**SKY WINGS AIRLINES**



**SESSION 2023 – 2027**

**Submitted by:**

Abdul Hanan 2023-CS-105

**Supervised by:**

Miss Maida Shahid

**Course:**

CSC-102 Programming Fundamentals

Department of Computer Science

**University of Engineering and Technology**

**Lahore Pakistan**

Table of Contents

[ Short Description of my Project: 5](#_Toc153911621)

[ Contribution towards CS: 6](#_Toc153911622)

[ Users of Application: 6](#_Toc153911623)

[ Admin: 6](#_Toc153911624)

[ Customer: 6](#_Toc153911626)

[ Manager: 6](#_Toc153911627)

[ Functional Requirements: 6](#_Toc153911628)

[Admin 7](#_Toc153911629)

[Customer 8](#_Toc153911630)

[Manager 8](#_Toc153911631)

[ Wireframes: 9](#_Toc153911632)

[Header: 9](#_Toc153911633)

[ Login Screen: 9](#_Toc153911634)

[9](#_Toc153911635)

[ Sign in Screen: 10](#_Toc153911636)

[ Sign Up screen: 10](#_Toc153911637)

[ Admin Interface: 11](#_Toc153911638)

[11](#_Toc153911639)

[ Add employee: 11](#_Toc153911640)

[ Remove employee: 12](#_Toc153911641)

[ Add Airlines: 12](#_Toc153911642)

[ Employee list: 13](#_Toc153911643)

[ Discount allocation: 13](#_Toc153911644)

[ Announcement to employees: 14](#_Toc153911645)

[ Salary management: 14](#_Toc153911646)

[ Percentage of Discount: 15](#_Toc153911647)

[ Price allocation: 15](#_Toc153911648)

[ Price allocation for services: 16](#_Toc153911649)

[ Customer Reviews: 16](#_Toc153911650)

[ Revenue Interface: 17](#_Toc153911651)

[ Manager Menu: 17](#_Toc153911652)

[ Schedule of Tickets: 18](#_Toc153911653)

[ View selected airlines: 18](#_Toc153911654)

[ View Announcemet from admin: 19](#_Toc153911655)

[ Profit and loss calculation: 19](#_Toc153911656)

[ View selected services: 20](#_Toc153911657)

[ View Information of Customer: 20](#_Toc153911658)

[ View how many tickets have been sold: 21](#_Toc153911659)

[ Customer Menu: 21](#_Toc153911660)

[ View the schedule: 22](#_Toc153911661)

[ View Discount Options: 22](#_Toc153911662)

[ Add Personal Information: 23](#_Toc153911663)

[ Add Flight Information: 23](#_Toc153911664)

[ Inflight Services: 24](#_Toc153911665)

[ Class Selection: 24](#_Toc153911666)

[ View Budget Requirements: 25](#_Toc153911667)

[ Enter your Budget: 25](#_Toc153911668)

[ Available Airlines: 26](#_Toc153911669)

[ View the price: 26](#_Toc153911670)

[ Reviews and Feedback: 27](#_Toc153911671)

[ Ticket Receipt: 27](#_Toc153911672)

[ Cancel the ticket: 28](#_Toc153911673)

[ Flowchart 29](#_Toc153911674)

[ Data structures: 30](#_Toc153911675)

[ Function prototypes: 30](#_Toc153911676)

[ Weaknesses in my project: 34](#_Toc153911677)

[ Future directions: 34](#_Toc153911678)

# Table of figures:

[Figure 1 8](#_Toc153903259)

[Figure 2 9](#_Toc153903260)

[Figure 3 9](#_Toc153903261)

[Figure 4 10](#_Toc153903262)

[Figure 5 11](#_Toc153903263)

[Figure 6 11](#_Toc153903264)

[Figure 7 12](#_Toc153903265)

[Figure 8 12](#_Toc153903266)

[Figure 9 13](#_Toc153903267)

[Figure 10 13](#_Toc153903268)

[Figure 11 14](#_Toc153903269)

[Figure 12 14](#_Toc153903270)

[Figure 13 15](#_Toc153903271)

[Figure 14 15](#_Toc153903272)

[Figure 15 16](#_Toc153903273)

[Figure 16 16](#_Toc153903274)

[Figure 17 17](#_Toc153903275)

[Figure 18 17](#_Toc153903276)

[Figure 19 18](#_Toc153903277)

[Figure 20 18](#_Toc153903278)

[Figure 21 19](#_Toc153903279)

[Figure 22 19](#_Toc153903280)

[Figure 23 20](#_Toc153903281)

[Figure 24 20](#_Toc153903282)

[Figure 25 21](#_Toc153903283)

[Figure 26 21](#_Toc153903284)

[Figure 27 22](#_Toc153903285)

[Figure 28 22](#_Toc153903286)

[Figure 29 23](#_Toc153903287)

[Figure 30 23](#_Toc153903288)

[Figure 31 24](#_Toc153903289)

[Figure 32 24](#_Toc153903290)

[Figure 33 25](#_Toc153903291)

[Figure 34 25](#_Toc153903292)

[Figure 35 26](#_Toc153903293)

[Figure 36 26](#_Toc153903294)

[Figure 37 27](#_Toc153903295)

[Figure 38 27](#_Toc153903296)

[Figure 39 28](#_Toc153903297)

# Short Description of my Project:

This application is about an airline management system which includes the booking of tickets, the cancellation of tickets, and the allotment of different services to the customers. This application will help in allocating the discount to the customers and will also facilitate the owner and paying the salary to the employees with the total revenue generated.

# Contribution towards CS:

This application uses the field of computer science to enhance the travel experience of customers. And it also uses the concepts of computer science to provide to the user a console based experience. This helps the users in booking of tickets.

# Users of Application:

There are three users’ types of Sky wings Airlines:

## **Admin:**

The owner of sky wings airlines will have access to all the revenue generated and will have the record of all the employees registered and will be able to made announcements to the employees and set the price of ticket and will be able to add new airlines and if he wants to remove an employee.

## Customer:

The customer of sky wings airlines will be able to add his information and will be able to add the information regarding to the booking of ticket like his CNIC and other things. And the customer will be able to avail the discount given by owner on certain days.

## Manager:

The manager of sky wings airlines will be able to view the airlines on which the tickets have been booked. And will also be able to view how many tickets have been sold. As a manager he will surely need to view the services which the customers have selected so that he can facilitate the customers.

# Functional Requirements:

*User Story ID:*

*As a* *I want to perform So that I can*

|  |  |  |
| --- | --- | --- |
| Admin | Add employee | Add the desired employees so that the admin have the record of all employees. |
| Remove employee | Remove the employee whom the admin wants to remove and the employee will be removed. |
| Revenue interface | The admin will have access to the total income so that he have the idea of the income. |
| Add Airlines | The admin will be able to add the desired airlines to the management system. |
| Price allocation | Through this functionality the admin can allocate or update the price of the tickets. |
| Discount allocation | As a result of this functionality the admin will facilitate the customers with the discount option. |
| Announcements to the  employee | The admin will announce something or the thing he wishes to announce to employees. |
| View the customer  reviews | The admin can view the reviews made by customers. |
| Price allocation for  services | The admin can allocates the prices of services and extra charges. |
| View the employees  List | The admin can see the list of the employees who are added up by the admin. |
| Salary allocation | The admin will pay the salary of his employees and can select whose salary he wants to pay. |
| Customer | Add personal  Information | The customer will add the personal information here so that it can help him in booking the ticket. |
| Add flight information | The customer will add the flight information regarding  Where he wants to travel. |
| View Discount options | The customer will view on which certain days the discount is available and can avail the discount. |
| Inflight Services | The customer can avail the different services provided in the flight. |
| Feedback | The customer can give the feedback related to their experience with this application. |
| Cancel the ticket | At the end the customer will have a choice if he wants to cancel the flight or not. |
| Manager | View the counts of  Tickets sold. | The manager can view how many tickets have been sold. |
| View the selected  airlines | The manager can review the selected airlines by customers. |
| View the selected  services | The manager can view the selected services by customers so that he can facilitate them. |
| View the  announcements | The manager can view the announcements made from the owner. |
| Set the schedule | The manager can set the schedule of tickets. |
| View the information  of customer | The manager can view the information of the customer who bought the ticket. |
| Profit and loss  calculation | The manager can calculate the profit and loss. |

# Wireframes:

## Header:

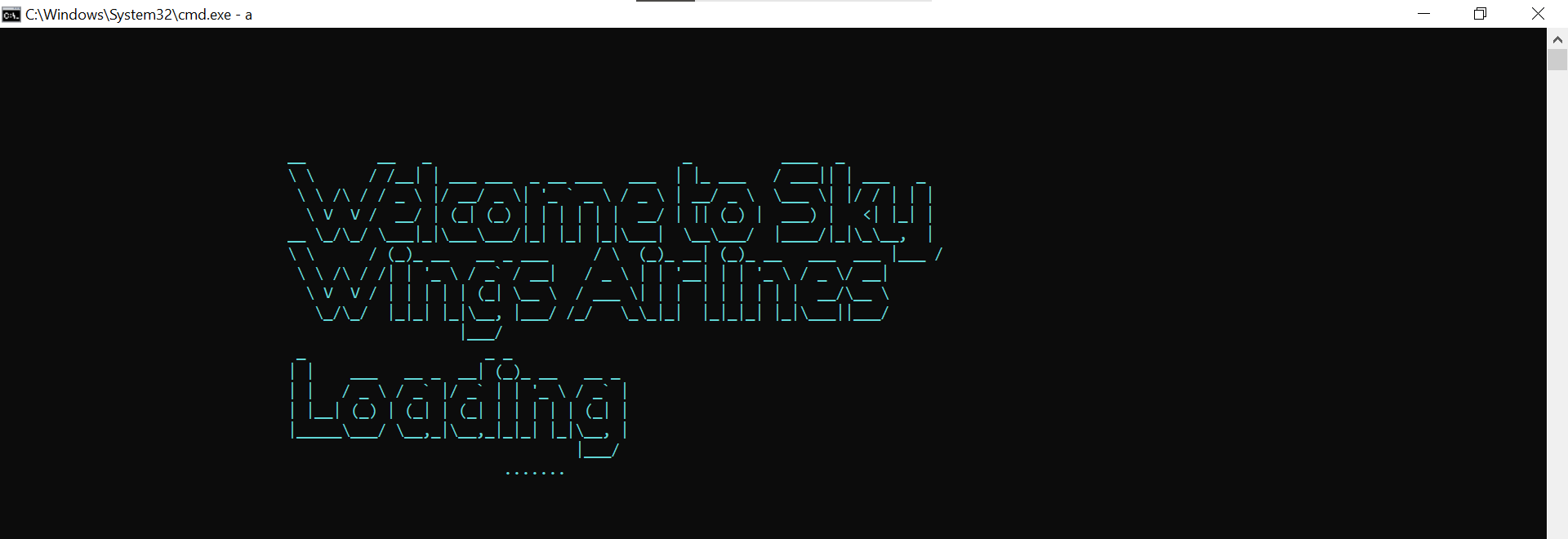


Figure 1

## Login Screen:

# 

Figure 2

## 

## **Sign in Screen**:

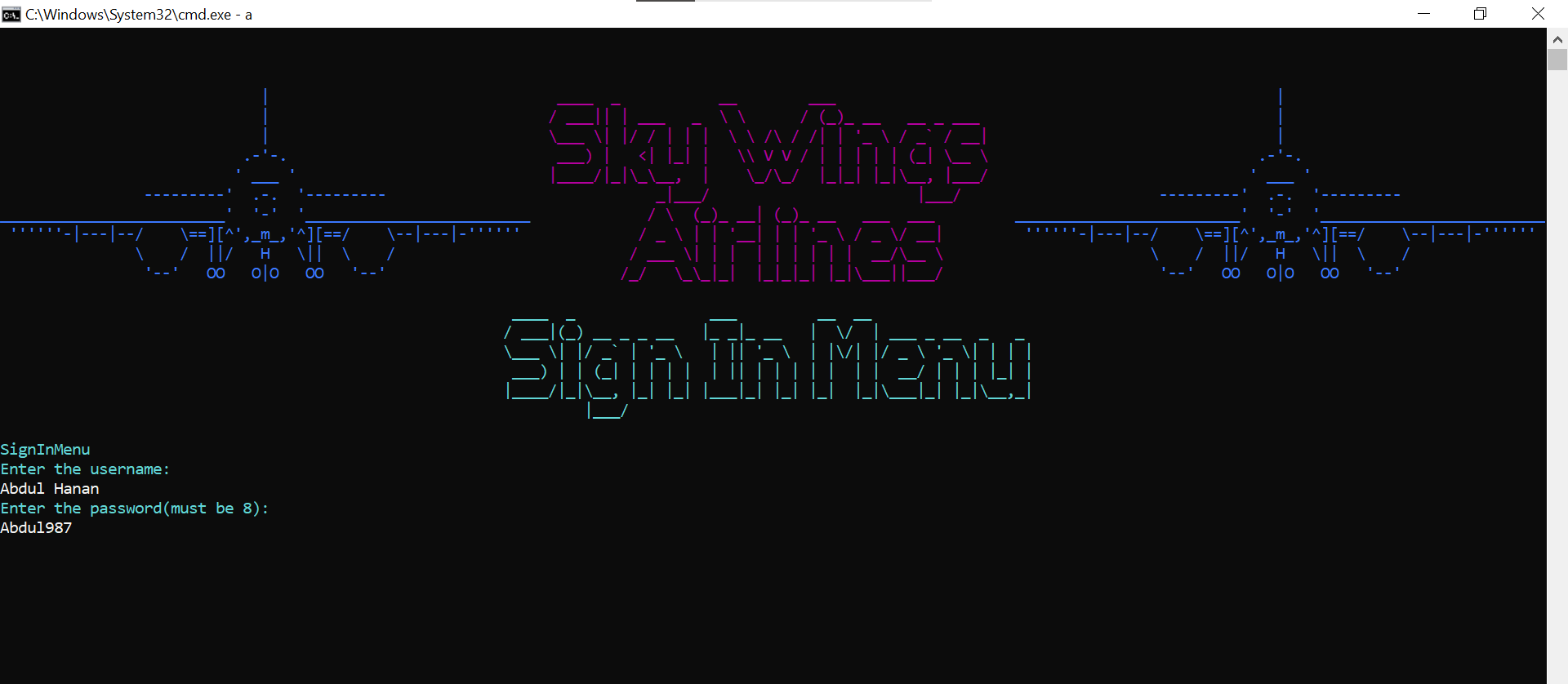


Figure 3

# 

## Sign Up screen:

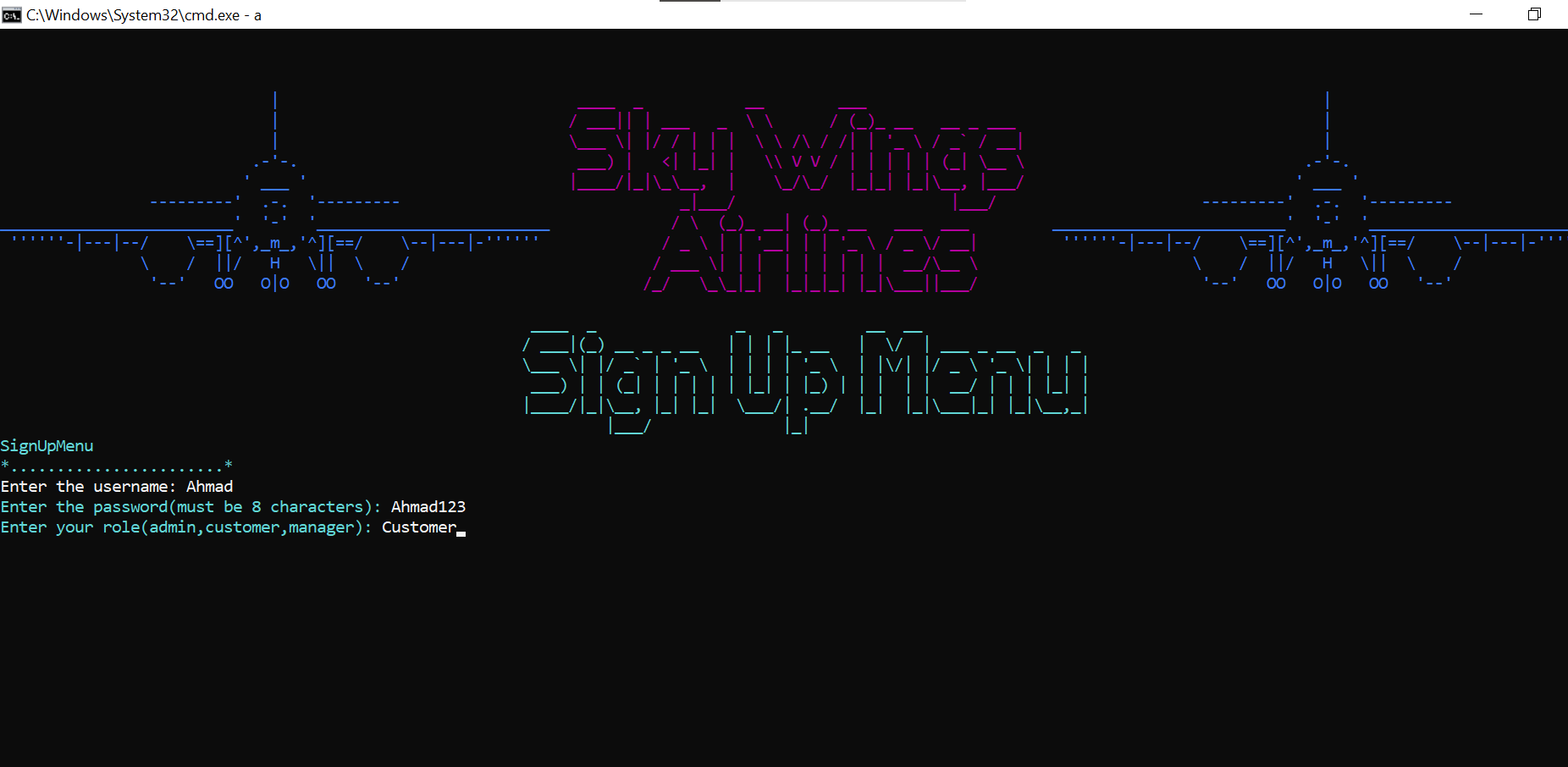


Figure 4

# Admin Interface:

# 

Figure 5

## Add employee:

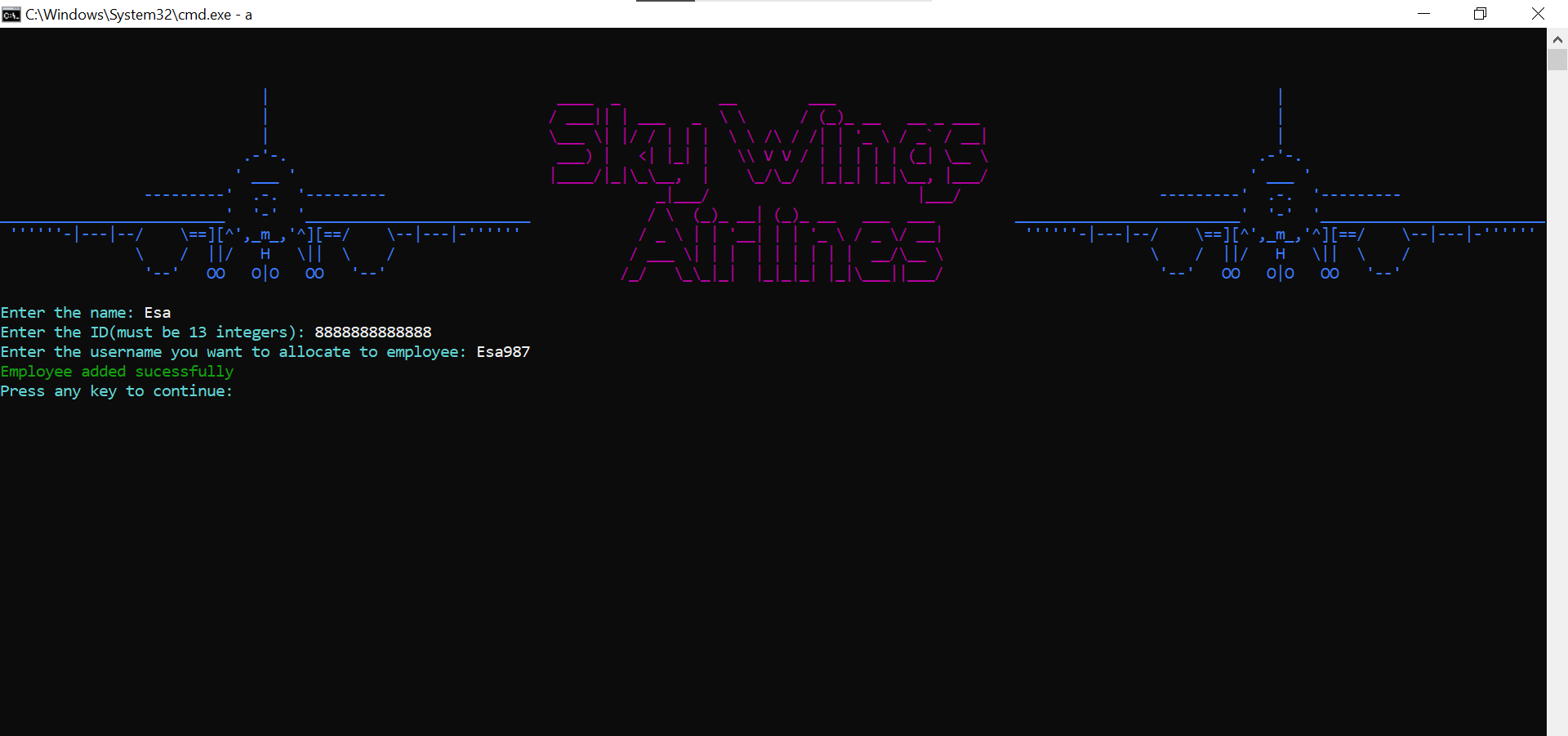


Figure 6

## Remove employee:

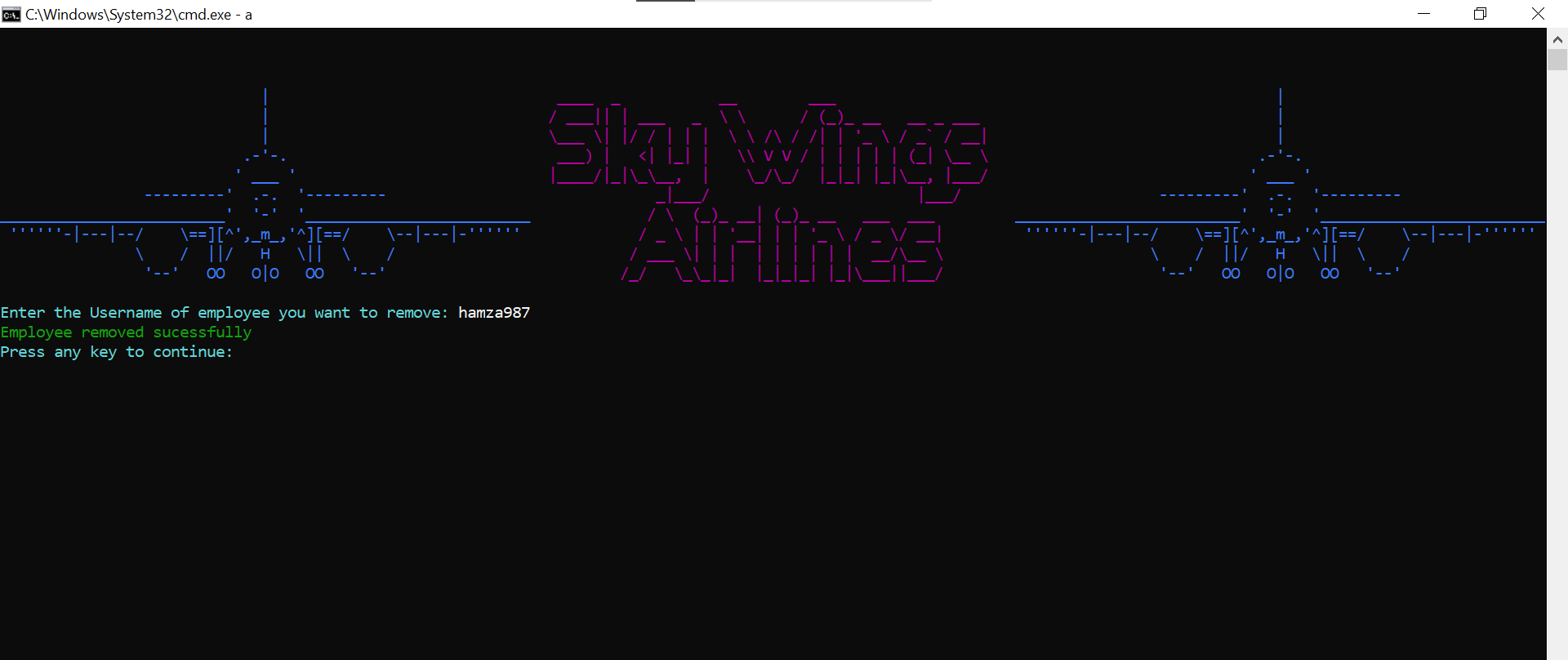


Figure 7

## Add Airlines:

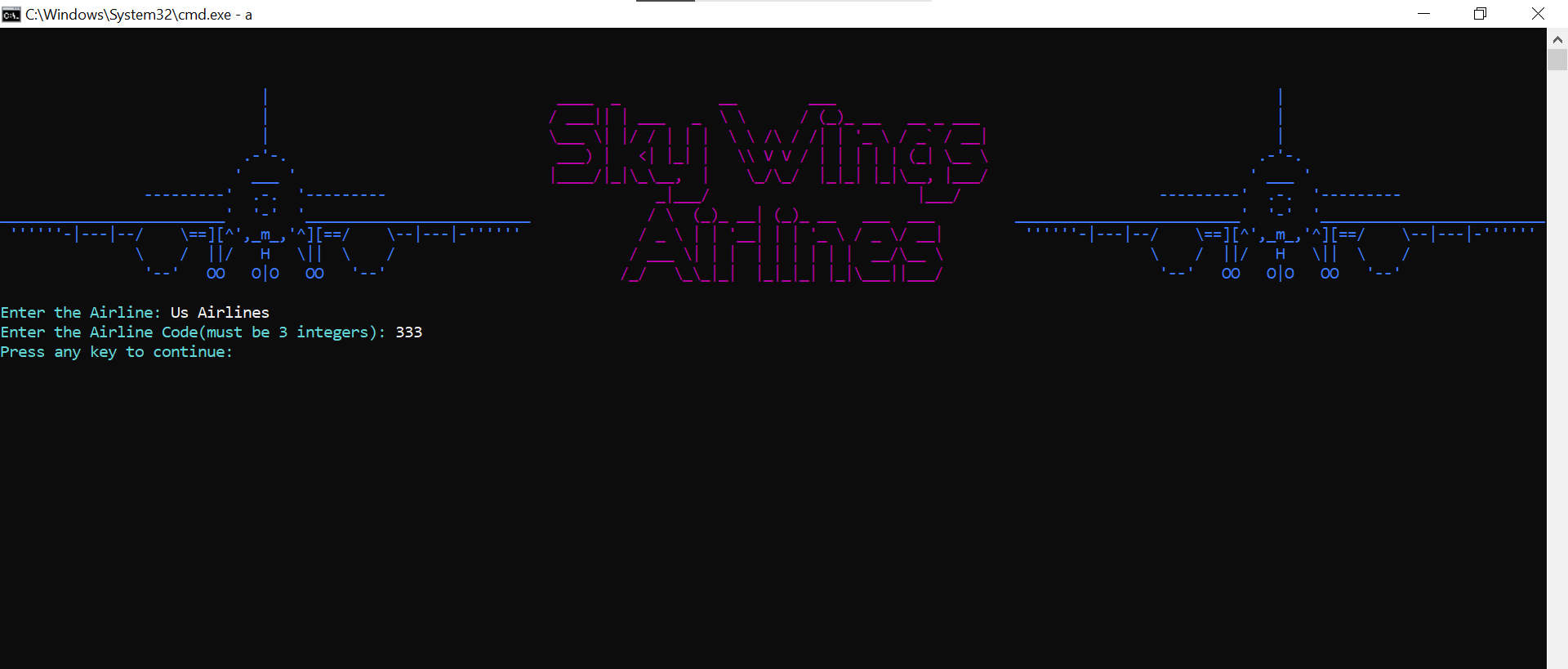


Figure 8

## Employee list:

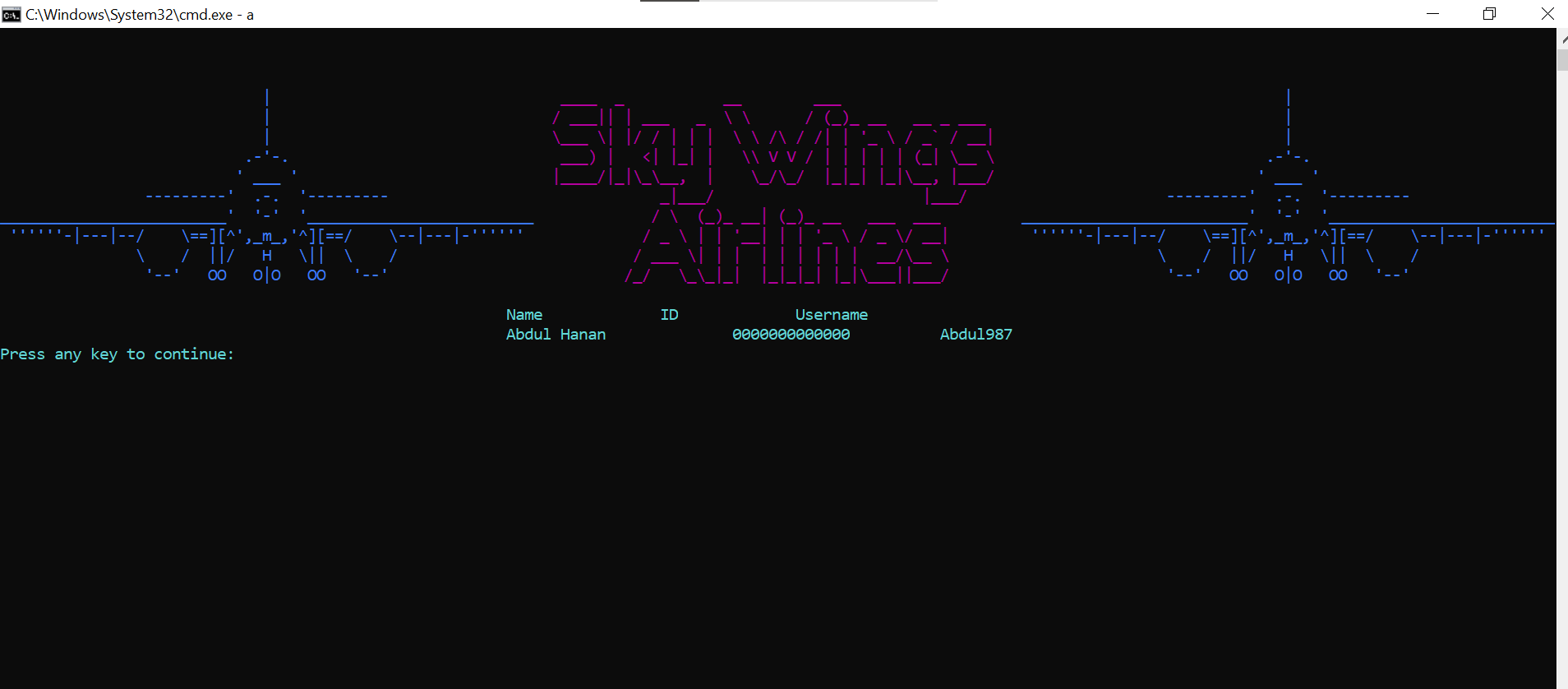


Figure 9

## Discount allocation:

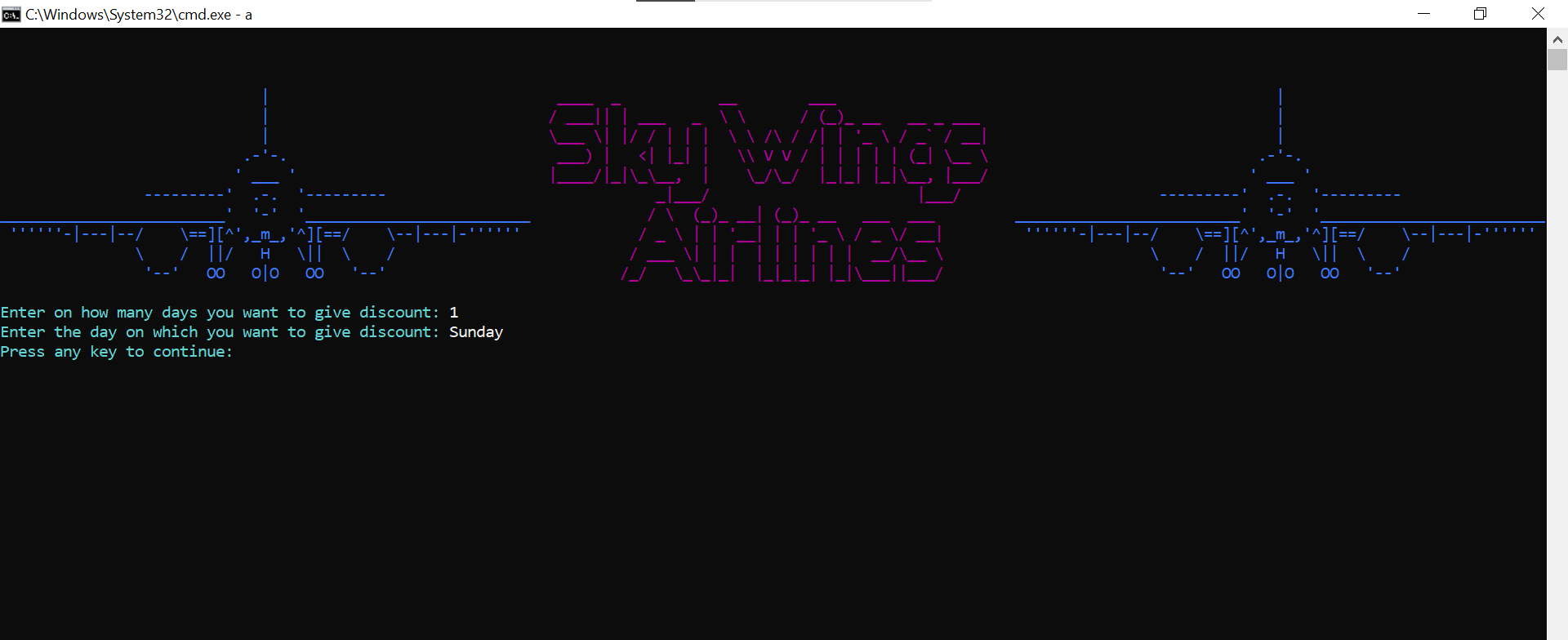


Figure 10

## Announcement to employees:

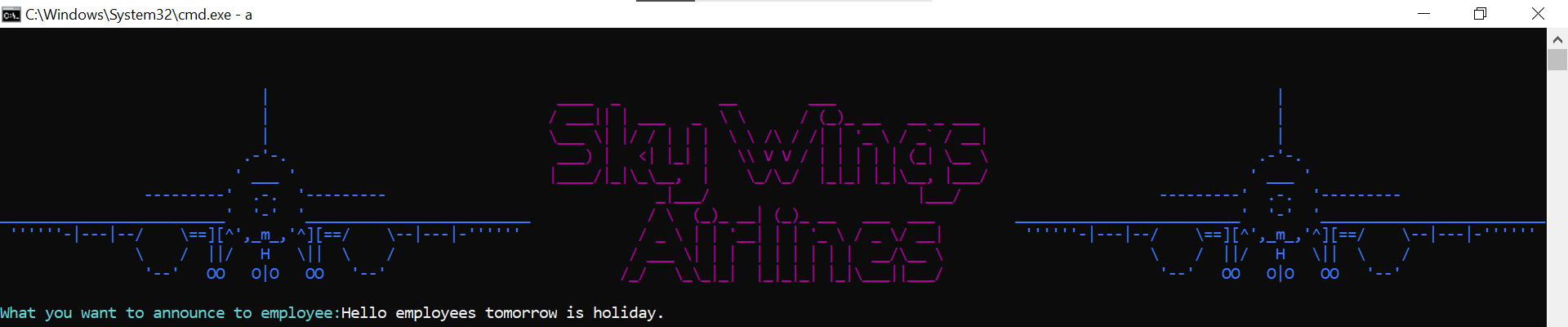


Figure 11

## Salary management:

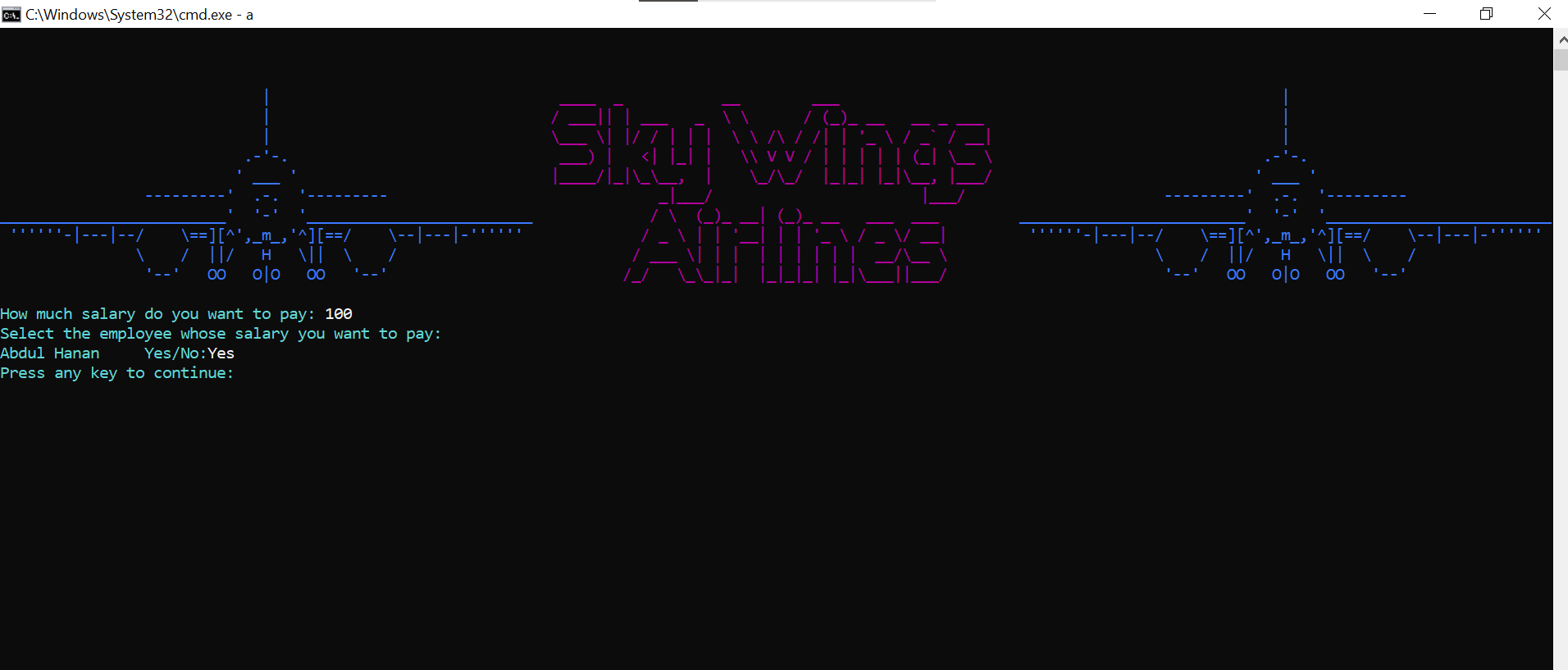


Figure 12

## Percentage of Discount:

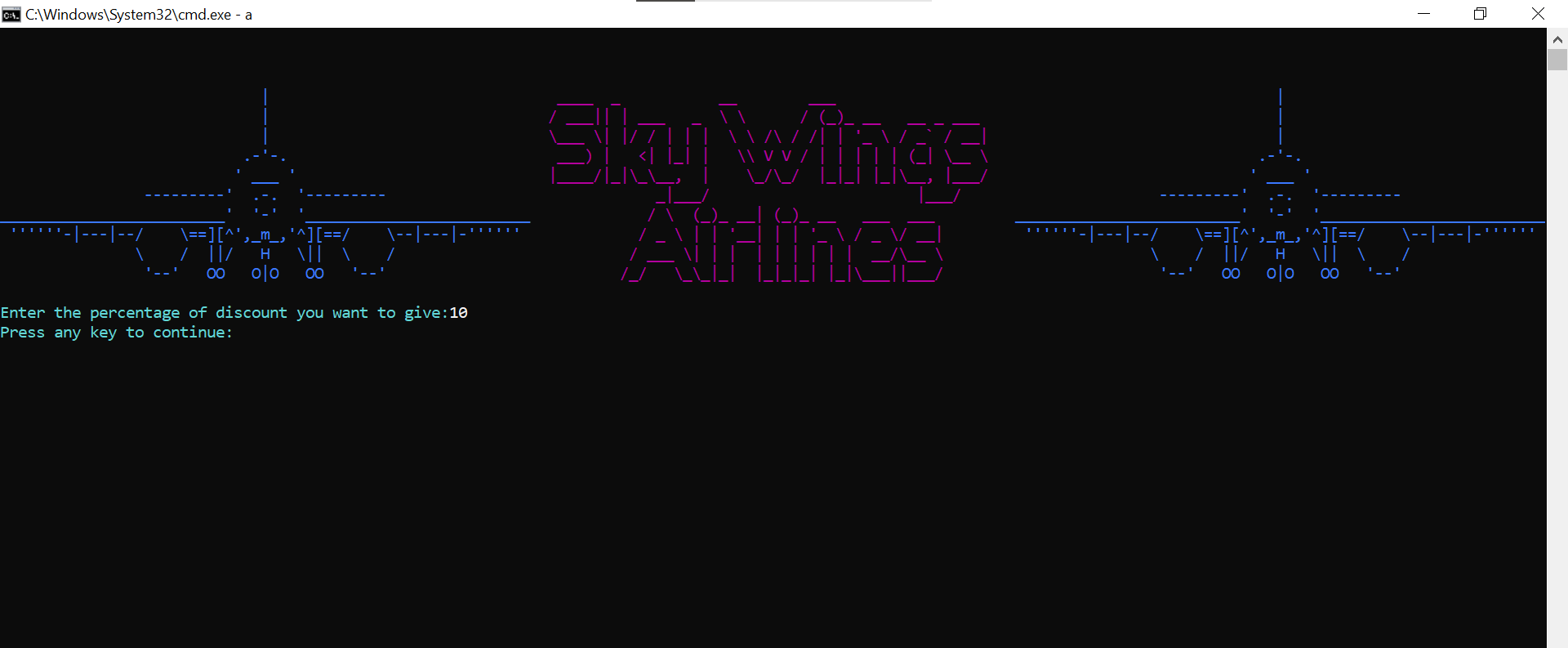


Figure 13

## Price allocation:

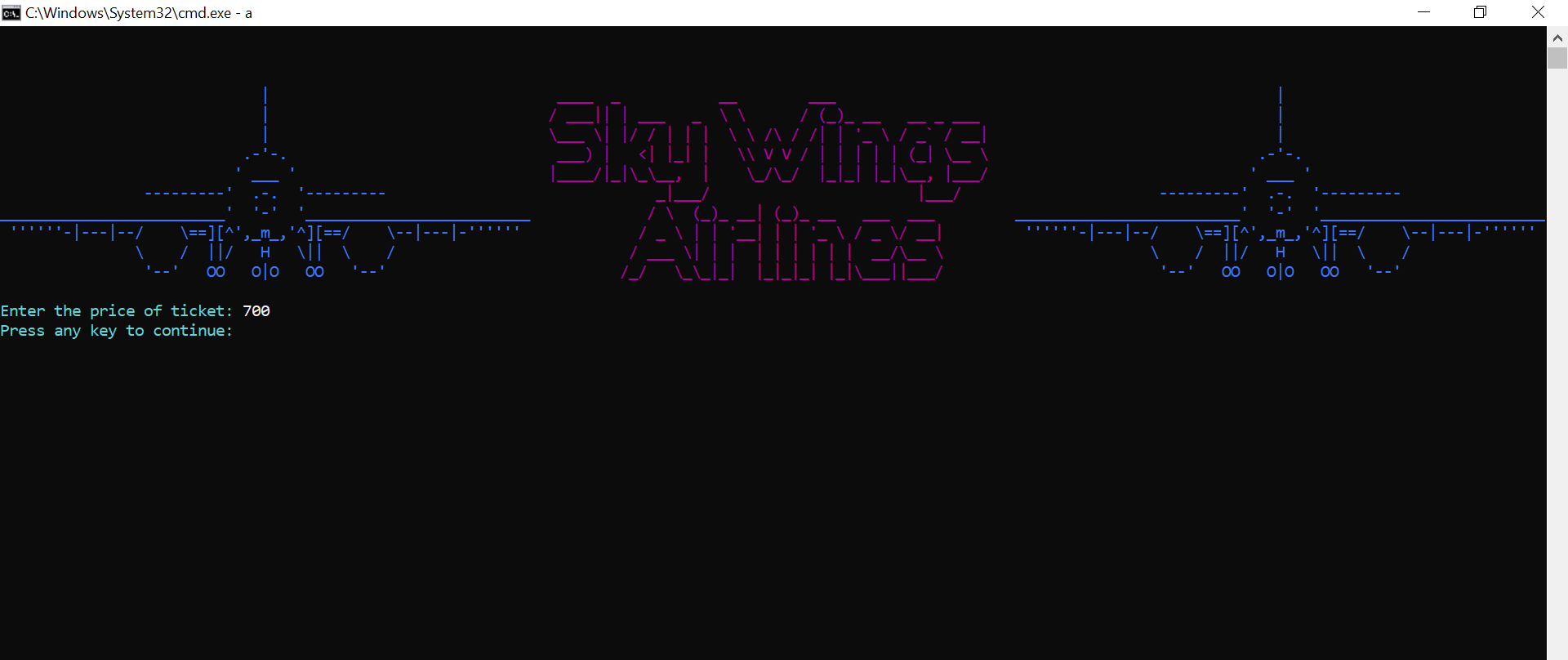


Figure 14

## Price allocation for services:

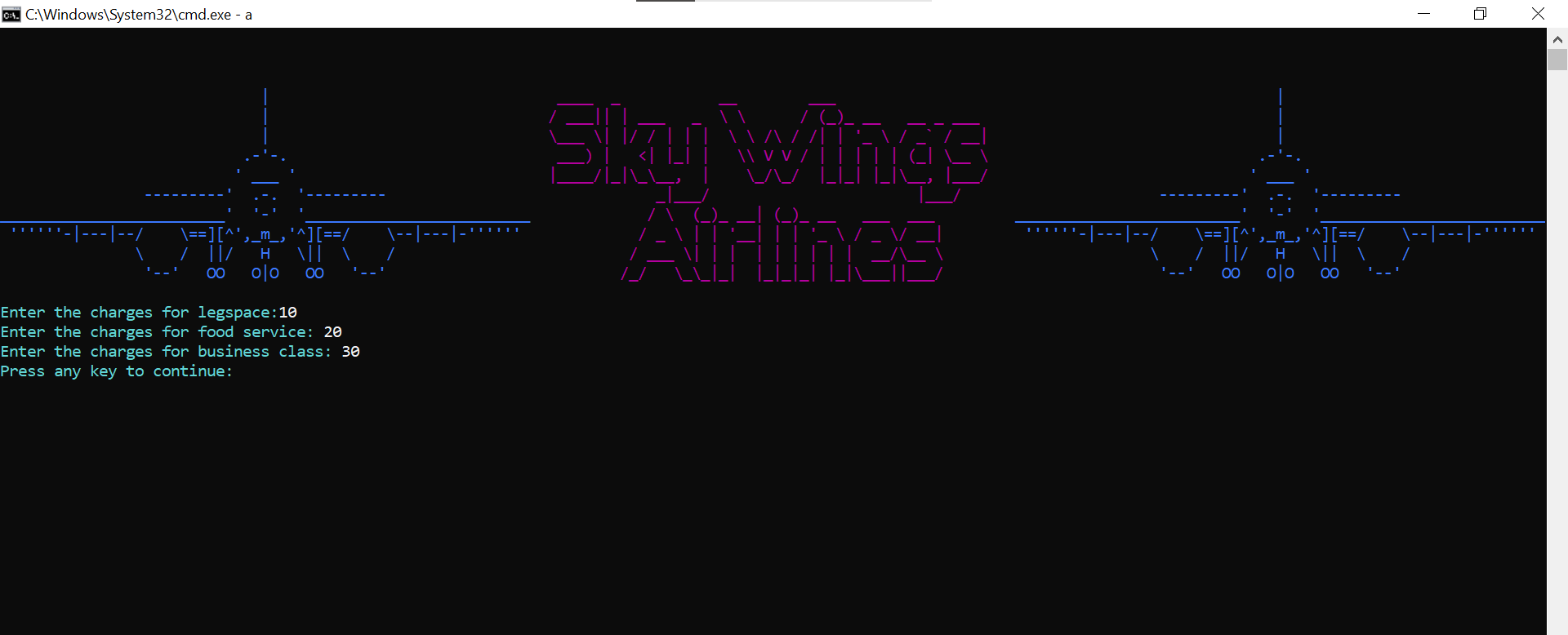


Figure 15

## Customer Reviews:

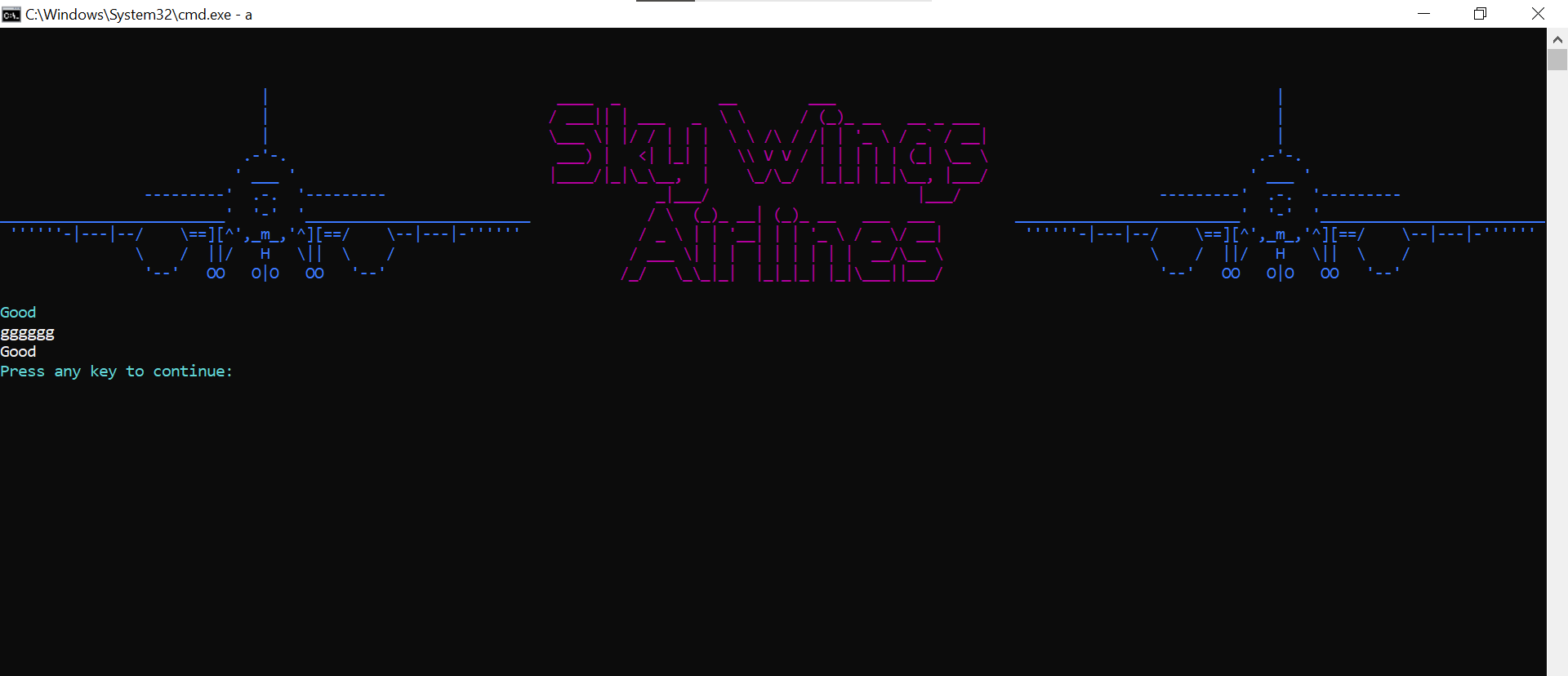


Figure 16

## Revenue Interface:

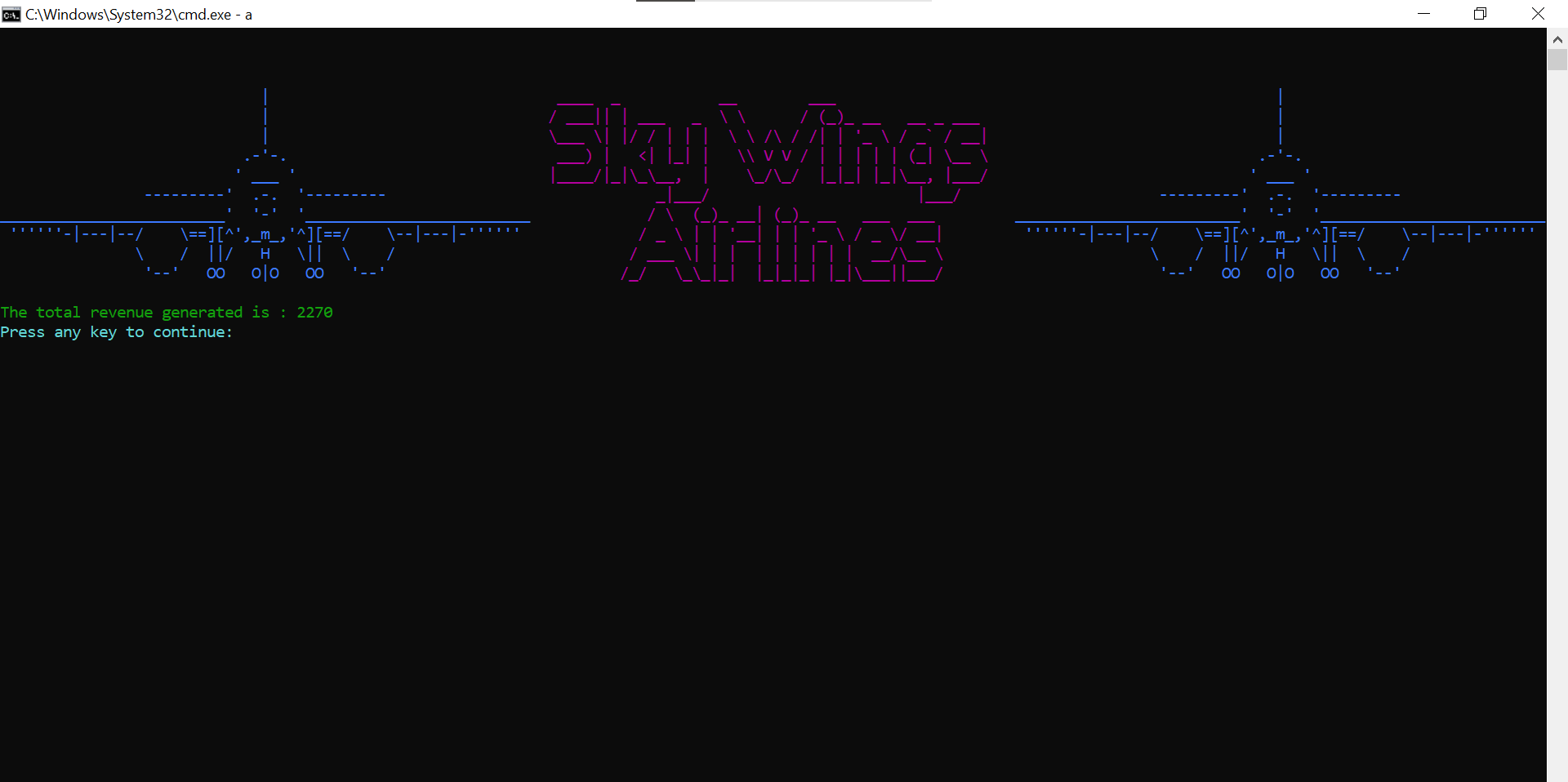


Figure 17

# Manager Menu:



Figure 18

## Schedule of Tickets:

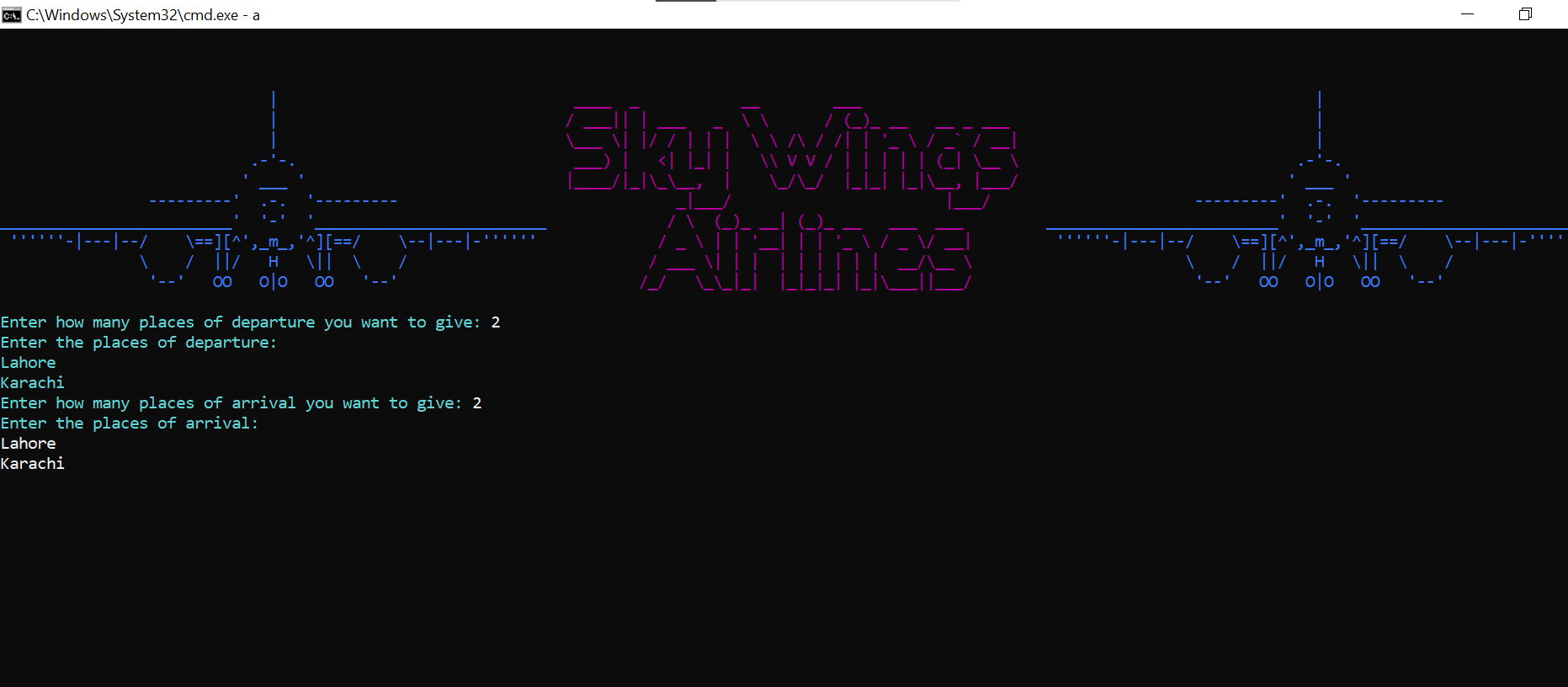


Figure 19

## View selected airlines:

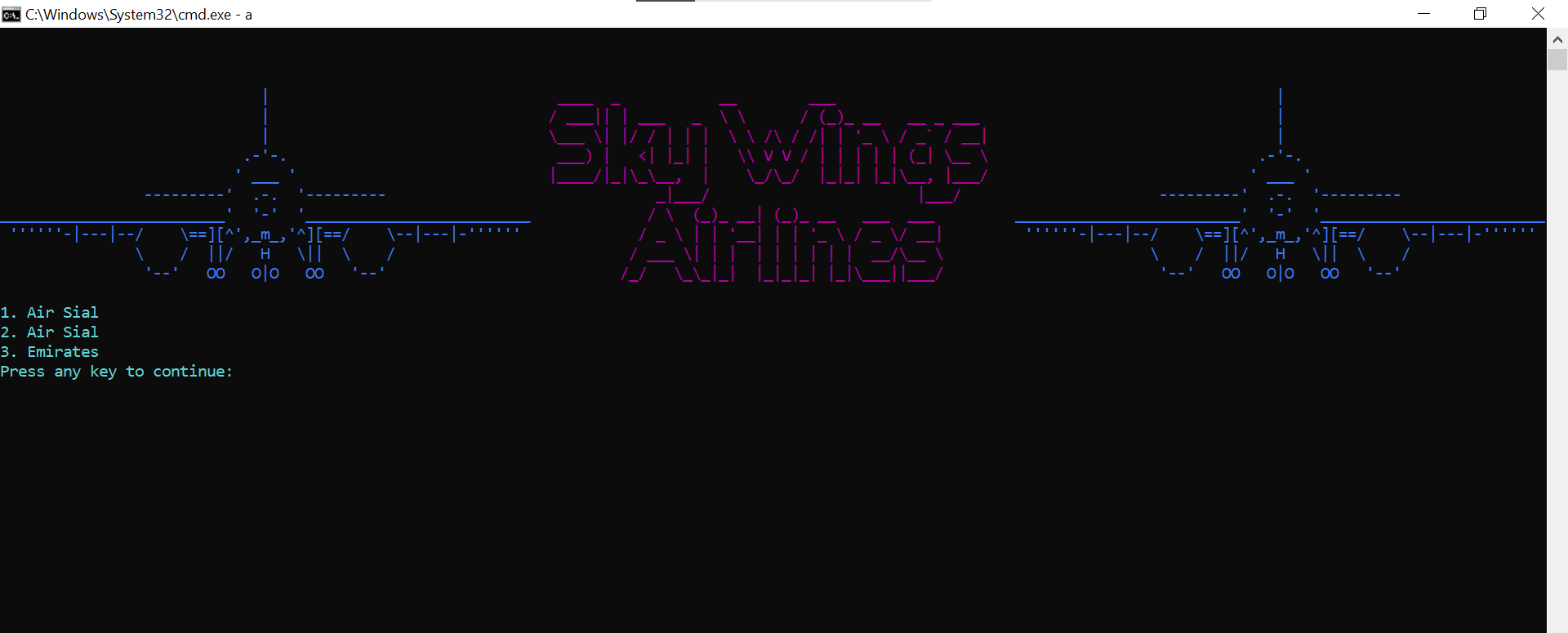


Figure 20

## View Announcemet from admin:

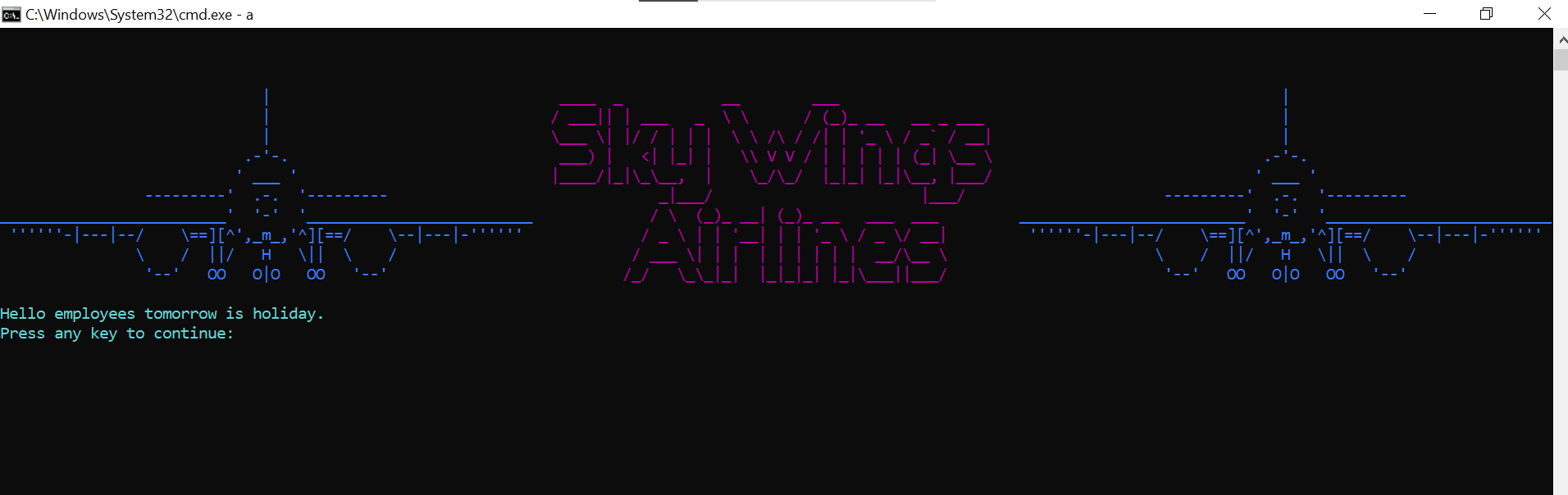


Figure 21

## Profit and loss calculation:

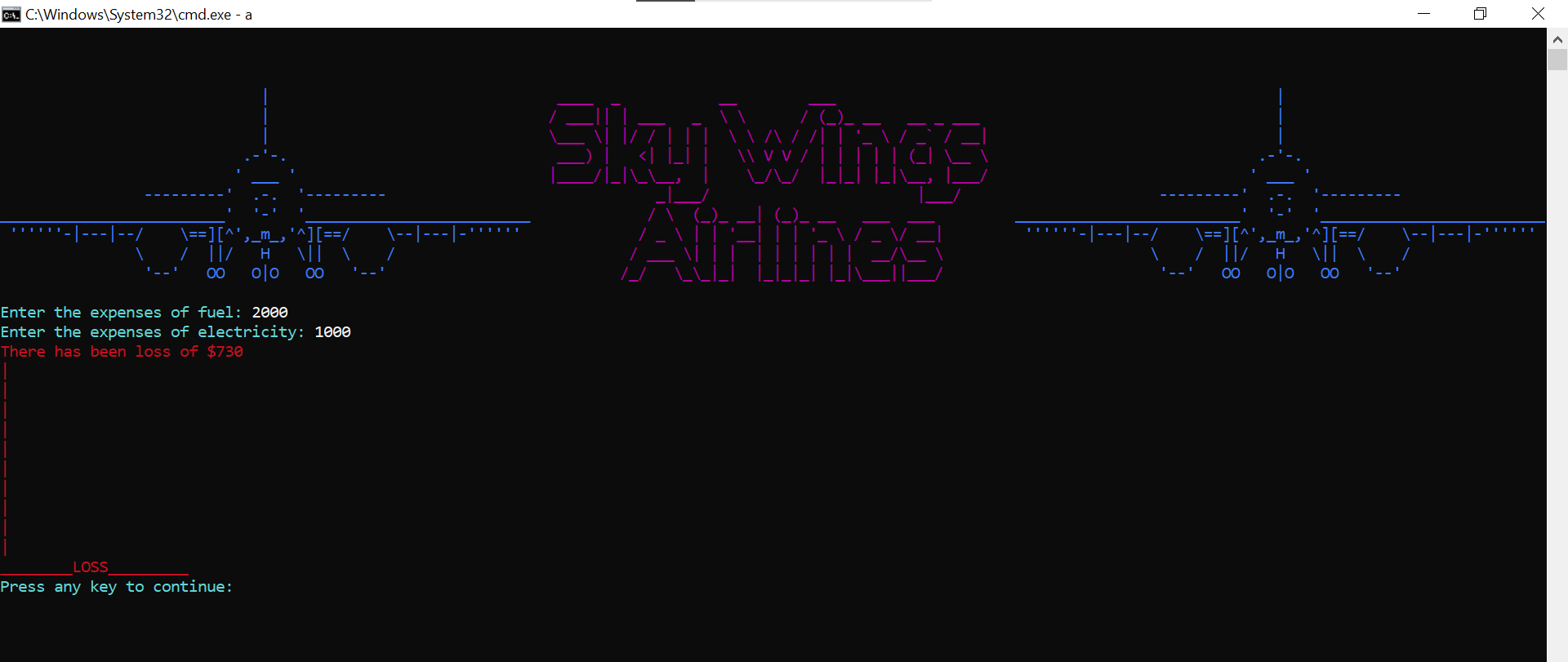


Figure 22

## View selected services:

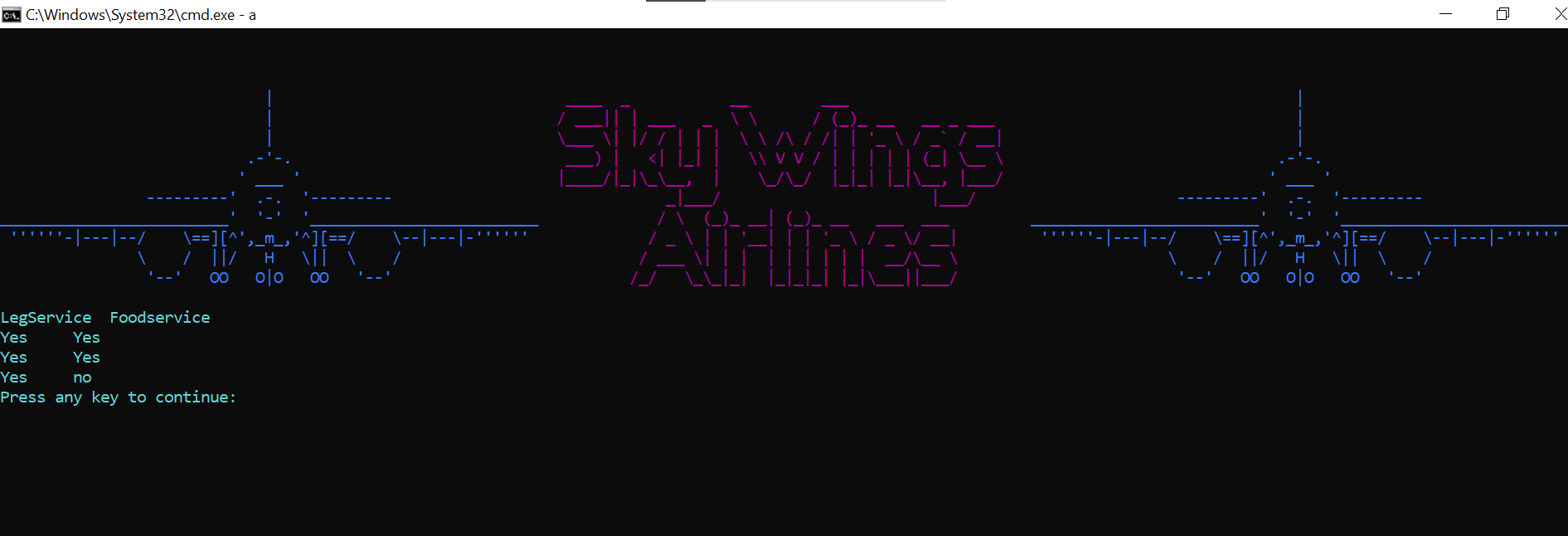


Figure 23

## View Information of Customer:

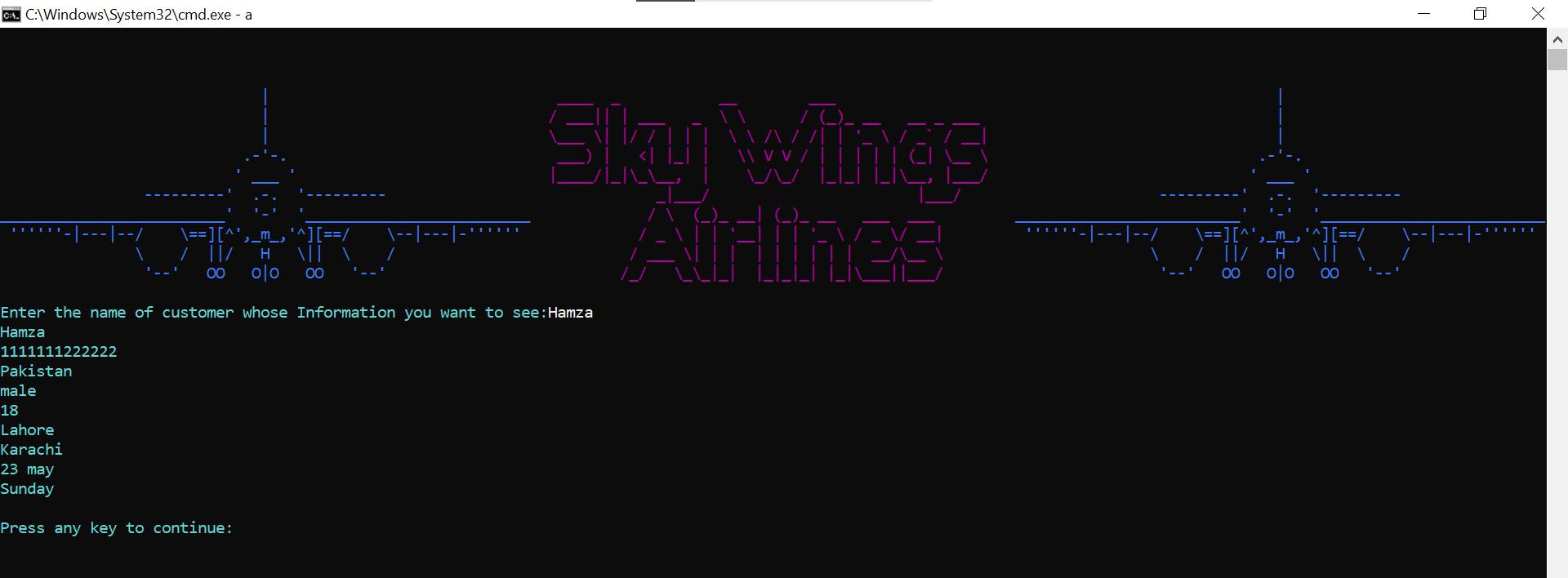


Figure 24

## View how many tickets have been sold:

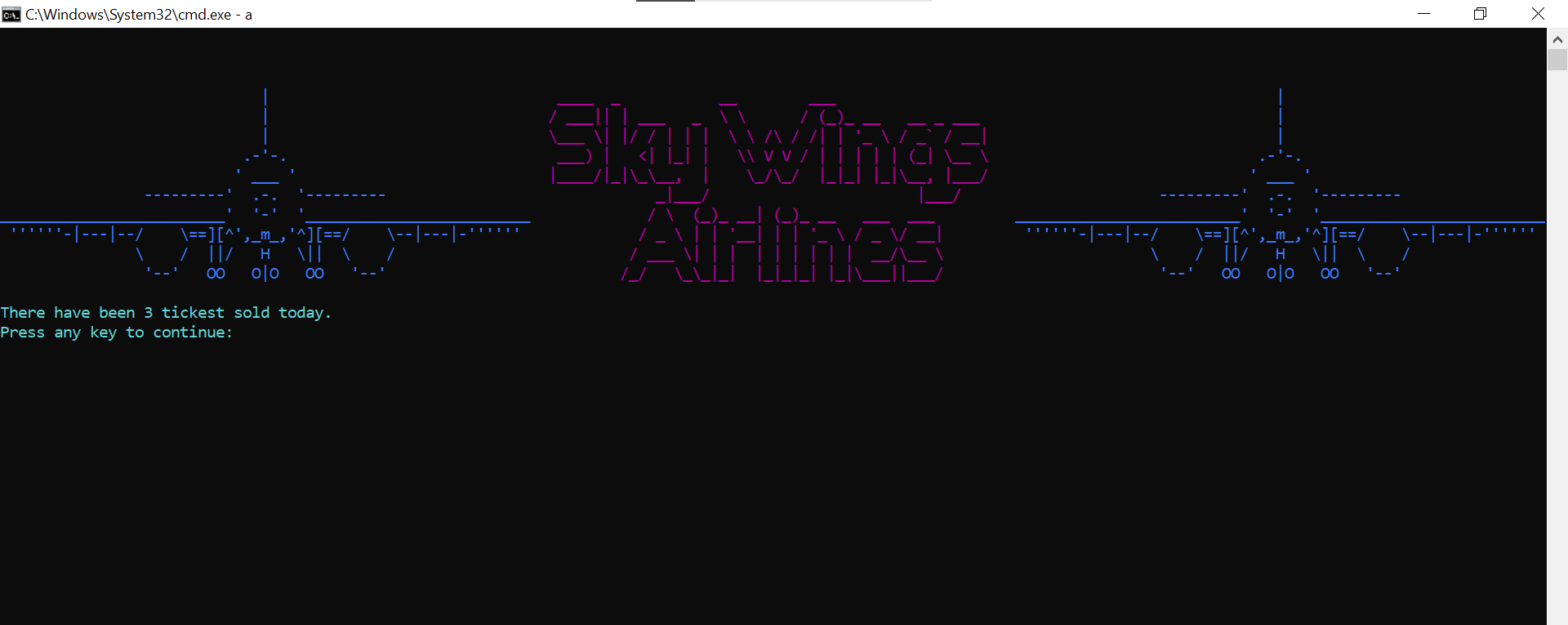


Figure 25

# Customer Menu:

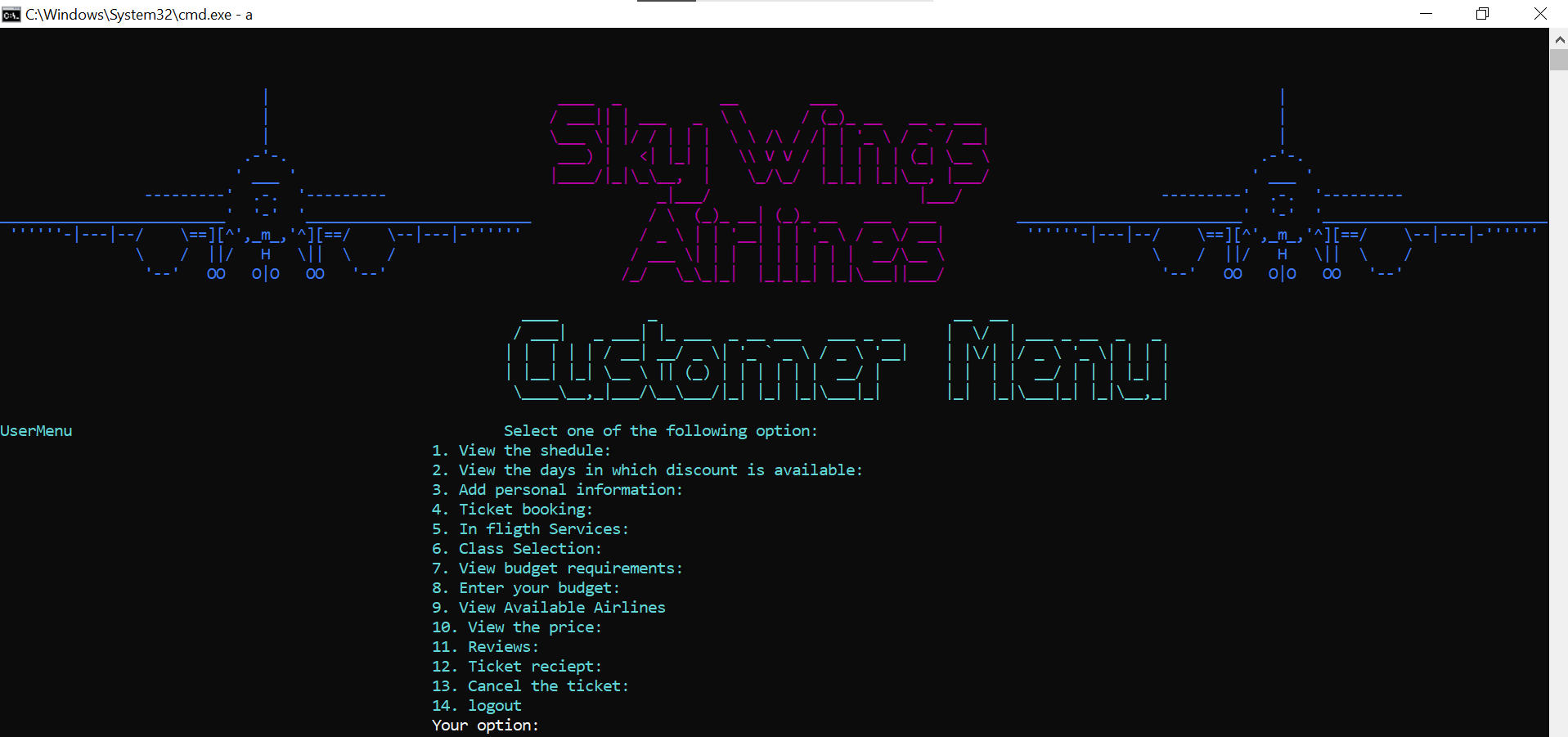


Figure 26

## View the schedule:

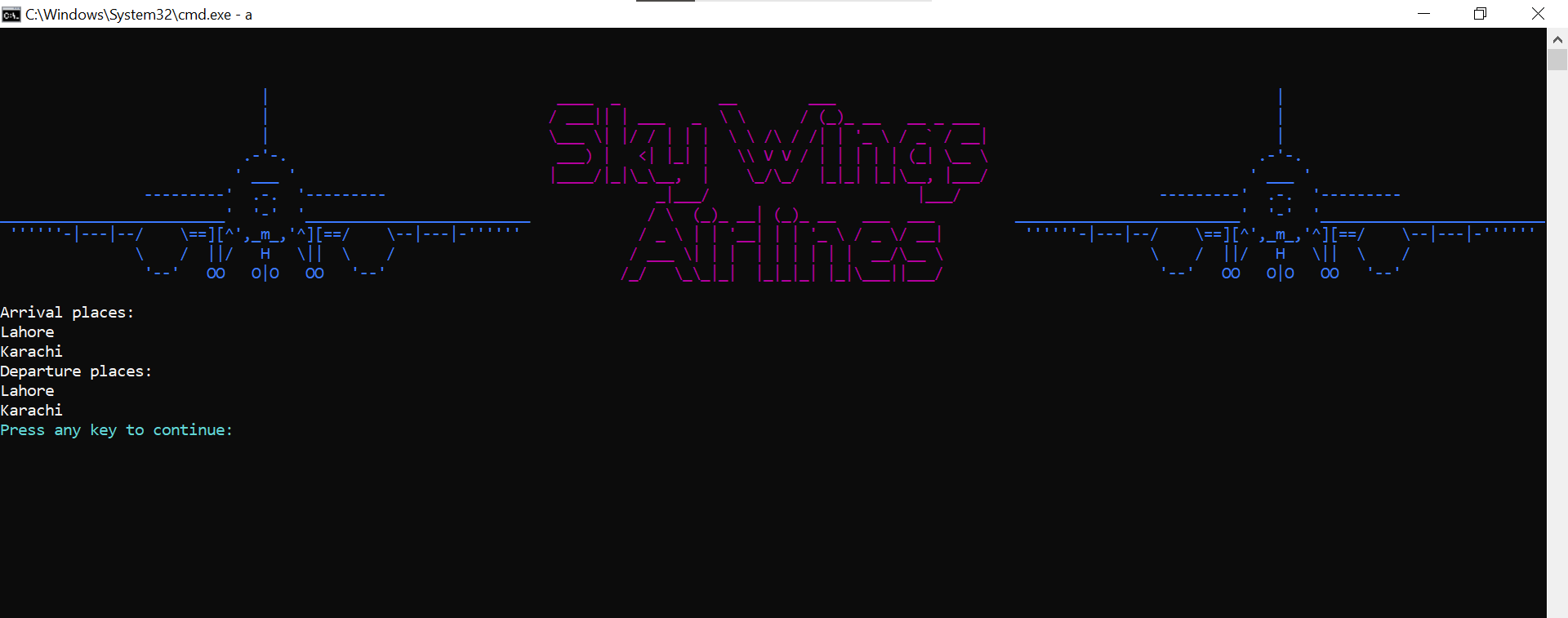


Figure 27

## View Discount Options:

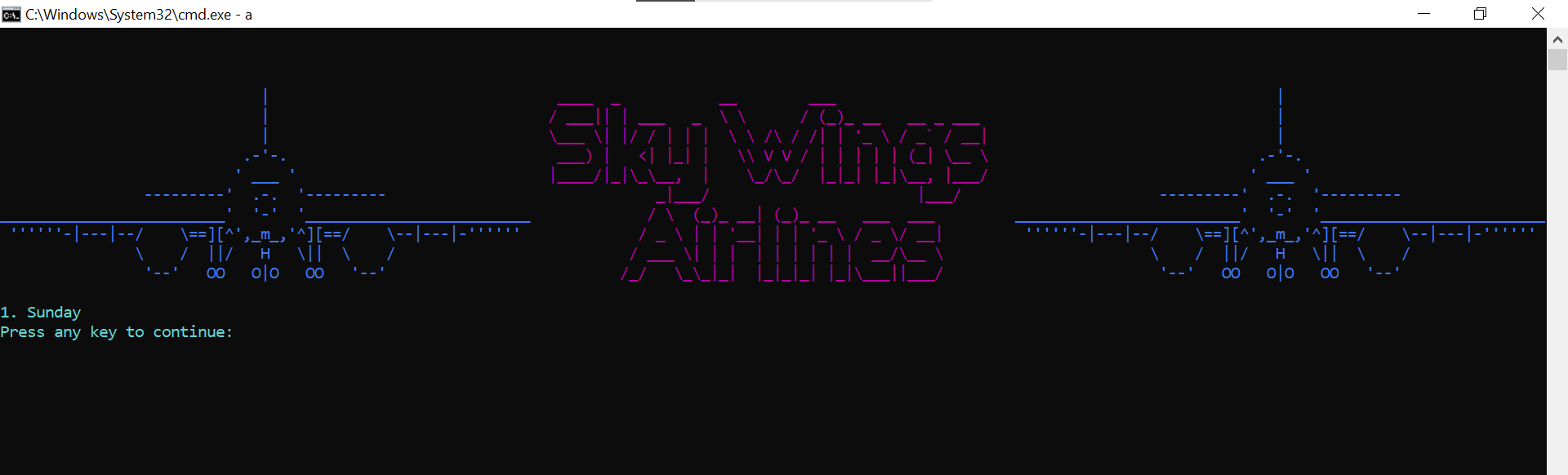


Figure 28

## Add Personal Information:

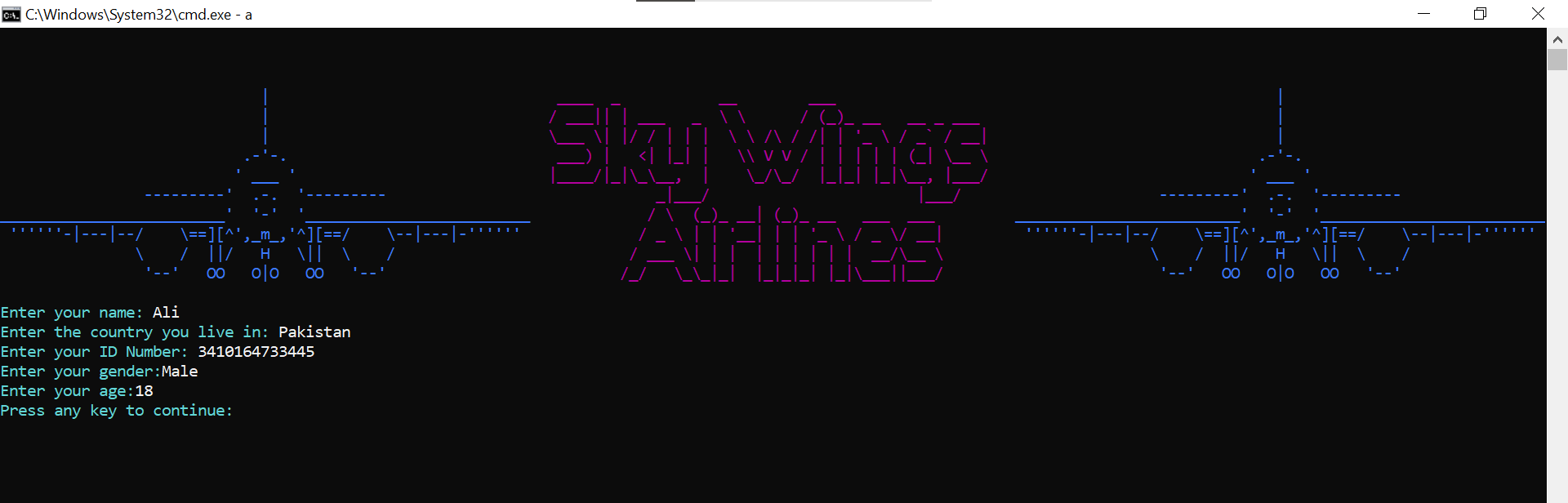


Figure 29

## Add Flight Information:

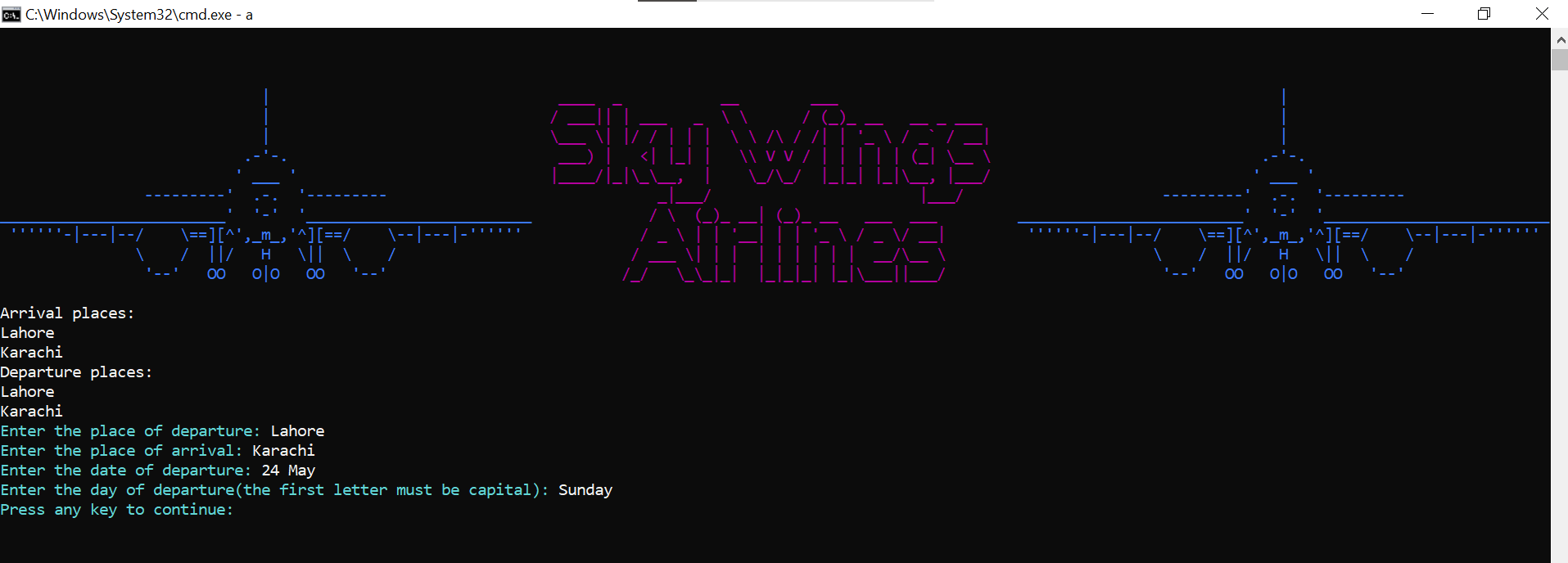


Figure 30

## Inflight Services:

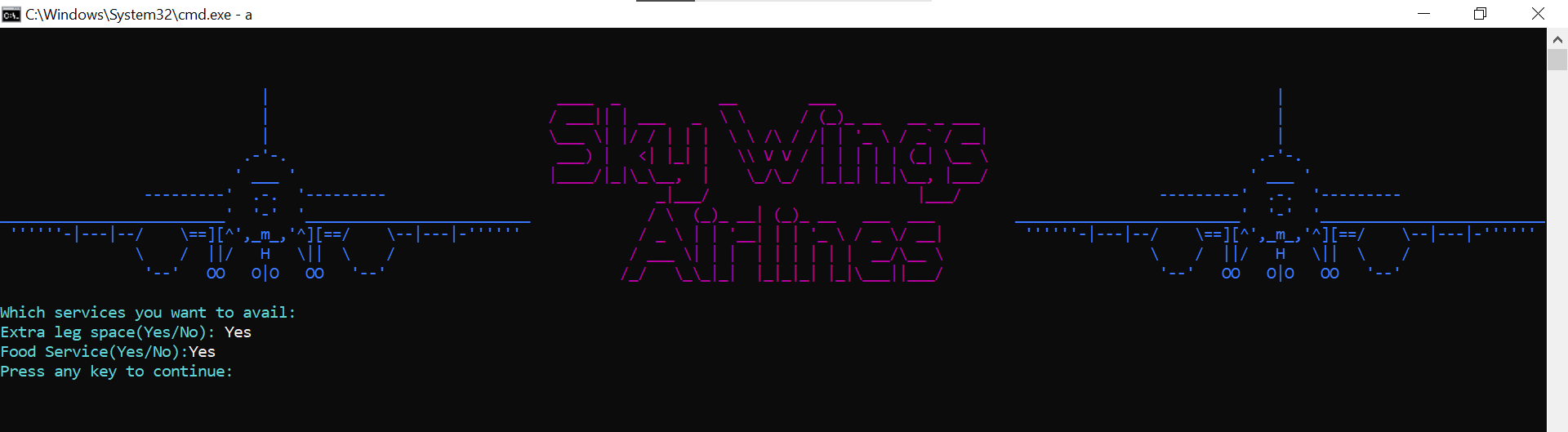


Figure 31

## Class Selection:

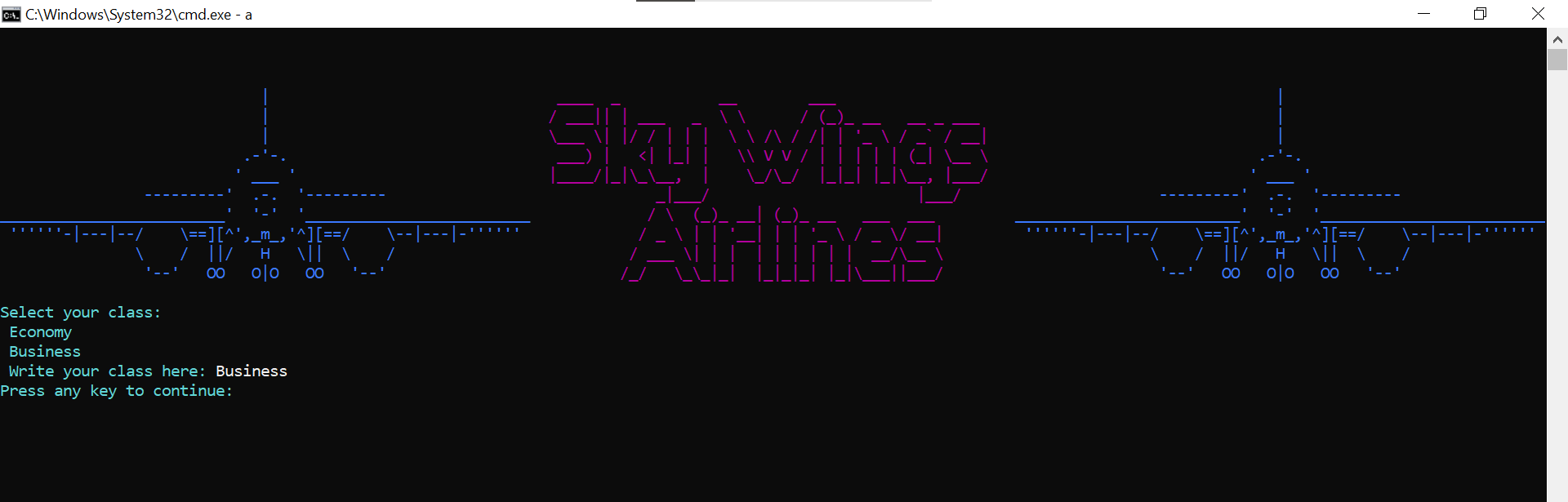


Figure 32

## View Budget Requirements:



Figure 33

## Enter your Budget:

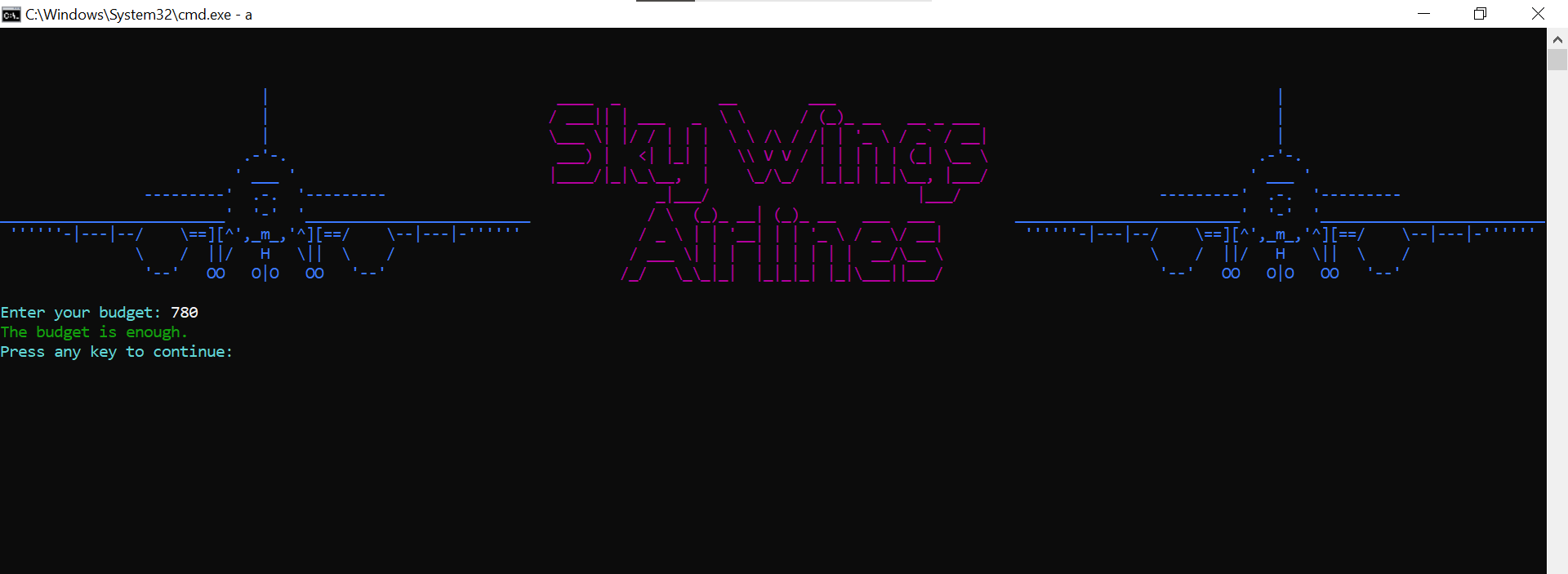


Figure 34

## Available Airlines:

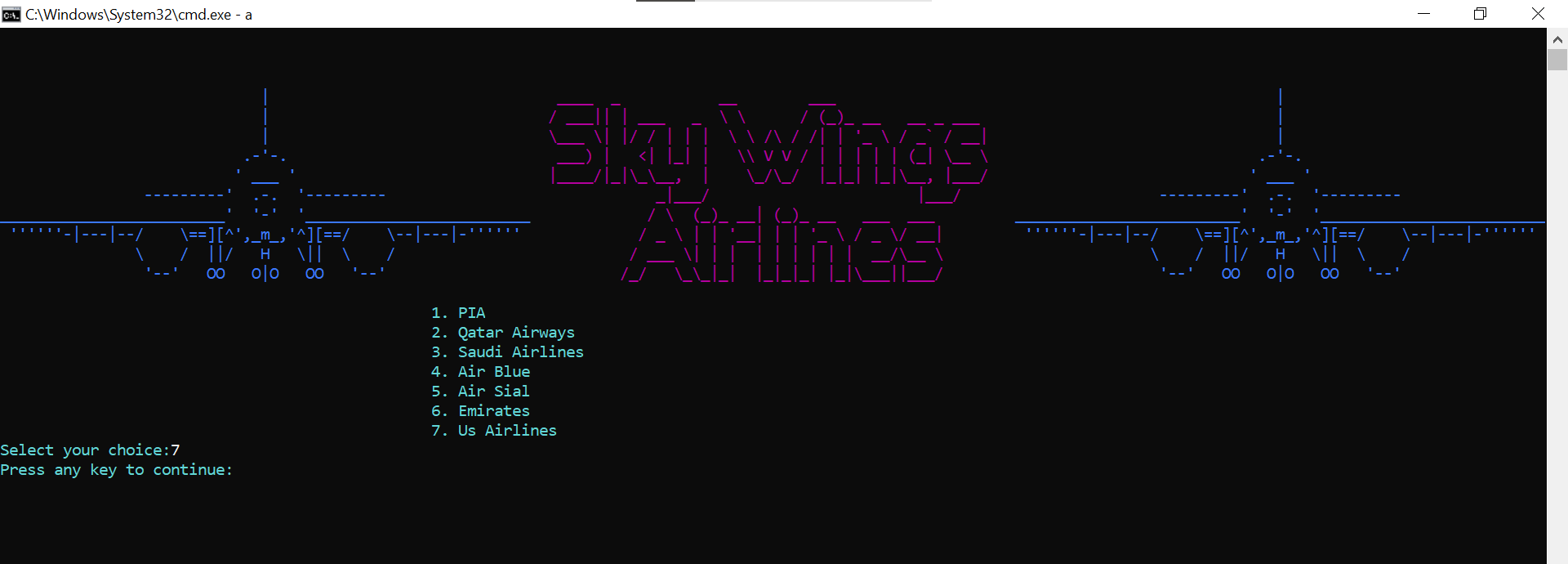


Figure 35

## View the price:

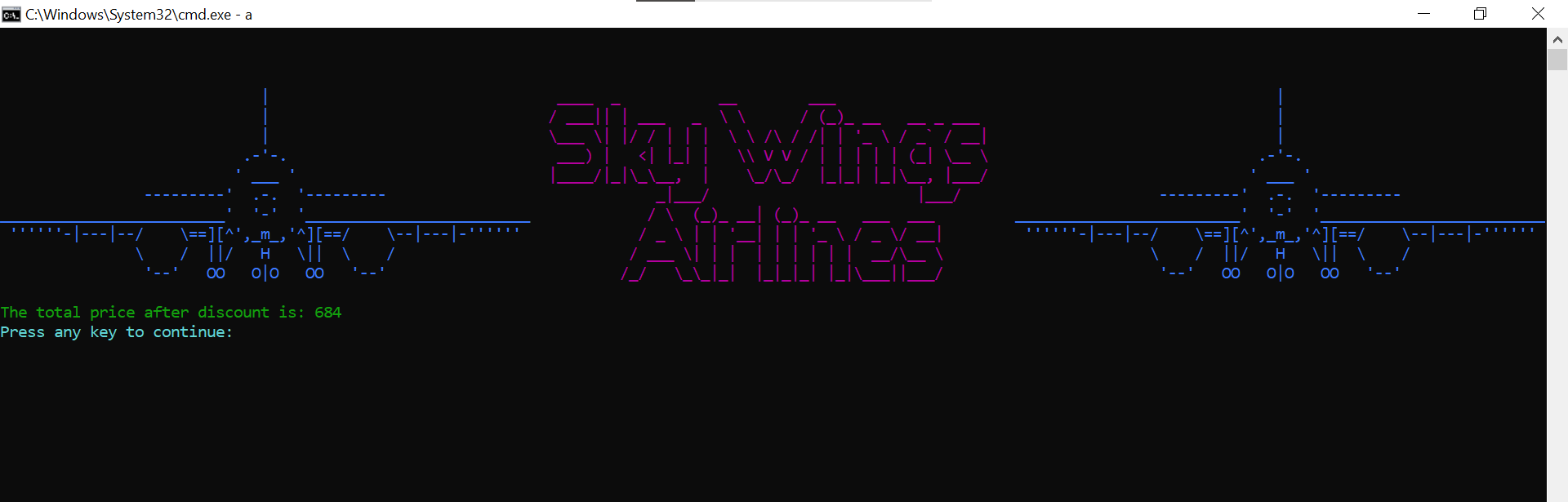


Figure 36

## Reviews and Feedback:

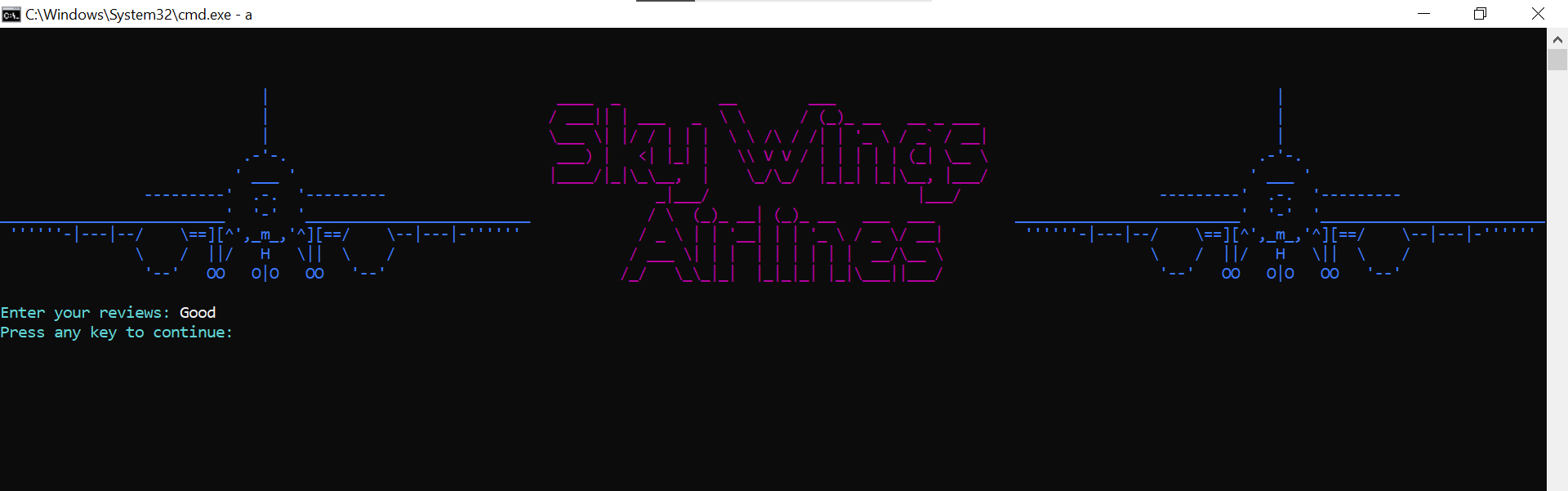


Figure 37

## Ticket Receipt:

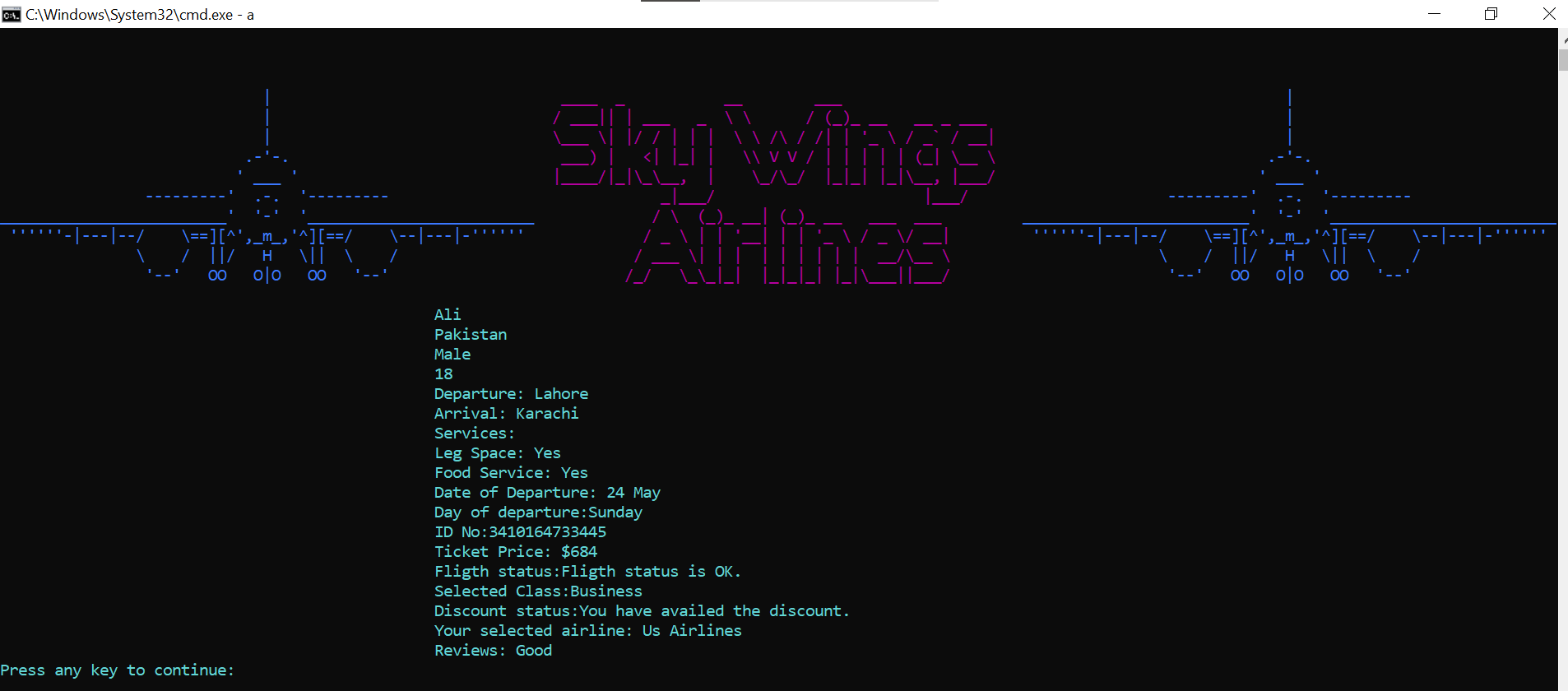


Figure 38

## Cancel the ticket:

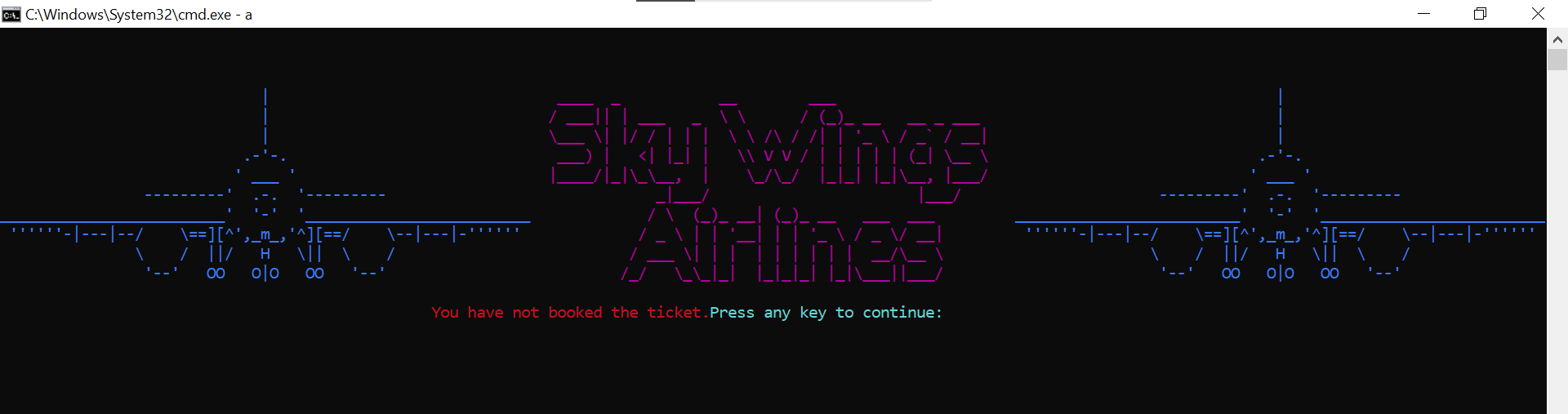
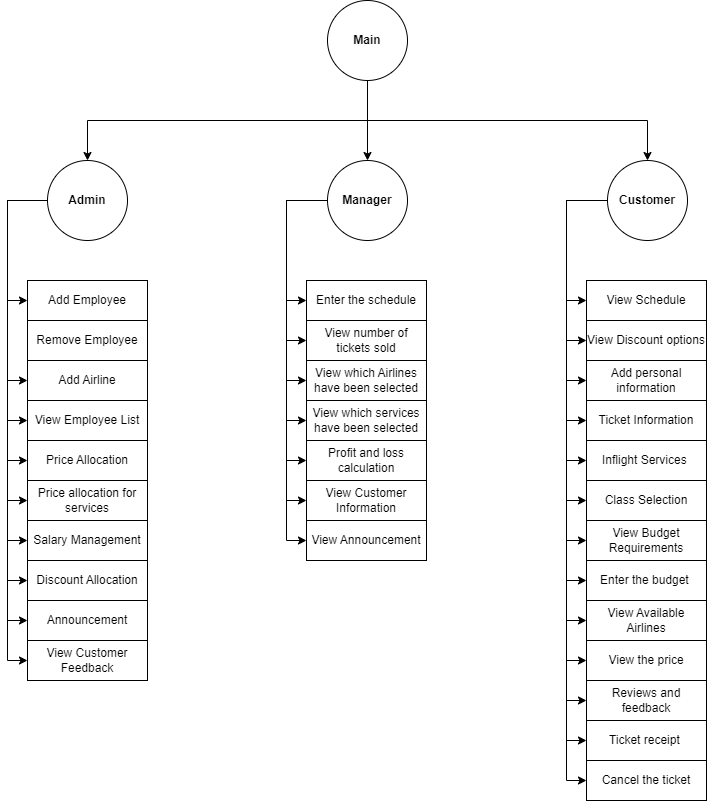


Figure 39

# Flowchart



# Data structures:

* Int arrsize =100;
* String Username [arrsize];
* String password [arrsize];
* string roles[arrsize];
* int usercountforCus = 100
* int signIncount[100] ;
* string nameOfUser[usercountforCus],
* country[usercountforCus],
* ID[usercountforCus],
* leg[usercountforCus],
* food[usercountforCus];
* string date[usercountforCus];
* string departure[usercountforCus], arrival[usercountforCus];
* string genderOfUser[usercountforCus;
* string ageofuser[usercountforCus];
* string clas[usercountforCus], customerreviews[usercountforCus];
* string employee[100];
* string IDNum[100];
* string airline[100];
* string aircode[100];
* string usernameForEmployee[arrsize];
* int budgetforCus[usercountforCus];
* string dayOfDeparture[usercountforCus];
* int discountcount = 100;
* string discountonTicket[discountcount];
* string arrivalcount[arrsize];
* string departurecount[arrsize];

# Function prototypes:

* void printHeader();
* string menu(string op);
* void output(string Name[], string departure[], string arrival[], string legspace[], string extrafood[], string clas[], string Date[], string ID[], float price, int logincount, string reviews[], string gender[], string age[], string country[], bool budgetresult, string dayofdeparture[], bool discountavail, string airline[], int ticketprice, int budgetforCustomer);
* int cal1(string leg[], string food[], int logincount, int totalprice, int ticketprice, string clas[], int business, int legspace, int foodservice);
* string login(string login);
* string signIn(string name, string password, string users[], string passwords[], string roles[], int usersCount);
* bool signUp(string name, string password, string role, string users[], string passwords[], string roles[], int &usersCount, int userArrSize);
* void personalInformation(string name[], string country[], string ID[], string gender[], string age[], int logincount);
* void flightInformation(string arrival[], string departure[], string date[], string dayOfDeparture[], int logincount,string availableDepartue[],string availableArrivals[],int departurecount,int arrivalcount);
* string classSelection(string clas[], int logincount);
* void InflightServices(string leg[], string food[], int logincount);
* string reviews(string reviews[], int logincount);
* string OwnerInterface(string op);
* bool Addemployee(string name, string ID, string employee[], string IDNUM[], int &logincount,string username, string usernameforemployee[]);
* int totalprice(int total);
* string subMenuBeforeMainMenu(string message);
* bool removeEmployee(string removingEmployee, string name[], string ID[], int &logincount, string usernameForEmployee[]);
* void addAirline(string airline[], string airlinecode[], int &aircount);
* void customerReviews(string customerReviews[], int logincount);
* bool budget(int budget, int ticketprice);
* void availableAirline(string airline[], int &count);
* void selectairline(string airlines[], int airlineChoice, int aircount,string selectedairlines[],int logincount);
* void employeeList(string employee[], string IDnum[],string username[], int &logincount);
* int pricealloaction(int &ticketprice);
* void extraCharges(int &business, int &legspace, int &food);
* int salaryfixation(int &salary);
* int salarymanagement(string employee[], int count, int &totalrevenue, string IDnum[], int salary);
* void discount(string discount[], int &dayscount);
* void viewDiscountOptions(string discount[], int dayscount);
* string discountday(string discountday[], int dayscount, int option);
* float percentageOfDiscount(float &percentage);
* bool isDiscountAvailable(string day[], int logincount, int daysofdiscount, string discount[]);
* float discountedvalue(int &totalprice, float percentage);
* string Employeeinterface(string op);
* int ticketscount(int logincount);
* void viewselectedAirlines(string selectedairline[], int logincount);
* void viewselectedServices(string leg[], string food[], string clas[], int logincount);
* void profitandLoss(int &totalrevenue);
* void announcementToEmployee(string &announcement);
* void viewAnnouncements(string announcement);
* void cancelllationOfTicket(string name[], string ID[], string gender[], int &totalrevenue, string age[], string departure[], string arrival[], string leg[], string food[], string clas[], string date[], string day[], int &price, string reviews[], int logincount, string airline[]);
* bool check\_number(string check, int length);
* bool check\_int(string check);
* bool checkeEmpty(string check);
* void setColor(int color);
* void printaAeroplane();
* void clearscreen();
* void printscreen();
* void viewCustomerInformation(string name[], string ID[], string country[], string gender[], string age[], string arrival[], string departure[], string date[], string day[], int logincount);
* bool checkstring(string check);
* void signUpinterface();
* void signininterface();
* void adminMenu();
* void customerMenu();
* void managerMenu();
* void loading();
* bool checkSpace(string check);
* void loginmenu();
* bool checkPlaces(string arrival[], string departure[], int logincount);
* void checkUser(string name, string username[] , int &logincount);
* void outputAgain(string Name[],string country[],string gender[],string age[],string departure[],string arrival[],string legspace[],string extrafood[],string Date[],string dayofdeparture[],string ID[],string clas[],string airline[],string reviews[],int logincount);
* bool checkCNIC(string ID[],string IDnum,int logincount);
* void shedule(string departure[],string arrival[],int &logincountfordeparture, int &logincountForarrival);
* void viewShedule(string arrival[],string departure[],int &departurecount,int &arrivalcount);
* bool checkDeparture(string availableDeparture[],string departure,int logincount,string arrival[],int logincount1);
* bool checkArrival(string availableArrival[],string arrival,int logincount,string departure[],int logincount1);
* string getField(string record, int field);
* void readData(string names[], string passwords[],string roles[], int &idx);
* void readDataForSignUp(string name,string password,string role,int &usersCount);
* void addEmployeeData(string name,string ID,string usernameForEmployee[],int logincount);
* void AirlineData(string airline,string airlineCode);
* void addPrice(int ticketprice);
* void addExtraCharges(int business,int legspace,int food);
* void discountData(string discount[],int discountcount);
* void announcementData(string announcement);
* void sheduleData(string departure[],string arrival[],int departurecount,int arrivalcount);
* bool checkSpaces(string var);
* bool checkSpaces(string var) ;
* void addEmployeeDataAfterRemoval(string employee[],string ID[],string username[],int &logincount);
* bool removeEmployeeData(string removingEmployee,string employees[],string ID[],string username[],int &logincount);
* void printBudgetRequirements(string leg[], string food[], int logincount, int totalrevenue, int &ticketprice, string clas[], int business, int legspace, int foodservice);
* void readExtraCharges(int &clas,int &leg,int &food ) ;
* void daysData(int departurecount,int arrivalcount);
* void readDiscountPercentage(float &discountpercentage);
* void totalrevenueData(int totalrevenue);
* void readTotalrevenue(int &totalrevenue);
* void addTickestCount(int logincount) ;
* void readLogincount(int &logincount);
* void readReviews(string reviews[],int &var) ;
* bool checkDays(string check);
* void cancelTicketMessage();
* void storeUserData(string name,string Name[],string country[],string gender[],string age[],string departure[],string arrival[],string legspace[],string extrafood[],string Date[],string dayofdeparture[],string ID[],int price,string clas[],string airline[],string reviews[],int logincount,int signIncount[]);
* bool checkSignInAgain(string name,int &customercount,string Name[],string country[],string gender[],string age[],string departure[],string arrival[],string legspace[],string extrafood[],string Date[],string dayofdeparture[],string ID[],string clas[],string airline[],string reviews[]);
* bool checkRoles(string roles);
* void hideCursor();

# Weaknesses in my project:

The weaknesses in my project is that it does not hold many validations. Although it doesn’t crash on many times but it still have limitations. And also the customer is only able to sign in one time and is not able to update his information.

# Future directions:

In future I will be looking to make a better GUI based project that can have validations.so that it will be more user friendly using OOP.