**به نام** **خدا**

**اعضای گروه** : احمدرضا یزدانی ، امیرحسین زمانی لاری

**تمامی فعالیت ها در بستر گوگل میت و همزمان توسط دو نفر انجام شده است .**

**نام پروژه** : سیستم رزرواسیون بلیط هواپیما

**\*\*\*\*\*\*\*\*\*\*\*\*\* سیستم رزرواسیون بلیط هواپیما \*\*\*\*\*\*\*\*\*\*\*\***

**نیازسنجی ها :**

امروزه که دنیا به سمت الکترونیکی شدن پیش می‌رود تمام شرکت ها و آژانس هایی که خدمتی ارائه می‌دهند یا کالایی می‌فروشند به این فکر افتاده اند که کار خود را الکترونیکی کنند و آژانس های هواپیمایی نیز از این قاعده مستثنی نیستند.

امروزه ما به سامانه هایی نیاز داریم که مردم بتوانند از طریق آن  ، پرواز های موجود بین هر دو شهر در هر زمان را مشاهده کنند و برای خود و دیگران بلیط هواپیما تهیه کنند ،  به علاوه کاربران نیاز داشتند که بتوانند

لیستی از خرید های قبلی خود را نیز در دسترس داشته باشند.

**توضیحات تفضیلی نیازسنجی ها :**

**1) مشاهده بلیط های هوایی با مشخصات داده شده :**

بدین ترتیب که زمان، مبدا و مقصد پرواز به سامانه داده می شود و سامانه ما اطلاعات را از سامانه سراسری پرواز ها گرفته و به کاربر نمایش می دهد

**2) خرید بلیط :**

بدین صورت که کاربر آیدی بلیط و اطلاعات مسافر خود و اطلاعات کارت بانکی را به سامانه ما ارسال نموده سپس سامانه ما بلیط را در سامانه سراسری پرواز ها رزرو می کند ( تا شخص دیگری نتواند آن بلیط را خریداری نماید ) سپس اطلاعات کارت کاربر به سامانه پرداخت وارد می شود و پرداخت آن توسط بانک تایید شده در نهایت خرید بلیط در سامانه سراسری پرواز ها نهایی می شود، اطلاعات بلیط خریداری شده در سامانه ذخیره شده و یک رسید از بلیط برای کاربر ارسال می گردد.

**3) مشاهده خرید های قبلی :**

به نیت در دسترس داشتن آنلاین اطلاعات بلیط های خریداری شده، کاربر می تواند پس از ورود به سامانه ، اطلاعات خرید های قبلی خود را مشاهده نماید

**DFD level 1:**

A diagram of a diagram

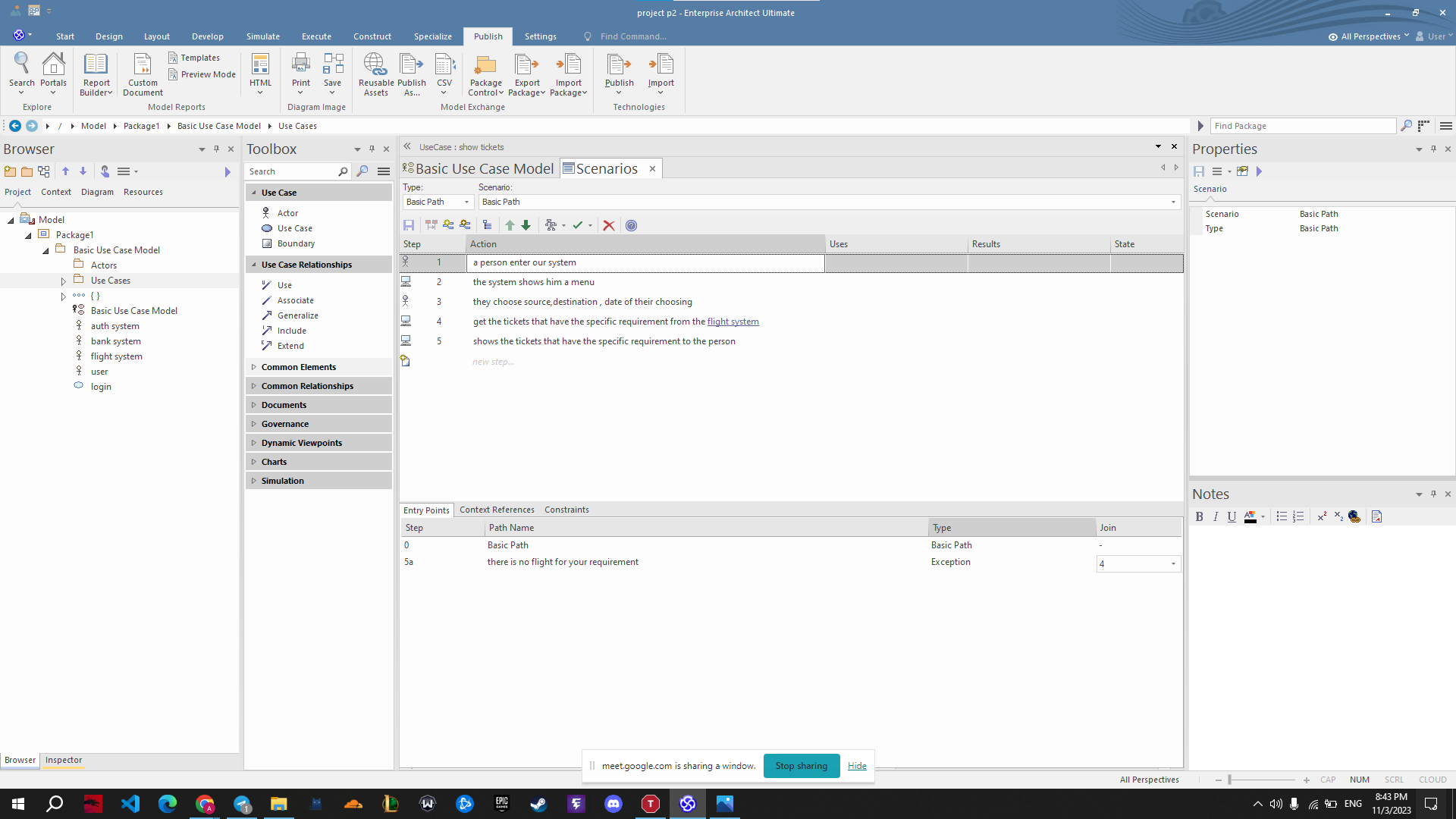
Description automatically generated

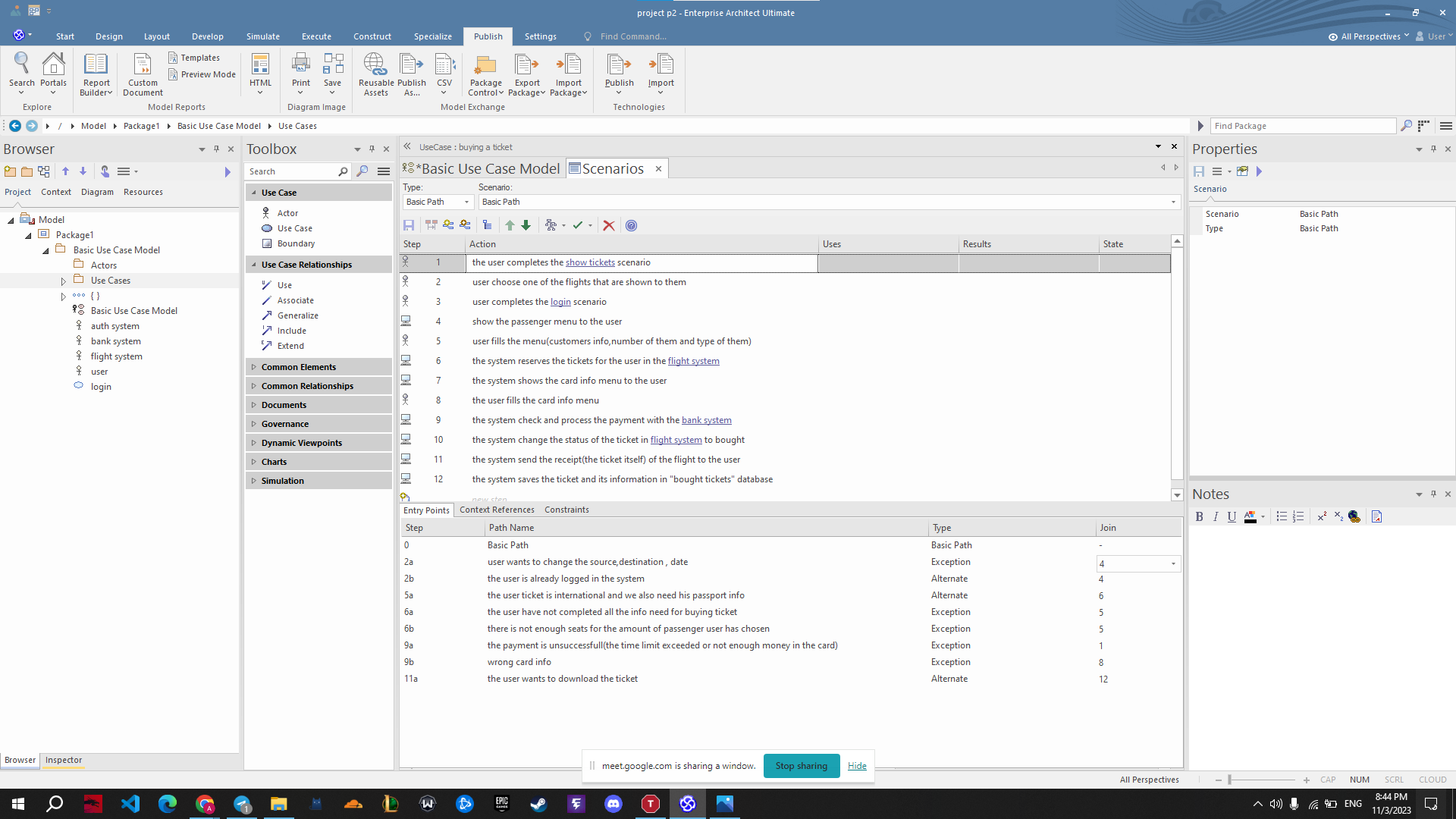
**Vision:  
Problem Statement:** We saw that there is a lack of flight reservation  
that has all the features necessary for easy and fast reservation of flight  
tickets, so we are making one.  
**Stakeholder Description:** Air lines, and we also own the project because  
it’s a startup.  
**Key High-Level Goals:** To have a complete system for reserving flights,  
that will also help to boost the sales of tickets for the airlines.  
**System Context Diagram**: \*in the zip file  
**Functional Features:** in our system we have:  
• A login and account making  
• Seeing all the flights with users limitations  
• Buying a ticket the ticket for the given flight  
• Seeing the history of all tickets the user has bought  
**Non-functional Features**: Our system has Reliability, Availability,  
Portability, Performance , Usability, Extendibility, Reusability.

A screenshot of a computer

Description automatically generated  
**Use case Model:**  
  
A diagram of a diagram

Description automatically generated  
  
  
**Show ticket scenario:**



**Buy ticket scenario:**

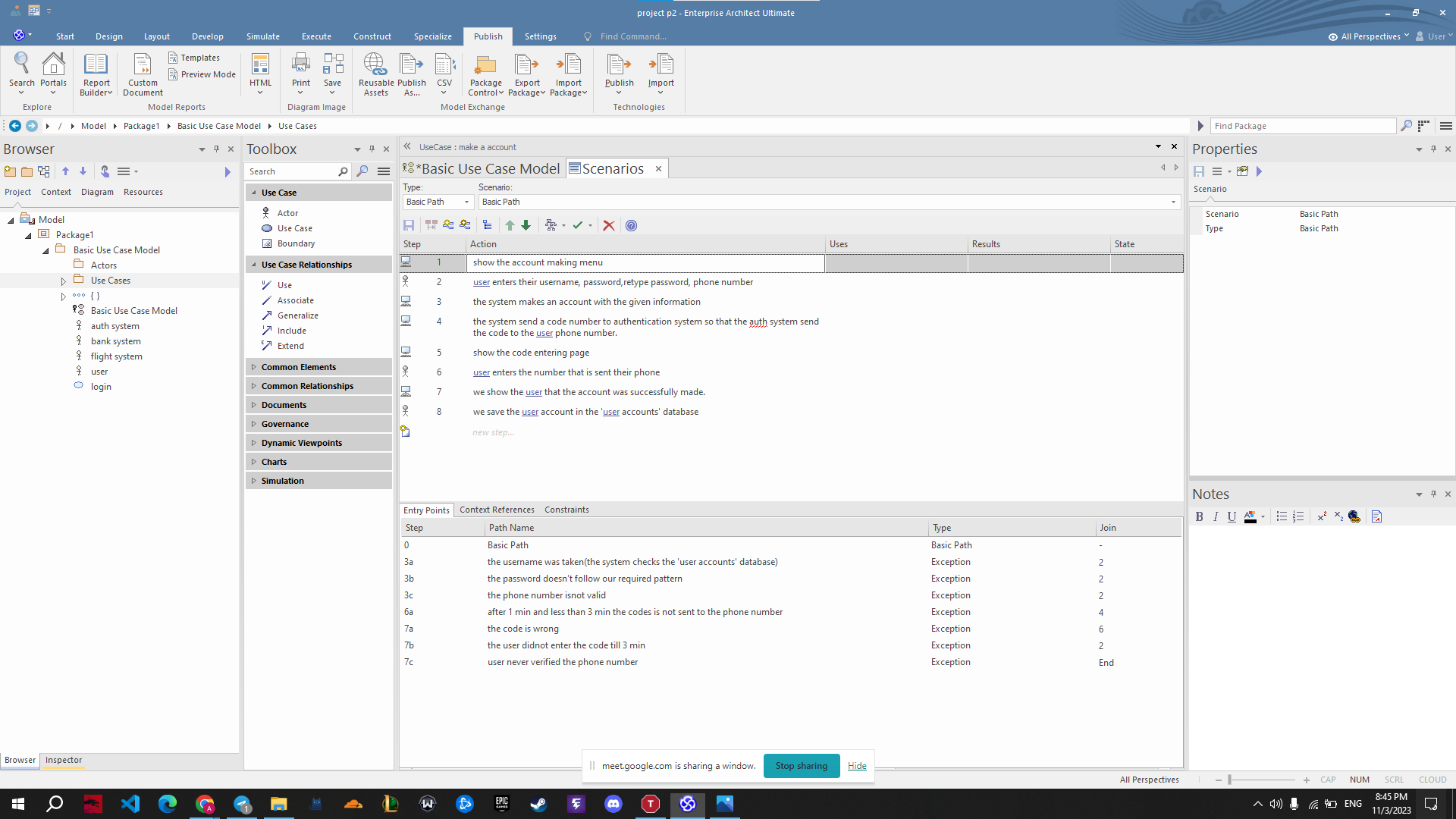
A screenshot of a computer

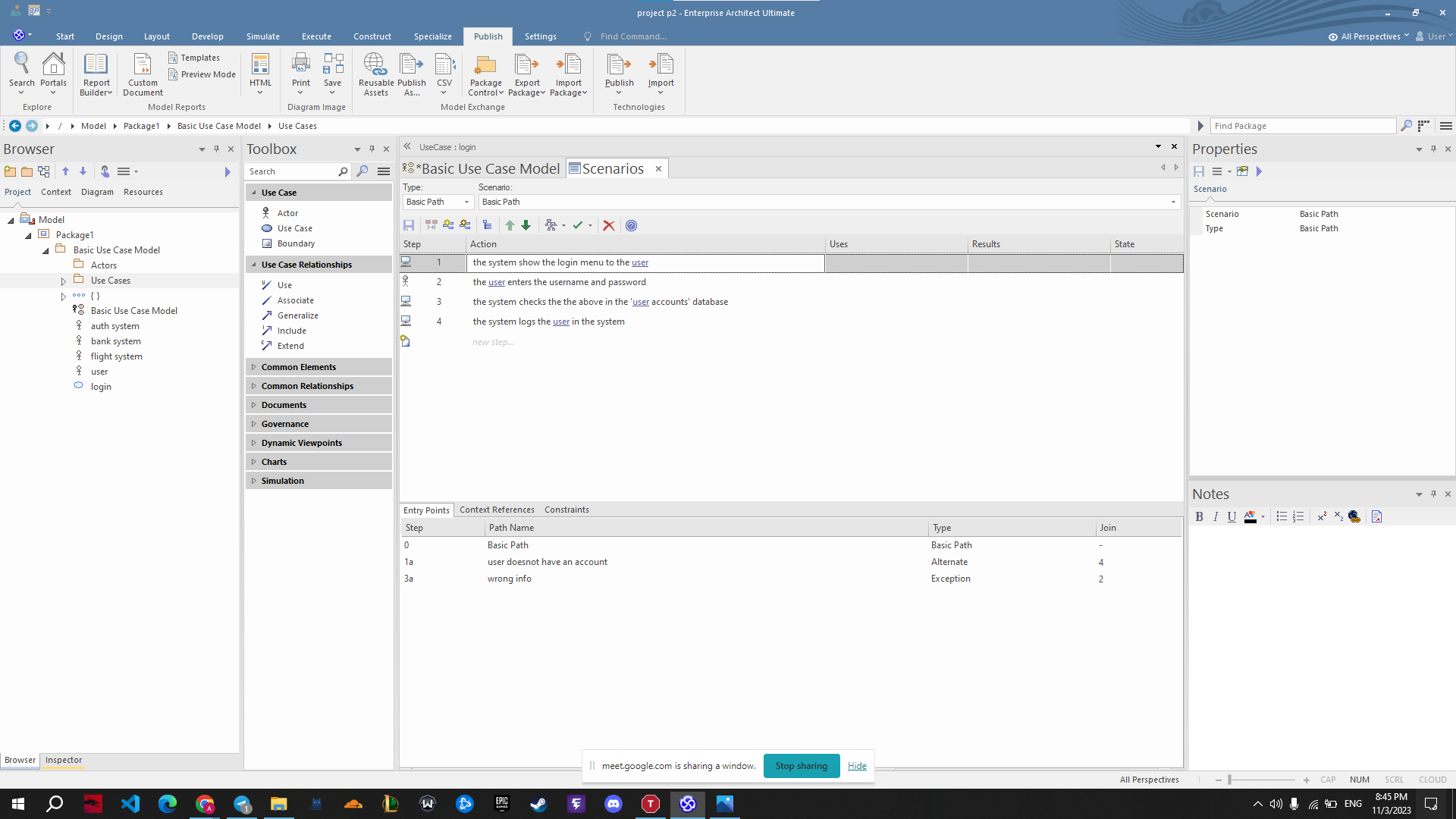
Description automatically generated

**Make account scenario:**

A screenshot of a computer

Description automatically generated

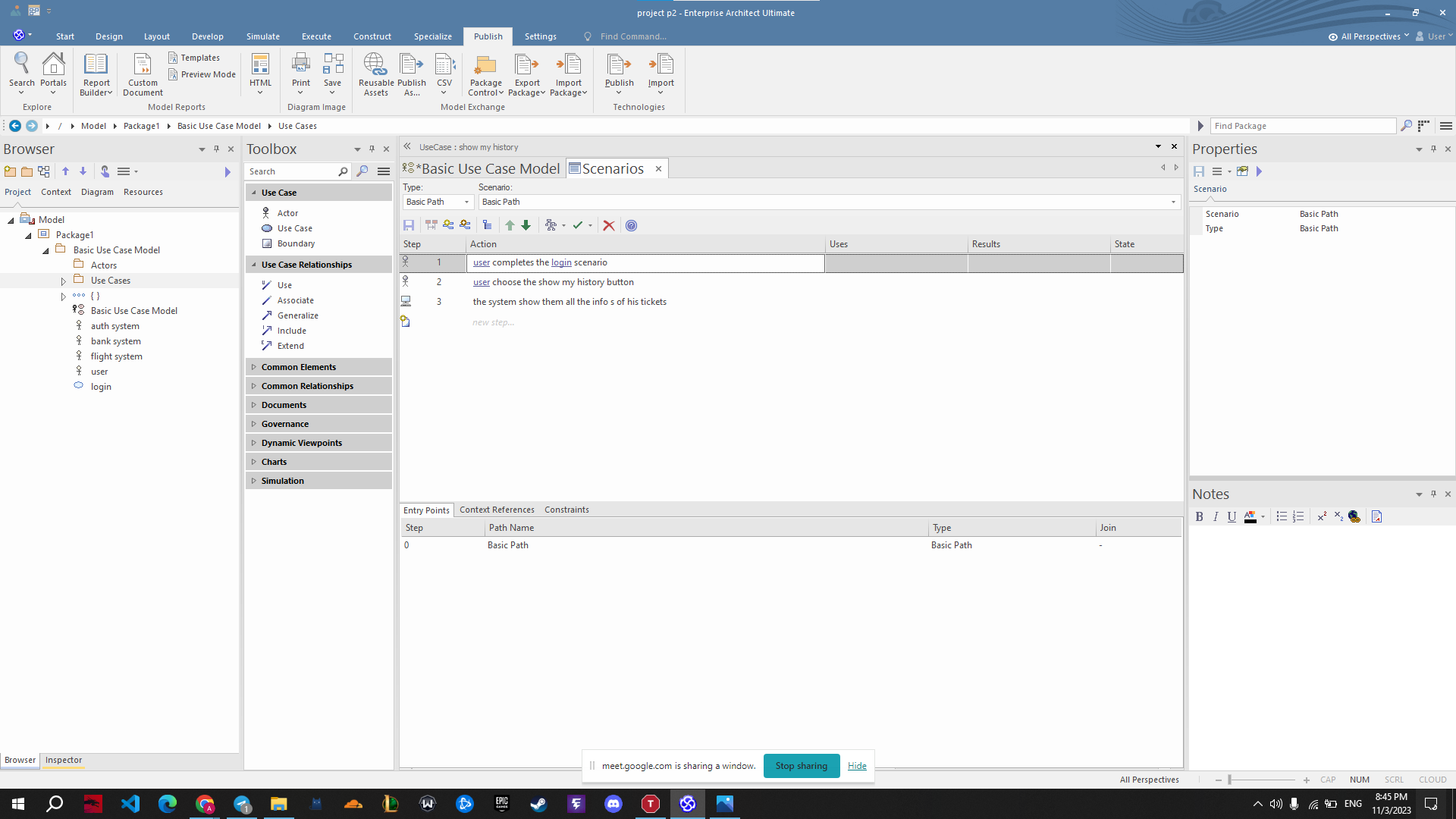


**Login scenario:**

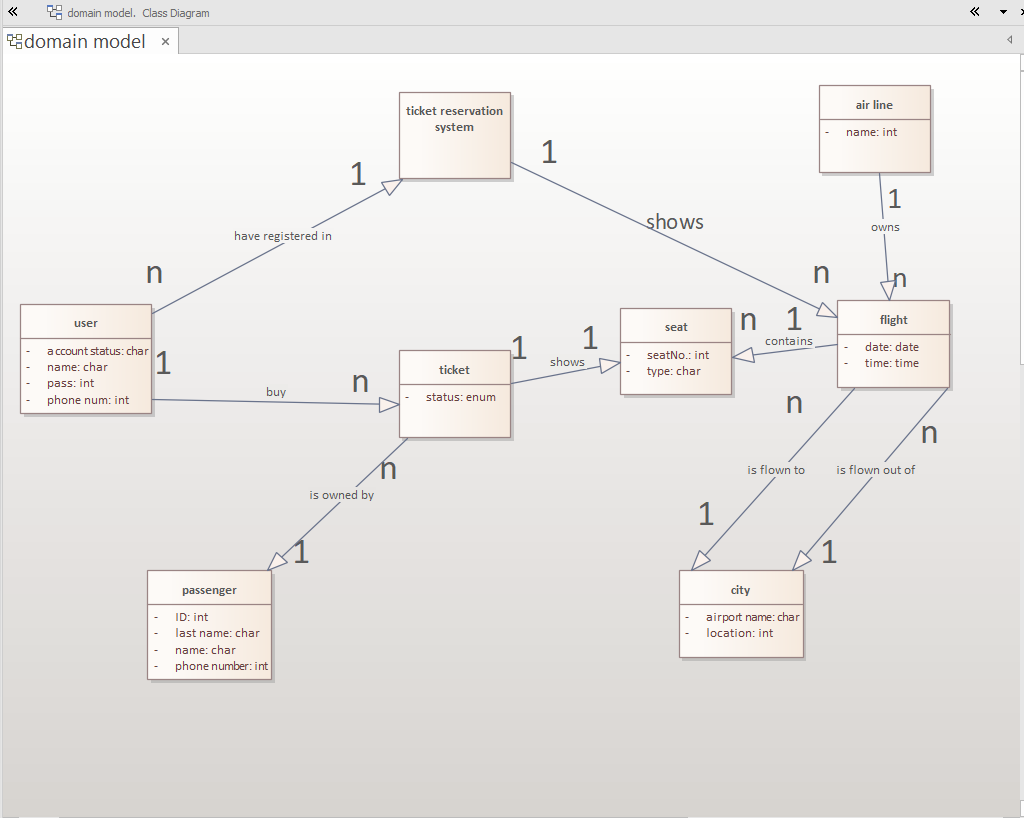
A screenshot of a computer

Description automatically generated

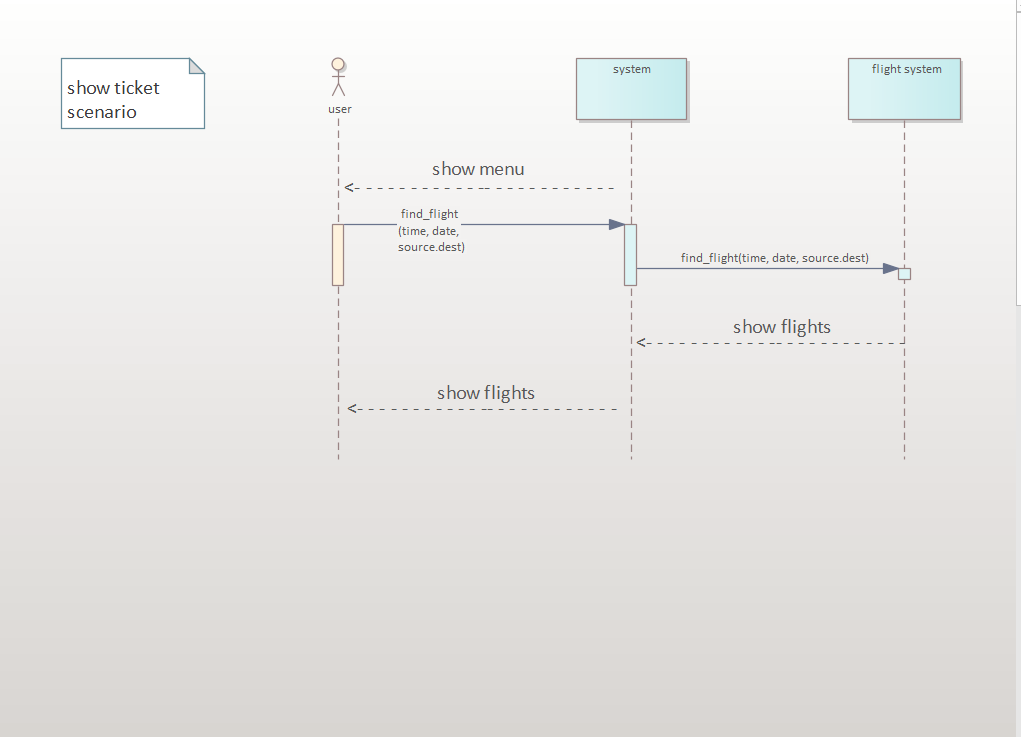
**Show history scenario:**



**Domain model:**



**SSD’s for scenarios:**

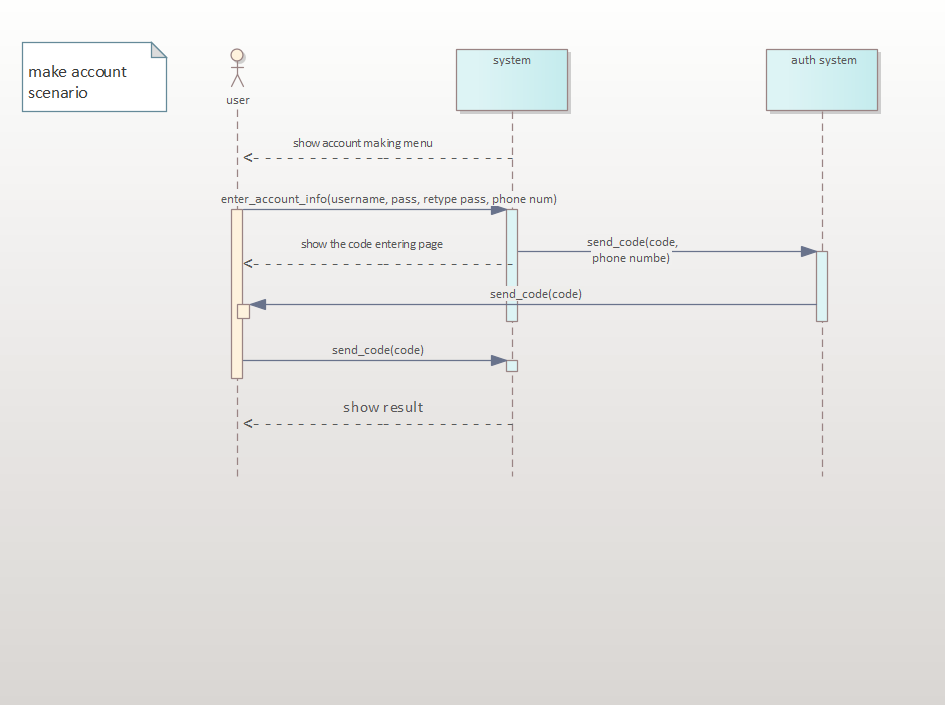


A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated

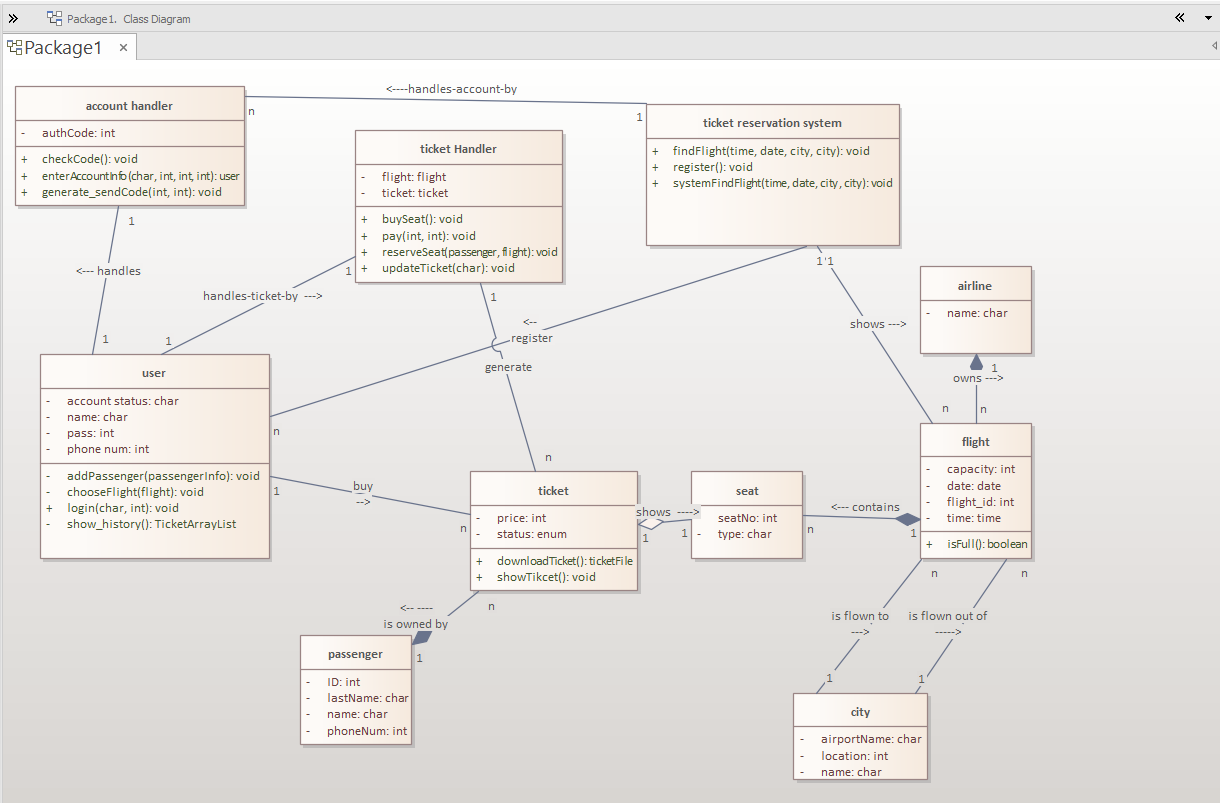
**Operation contracts:**

**Show ticket scenario:**1. Find\_flight(time , date, source , destination):  
Post condition:  
- Instance of F of ‘flight’ was created.  
- F was associated with ‘trs.  
Pre condition:  
- trs of ‘ticket reservation system’ exists.

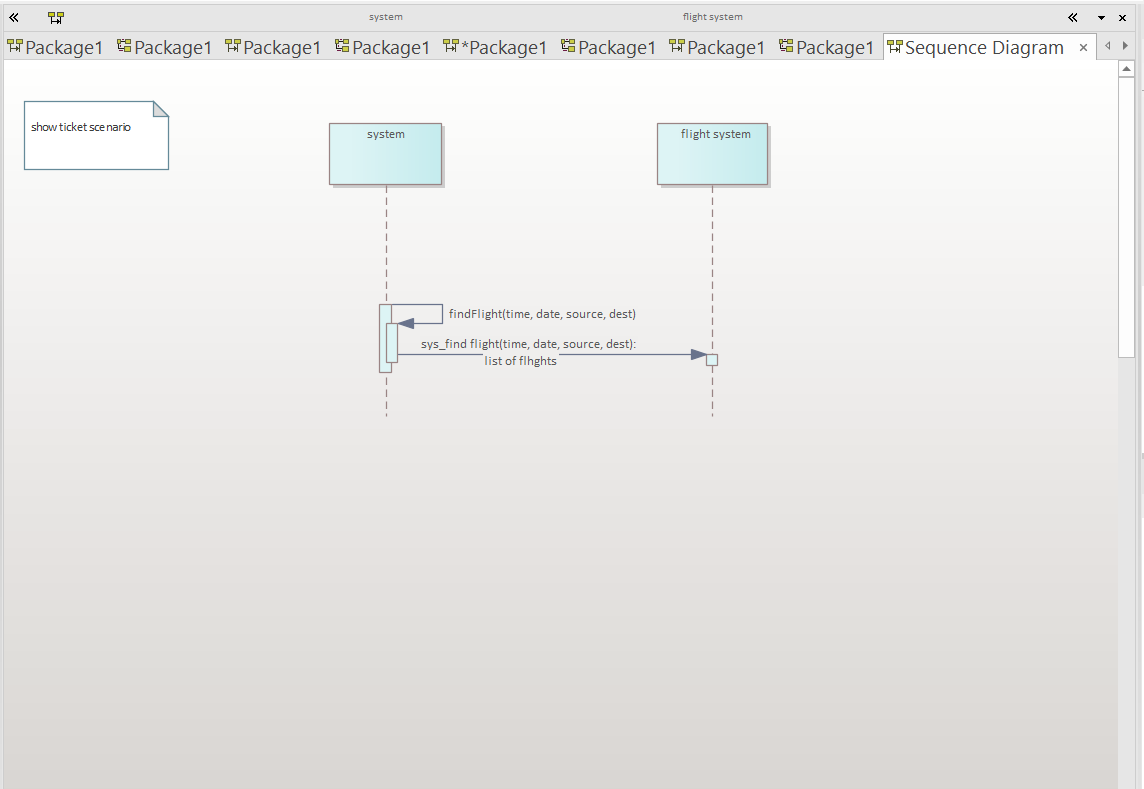
**Buy ticket scenario:**1. Add passenger(passenger):  
Post condition:  
- Instance of p of ‘passenger’ was created.  
- p was associated with t.  
Pre condition:  
- t of ‘ticket’ exists.  
2. Pay ticket(card \_info)  
Post condition:  
- status of t was set ‘bought’.  
Pre condition:  
- t of ‘ticket’ exists  
**Show history Scenario:**1. show history ():  
Post condition :  
- none  
Pre condition:  
- U of ‘user’ exists  
**Make account Scenario:**1. enter\_acount\_info(username, pass , retype pass,  
phone num )  
Post condition:  
- Instance of u of ‘User’ was created.  
- U was associated with trs.  
Pre condition:  
- trs of ‘ticket reservation system’ exists.  
2. send\_code(code)  
Post condition :  
- status of u was set ‘permanent’.  
Pre condition :

- u of ‘User’ exists

**Login scenario:**1. Login(username, password):  
Post condition:  
- none  
Pre condition:  
- U of ‘user’ exists

**Class diagram:**

**Communication diagram & sequence diagram:**



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Glossary:**

* Every time we have used the word ‘system’ alone , it means the system we are currently implementing , that is the ‘flight reservation system’.
* Every we have used ‘flight system’ , it is the central system that our system(flight reservation system) is getting its flight information from.
* ‘auth code’ is the code that we use to verify the user’s account by sending it to their phone number.
* The ‘status’ in the ‘ticket’ means that ,the ticket is ‘reserved’ or ‘bought’ or available.
* The ‘status’ in the ‘user’ means that , the account is ‘guest’ or is ‘unverified’ or ‘verified’ .