

به نام خدا

مستند تحلیل سیستم :

گروه آسان پرواز - گروه ۸

اعضای گروه :

فاطمه صبائی

ملیکا نوروزبیگی

ملیکا عبدالهی

زهرا دهقانیان

زمان تحویل : 8/10/96

نام تحویل گیرنده : دکتر مجید نورحسینی

• System description

❖ سیستم آسان پرواز یک شرکت هواپیمایی می باشد که هدف از آن ارائه یک سیستم یکپارچه به منظور بهره گیری از تمامی خدمات یک سامانه هواپیمایی می باشد.

❖ مسافران می توانند برای استفاده از این سیستم ابتدا در سیستم ثبت نام کنند و با وارد کردن نام کاربری خود و رمز عبور از خدمات این سیستم استفاده کنند .

سیستم هواپیمایی خدماتی را در اختیار مسافران می گذارد :

۱- جست و جوی سفر مورد نظر خود

۲- رزرو کردن سفر

۳- خریداری سفر

۴- کنسل کردن سفر

در صورتی که مسافر بخواهد بلیتی را رزرو یا خریداری کند درخواستی به سیستم می دهد و سیستم این درخواست را به شرکت هواپیمایی منتقل می کند و در صورتی که اون پرواز توسط کسی دیگر رزرو یا خریداری نشده بود درخواست مسافر تایید خواهد شد . زمانی که مسافر می خواهد بلیت را خریداری کند باید از طریق سیستم به درگاه بانک متصل شود و پس از عملیات بانک رسیدی به سیستم ارسال می کند و سیستم هم بلیت را صادر کرده و برای مسافر ارسال می کند.

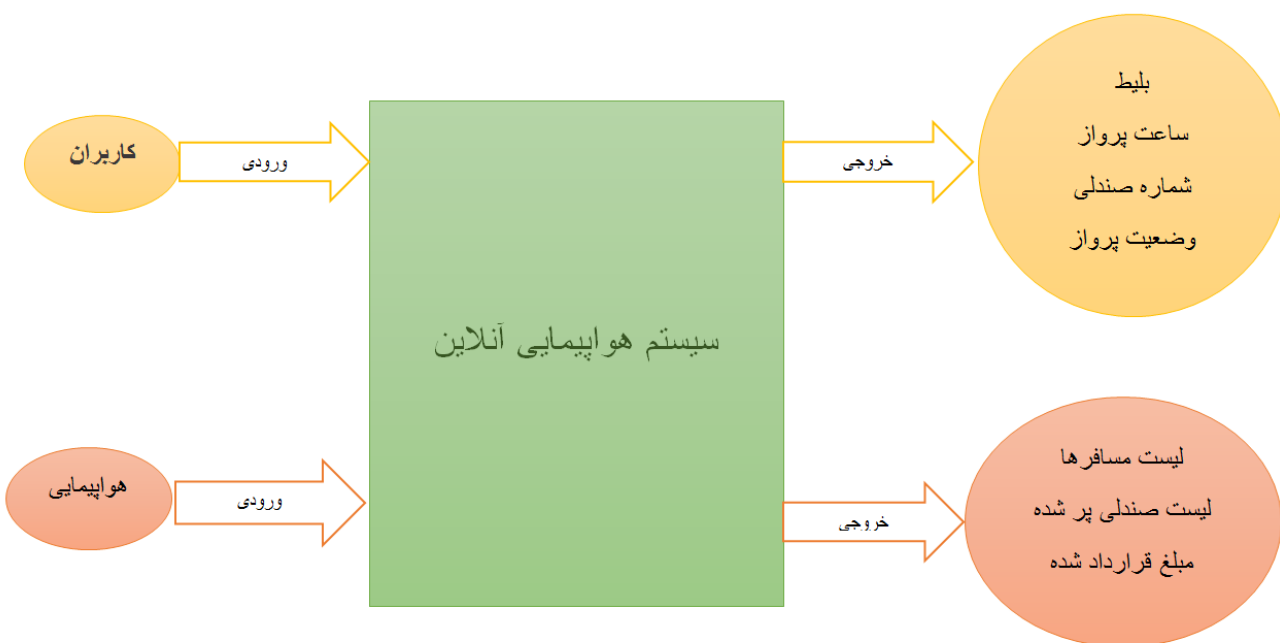
❖ شرکت هواپیمایی پس از وارد کردن نام کاربری و رمز عبور خود می تواند لیست پرواز هایش را برای سیستم ارسال می کند و اگر درخواستی از طرف سیستم به منظور رزرو بلیت یا خریداری بلیت یا کنسلی بلیت فرستاده شد می تواند پس از بررسی های لازم آن را رد یا تایید بکند.

❖ صاحب سیستم ماهانه سودی از سیستم دریافت می کند.

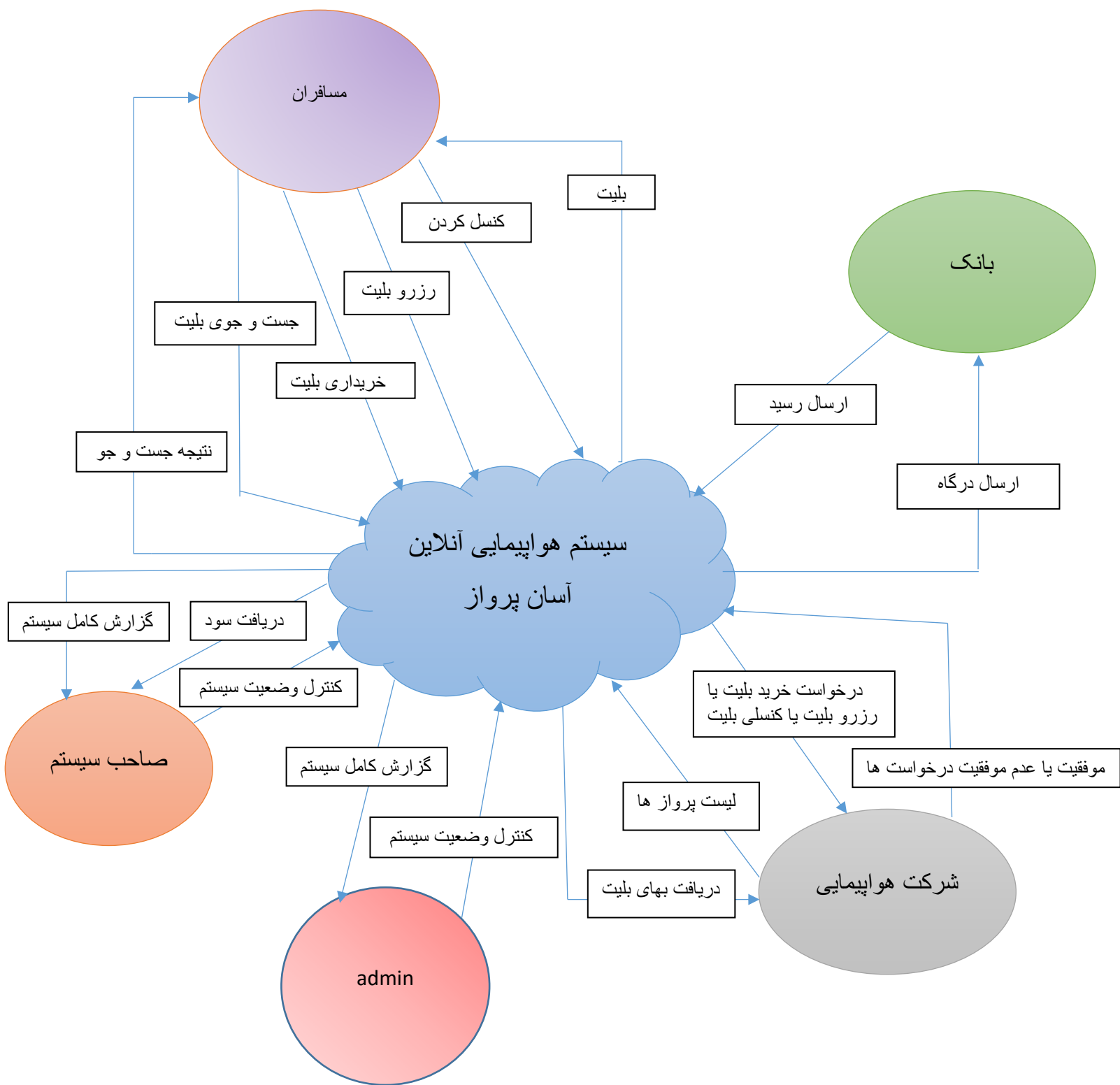
❖ ADMIN سیستم پس از واد کردن نام کاربری و رمز عبور خود می تواند پس از گرفتن گزارش کامل از تمام فعالیت هایی که سیستم انجام داده است وضعیت سیستم را کنترل کند.

- ورودی و خروجی سیستم :

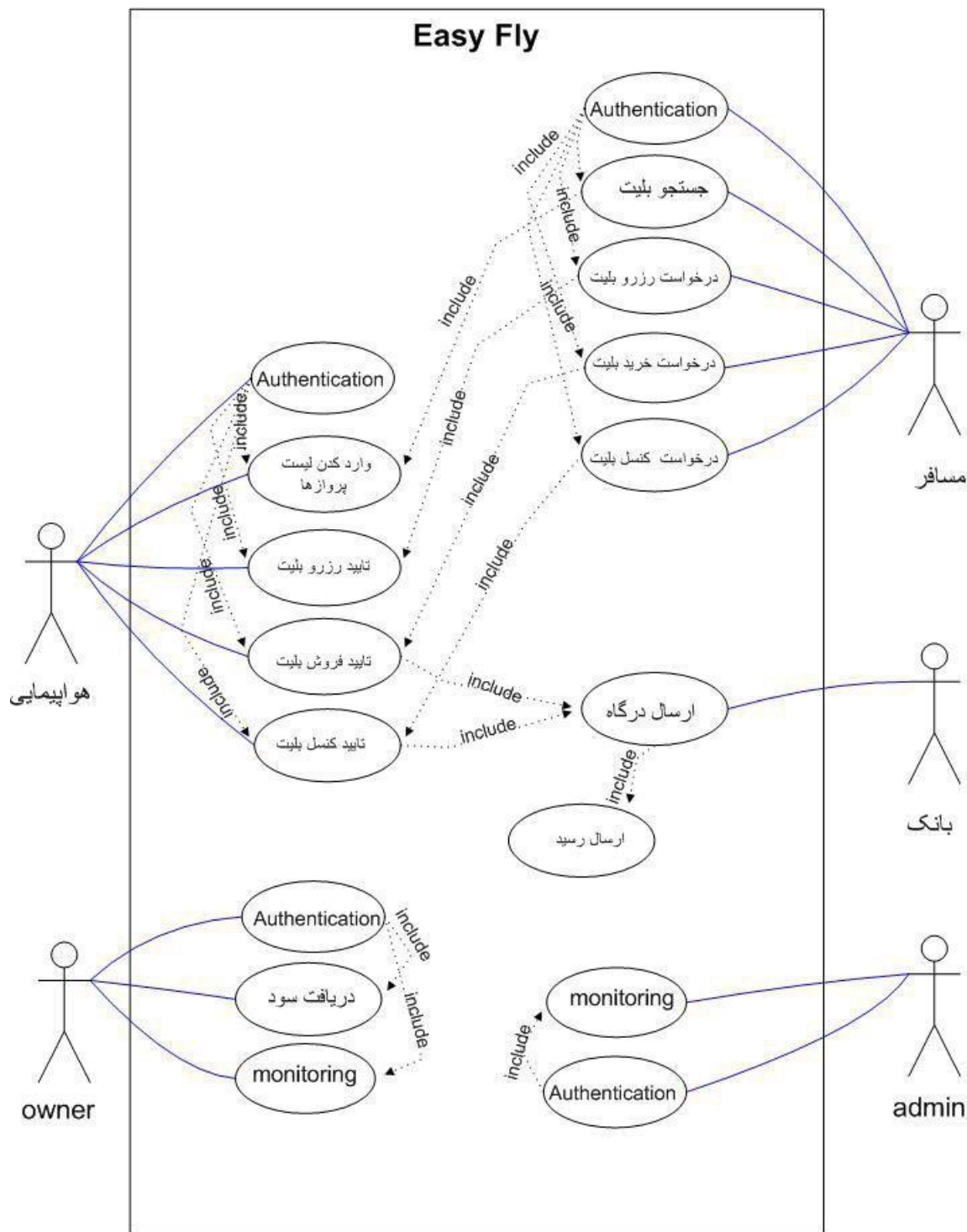
از کدام هر ازای به . قرارداد طرف هواپیمایی های شرکت و سیستم کاربران از عبارتند آنلاین هواپیمایی آژانس یک های ورودی این ورودی ها چند خروجی وجود دارد.



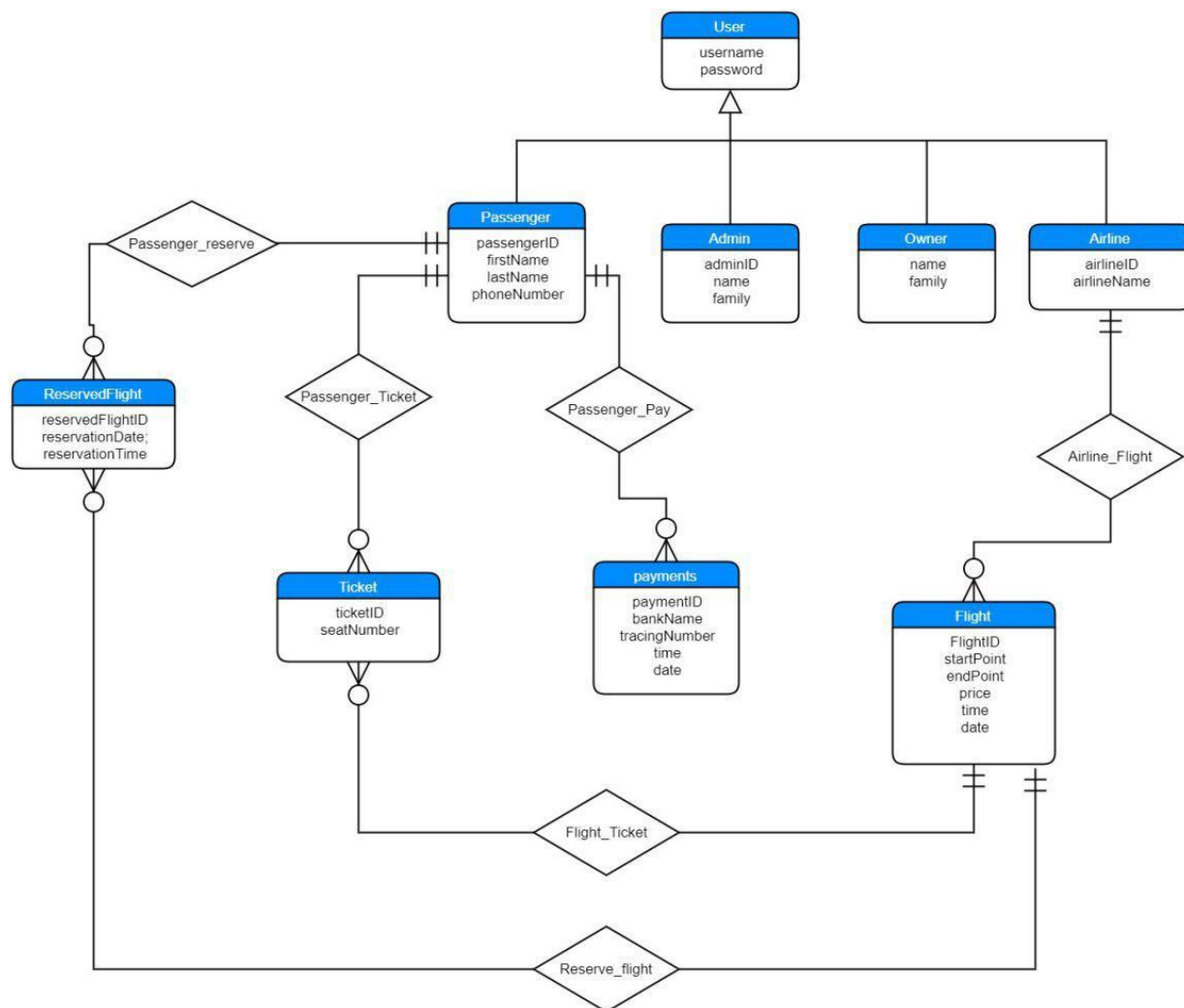
context diagram •



use case •



ERD •



• System architectur :

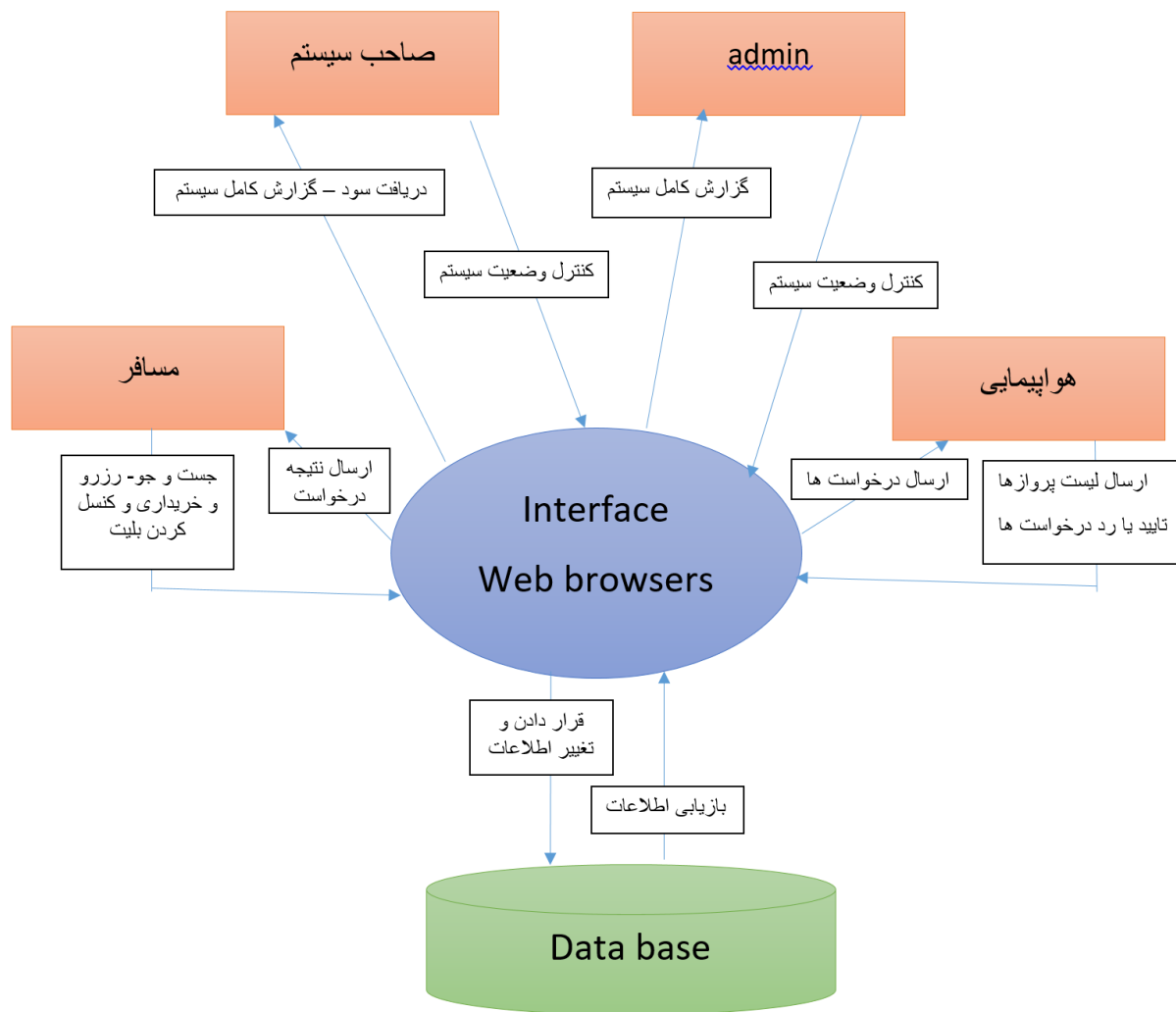
در این سیستم از معماری دیتا سنتریک استفاده شده است . دلیل این انتخاب این بوده است که در این سیستم داده ها و اطلاعات نقشی اساسی را ایفا می کنند و سایر اجزاء با داده ها به صورت مستقل و به دفعات ارتباط دارند در نتیجه باید از نوعی معماری برای سیستم استفاده کرد که در آن داده ها در یک مخزن مرکزی قرار دارند و سایر اجزاء به آن دسترسی دارند.

معایب :

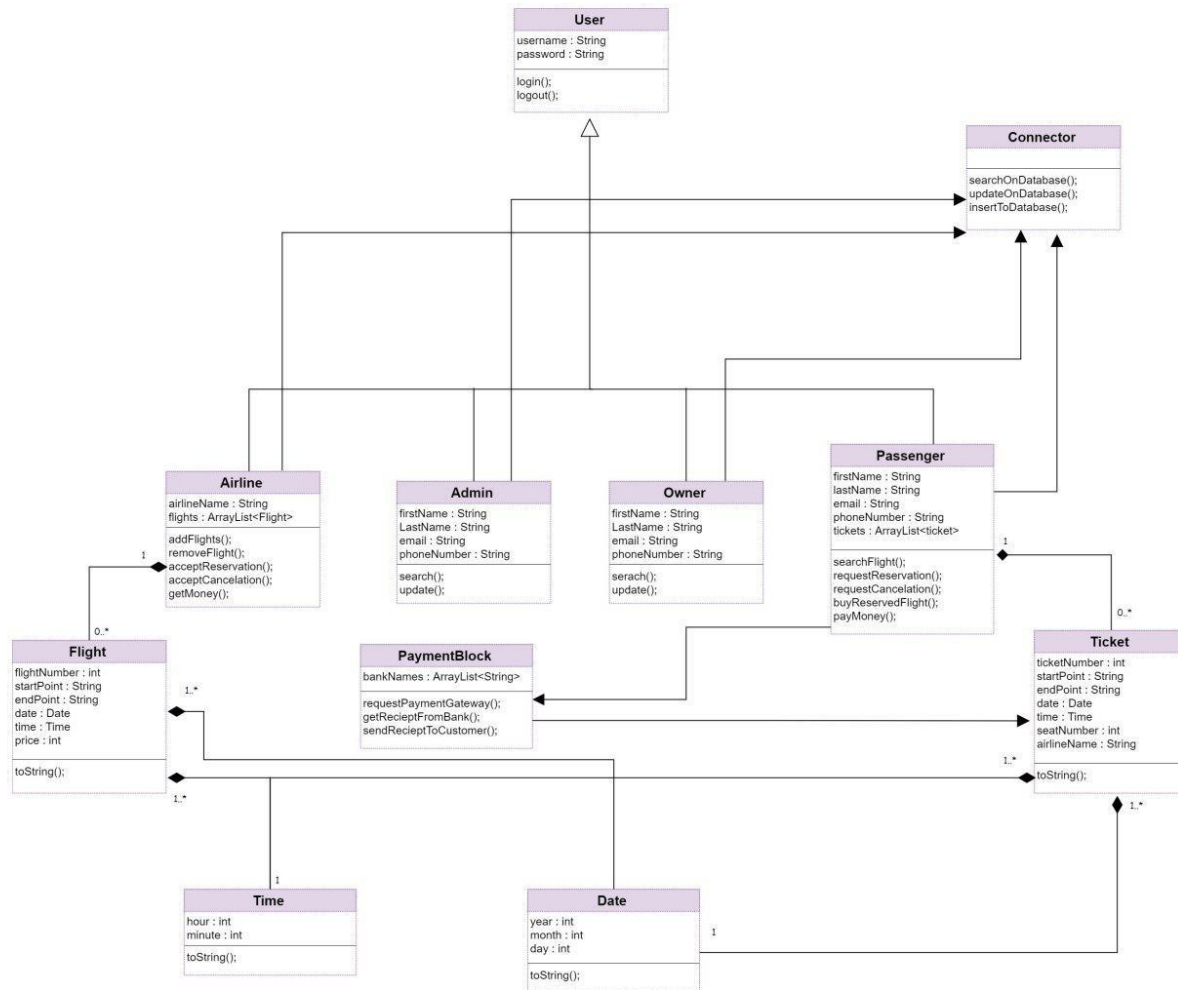
- وابستگی زیادی میان ساختار داده و برنامه های متصل وجود دارد .
- اگر بخواهیم تغییری در سیستم و اطلاعات انجام دهیم دشوار و هزینه بر می باشد.

مزایا :

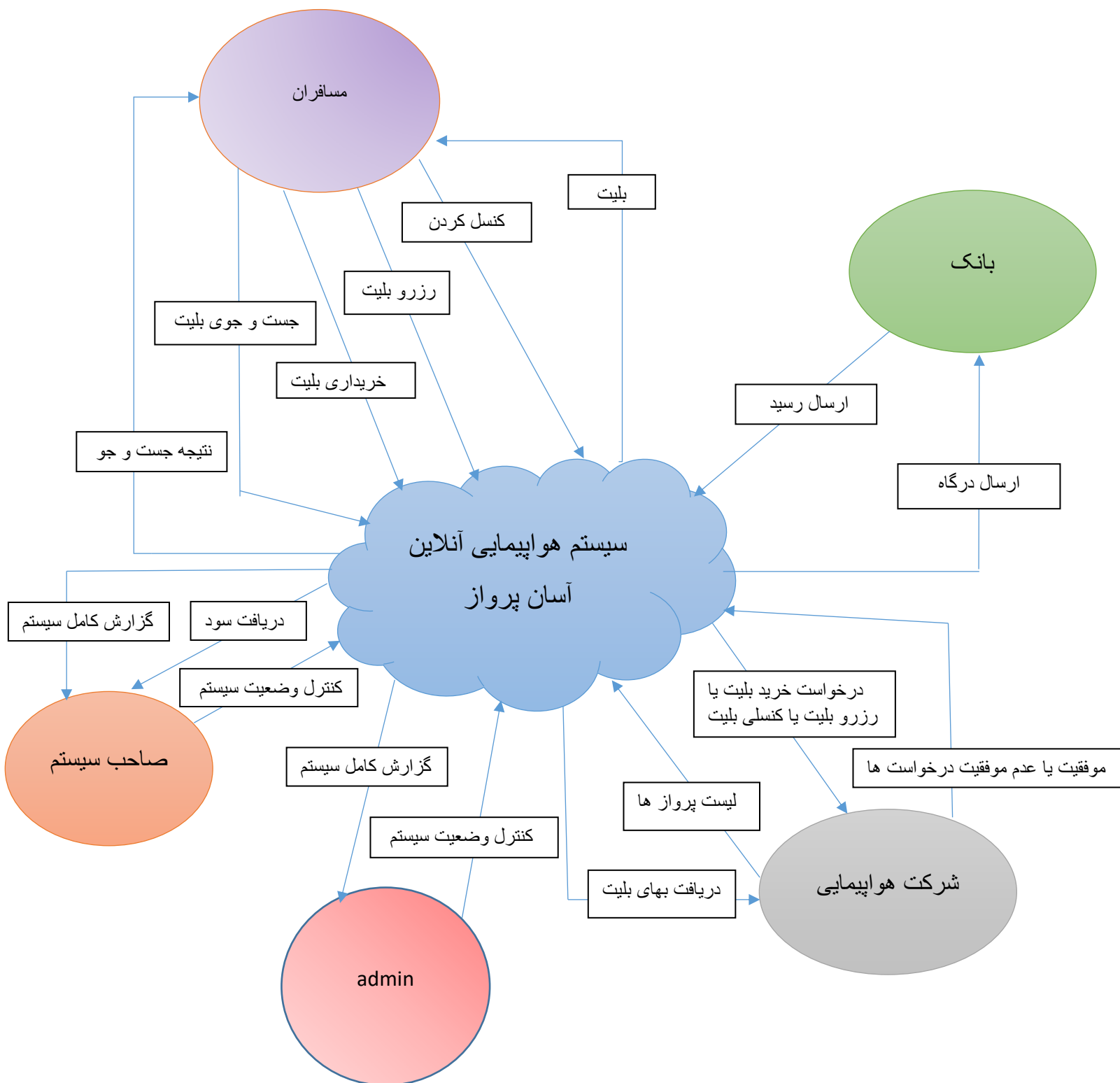
- می توان حجم بالایی از اطلاعات را با بازه بالایی ذخیره کنیم .
- معماری دیتا سنتریک مدیریت اطلاعات آسان تر می کند.
- باعث می شود سربار انتقال اطلاعات میان قطعات نرم افزاری را کاهش پیدا کند.



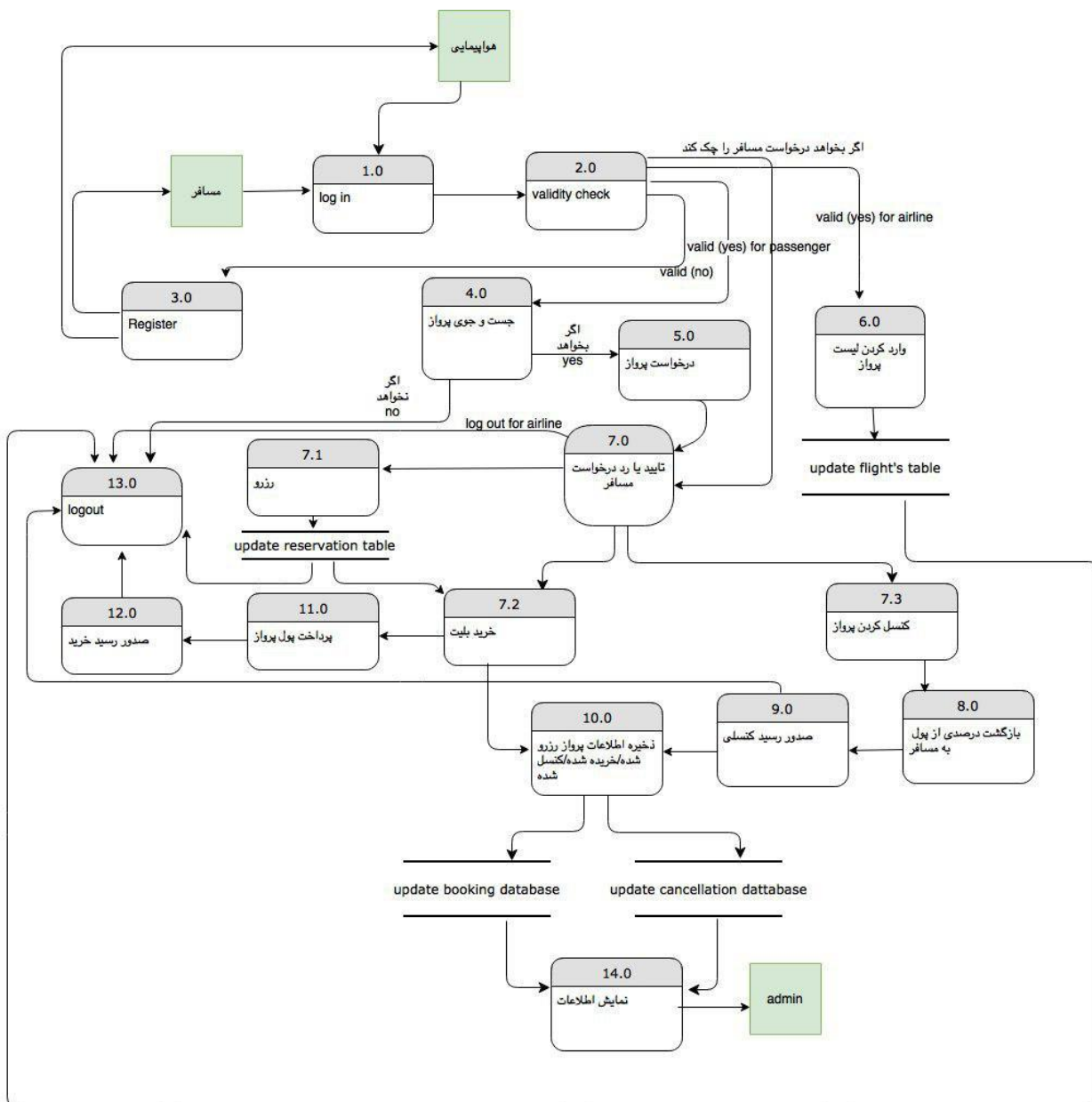
Class diagram •



DFD Level0 •

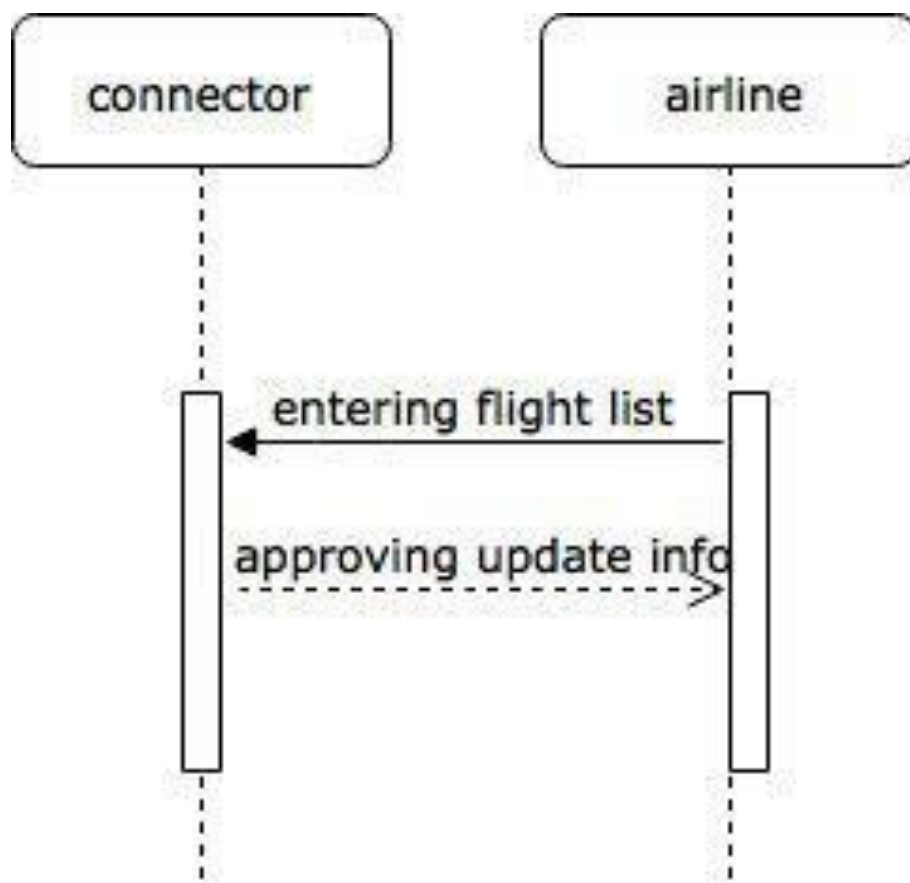


DFD Level1 •

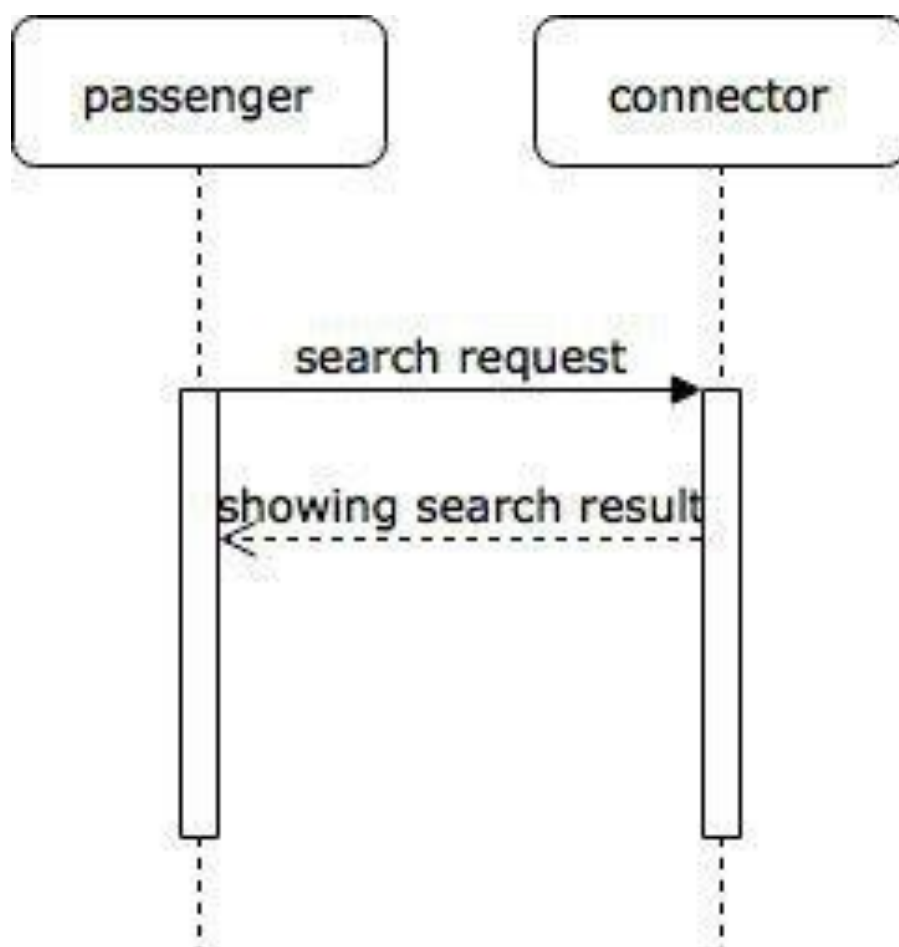


• Sequence diagram

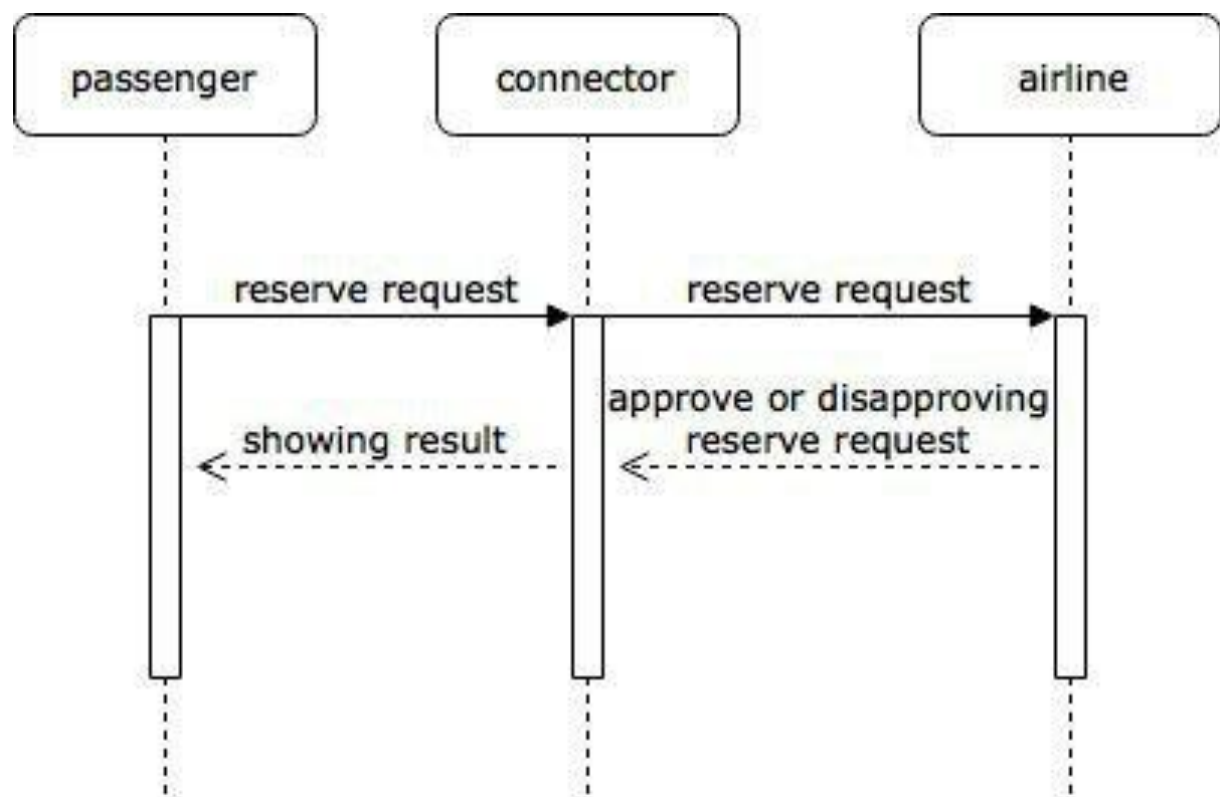
❖ شرکت هواپیمایی لیست پرواز را وارد سیستم می کند :



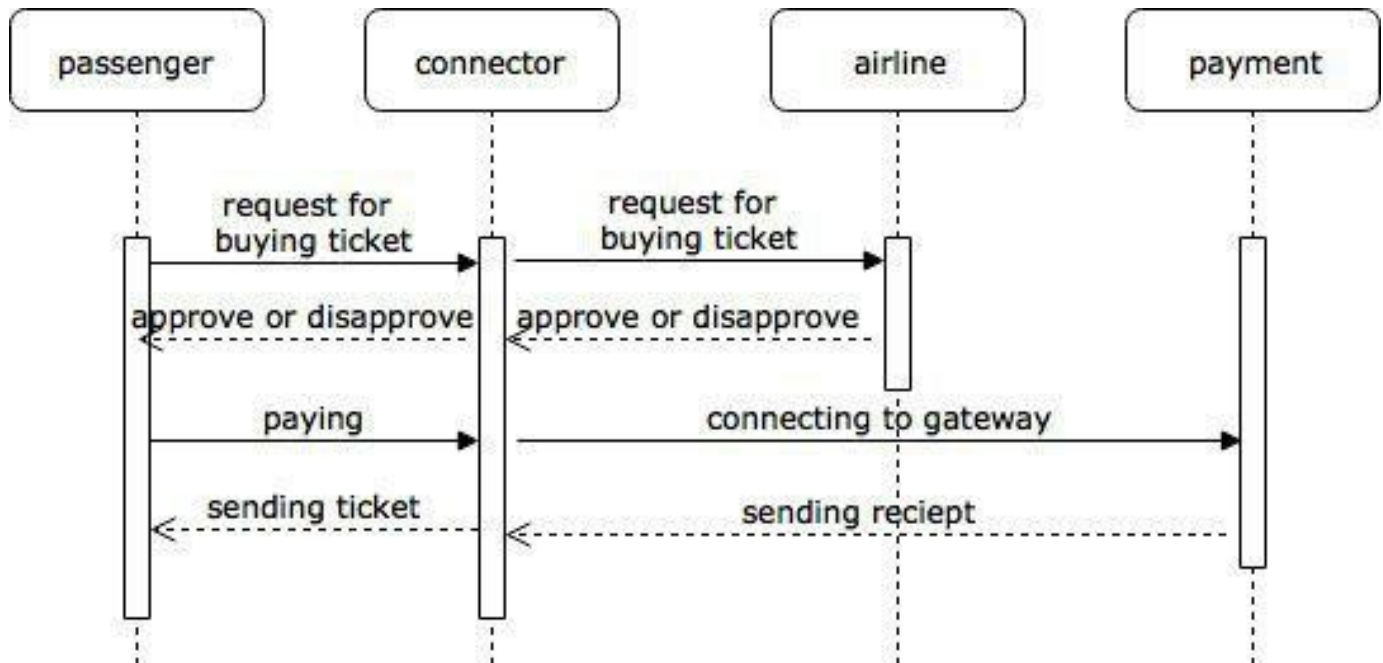
❖ مسافر مقصد خود را از سیستم جست و جو کند :



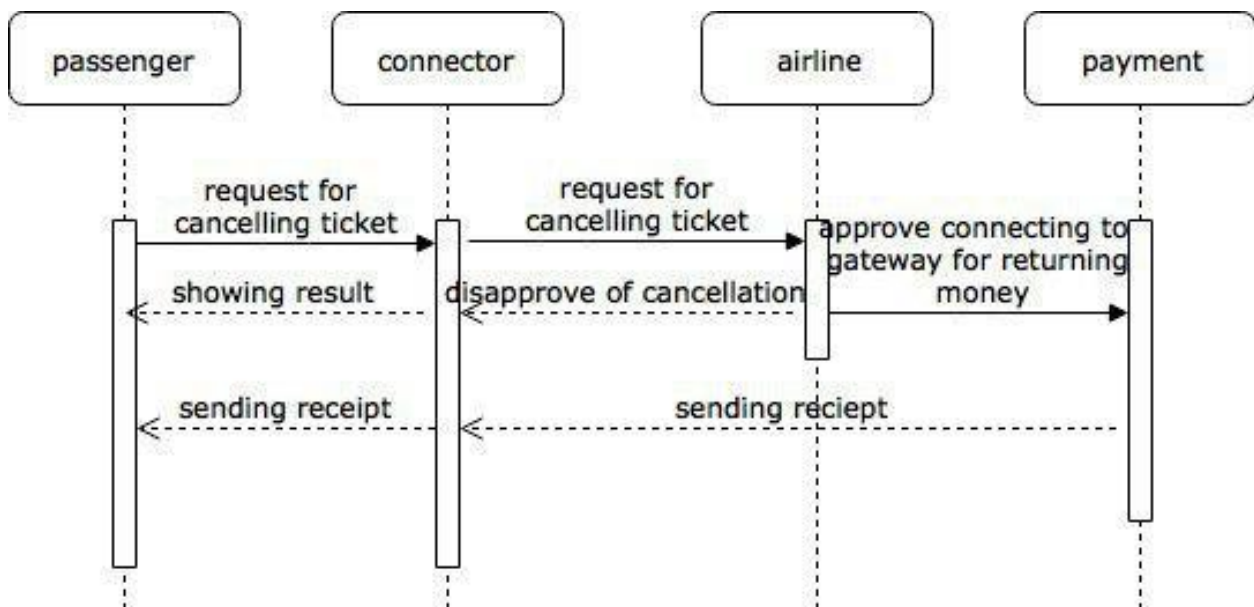
❖ مسافر می خواهد بلیت خود را رزرو کند :



❖ مسافر می خواهد بلیت خود را خریداری کند :



❖ مسافر می خواهد بلیت خود را کنسل کند :



Data dictionary •

Flight: class and entity

flightID:primary key, id of a flight

start point : origin of a flight

end point: destination of a flight

price: cost of a flight

time: the time that flight departures including hour and minutes.

Date: the date that flight departures including year, month and day.

Ticket:class and entity

ticketID:primary key, id of a ticket

start point : origin of a ticket

end point: destination of a ticket

seat number

time: the time that flight departures including hour and minutes.

Date: the date that flight departures including year, month and day.

Passenger: class and entity

Passengerid: primary key

Name: name of a passenger

Last name: last name of a passenger

Email

Phone number

Searchflight(): search a desired flight based on date and time

RequestReservation(): requesting to reserve a desired flight

RequestCancellation(): requesting to cancel a desired flight

BuyReservedFlight(): buying the flight that is already reserved

PayMoney(): paying the flight's price and getting the ticket

Airline: class and entity

Airlineid: primary key

Airline name: name of the airline

Flights: list of flights that airline offers

addFlights(): add the list of flights to the system

removeFlights(): remove the flights that has been cancelled

acceptReservation(): accepting the passenger's requested flight if the flight has capacity

acceptCancellation():accepting the passenger's request for canceling a flight if it is possible

Payment block: class

Bank names: names of the banks that system has contracts with them

RequestPaymentGateway(): request payment gateway from one of the banks

GetReceiptFromBank(): get the payment receipt from the bank

SendReceipttoCustomer(): sending the payment receipt for customer

Time: class

hour

minute

hour()

minute()

Date: class

date

date()

Admin: entity and class

First name

Last name

Email

Phone number

Owner : entity and class

First name

Last name

Email

Phone number

ReservedFlight : entity

ReservedFlightID

ReservationDate : the date that the passenger reserved the flight

ReservationTime : the time that the passenger reserved the flight

Connector : class

This class is an interface between users and database

SearchOnDatabase() : execute a query on database and search on it

UpdateOnDatabase() : execute a query on database and update it

InsertToDatabase() : execute a query on database and insert data into it

User: Entity and Class

Userid:primary key

Password: password for logging in

Login();

Logout();