employee:** consist basic data members; id, name, salary and designation for employees.
- **Airline_Employee:** publicly inherited from Employee class. It contains all data related to the employees of the airline.
- **Airline:** Consist of operations (store, show, add, delete and update) to be performed on airline employee
- **Airport_Employee:** publicly inherited from Employee class. It contains data related to employees of the airport base.
- **Airport:** Consist of operations (store, show, add, delete and update) to be performed on airport employee
- **Ticket_Booking:** class to book ticket for passengers by assigning them unique ticket ids.
- **Passenger:** Composition from Ticket_Booking as data member in order to perform operations (book, update, search, delete, backup, print) on ticket booking for passenger. (Includes file handling)
- **Flight_Schedule:** containing basic data related to flight details such as date, time, city, flight code.
- **Flight_Type:** objects of Flight_Schedule created as data members to store data as arrival and departure for each

flight type (international and domestic). It also performs operations such as add, delete, update on this data (involves file handling for storage).

- **Feedback():** function to take feedback from the user and store it in file via handling

5.Implementation

The number and implementation of classes as well as the class diagram were modified as per the requirements of the project .

The Airport Management System is implemented using the C++ programming language and follows the object-oriented paradigm.

The system utilizes

- classes (setters, getters)
- inheritance (single + multiple, public, protected)
- Polymorphism (virtual classes and methods, method overriding)
- function overloading
- dynamic memory (dynamic arrays)
- default constructors
- Parameterized constructors
- copy constructor
- pointers
- destructors
- file handling
- vectors
- template class

concepts to achieve the desired functionalities.

- Our code also followed good programming practices
- Involved encapsulation and separation of concerns

6. User Interface

The user interface for the Airport Management System can be implemented using console-based input/output and it is menu-driven.

7. Testing

Testing is an essential phase of software development to ensure the correctness and reliability of the system. The Airport Management System went through rigorous testing of various types , including unit testing, integration testing, and system testing.

8. Conclusion

The project demonstrates the application of object-oriented programming principles in developing a comprehensive system for managing airport operations ensuring the security too.
