SOFTWARE SPECIFICATIONS

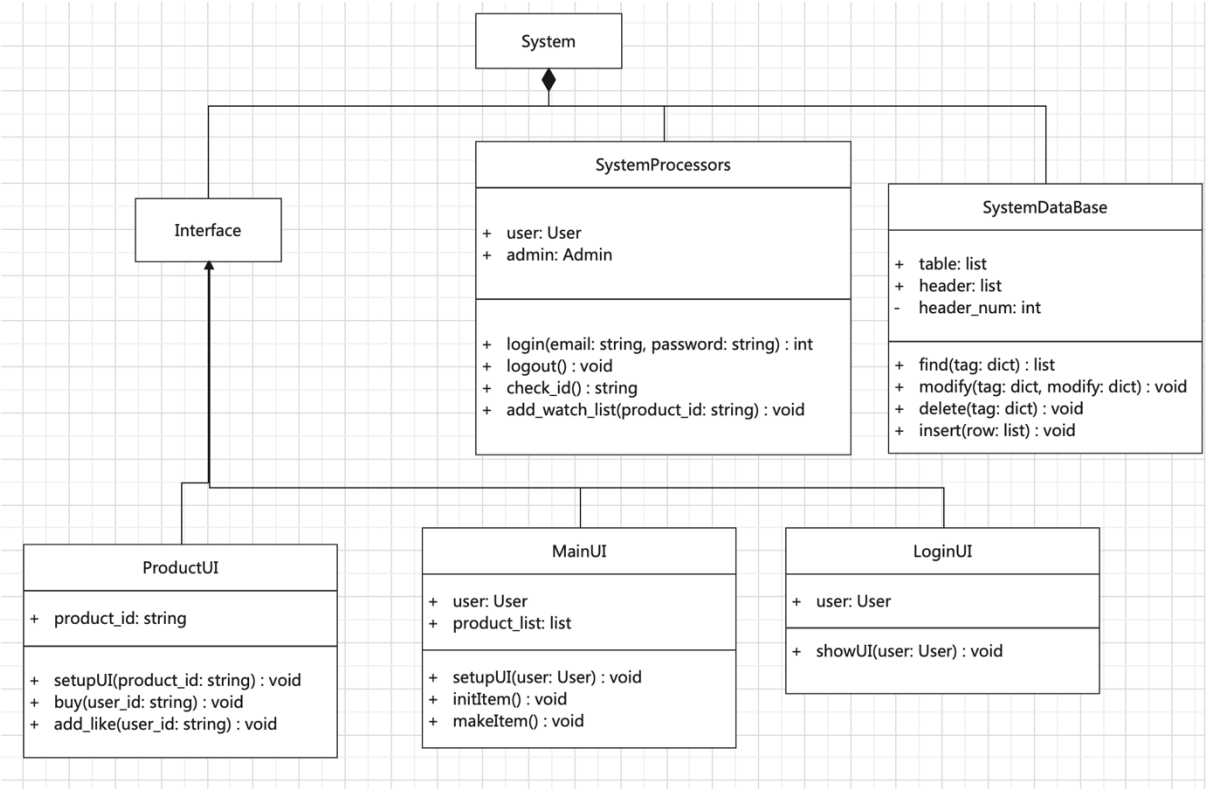
SHARK Second-hand Trading Platform

Group 4
舒梓轩 姜浩然 徐源松

Table of Contents

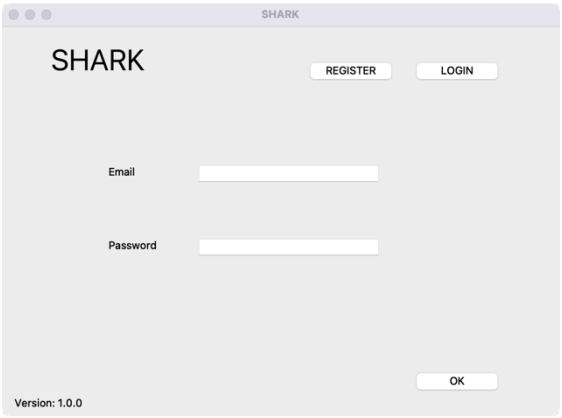
System Architecture

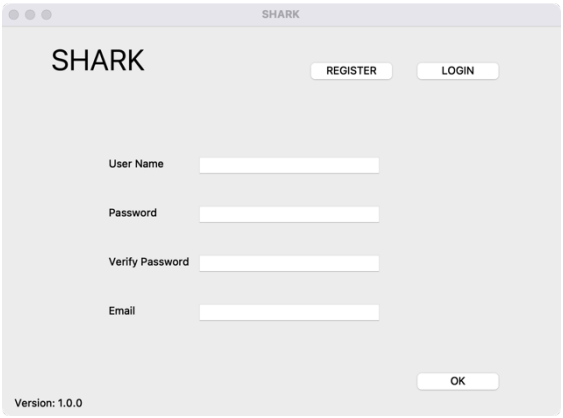
The system architecture is shown below:



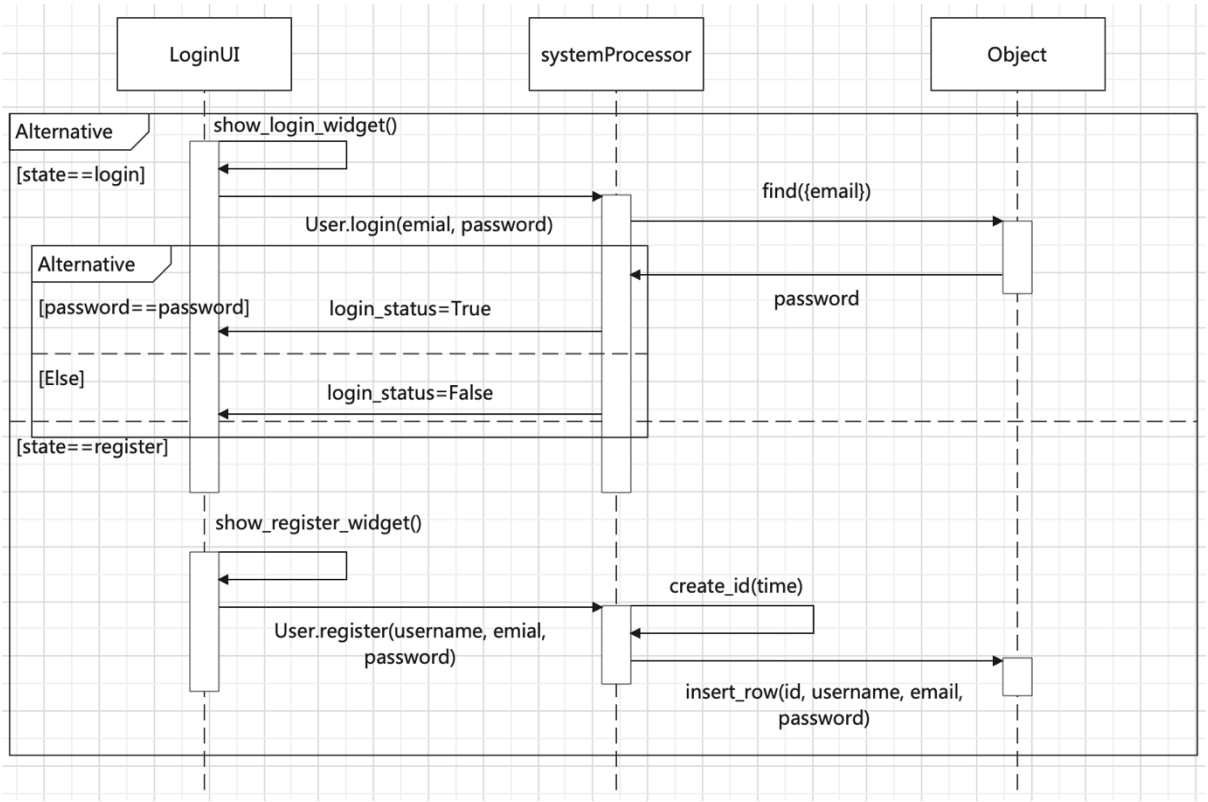
Software Specifications

S1: LoginUI implementation





S1.1: Select login/register

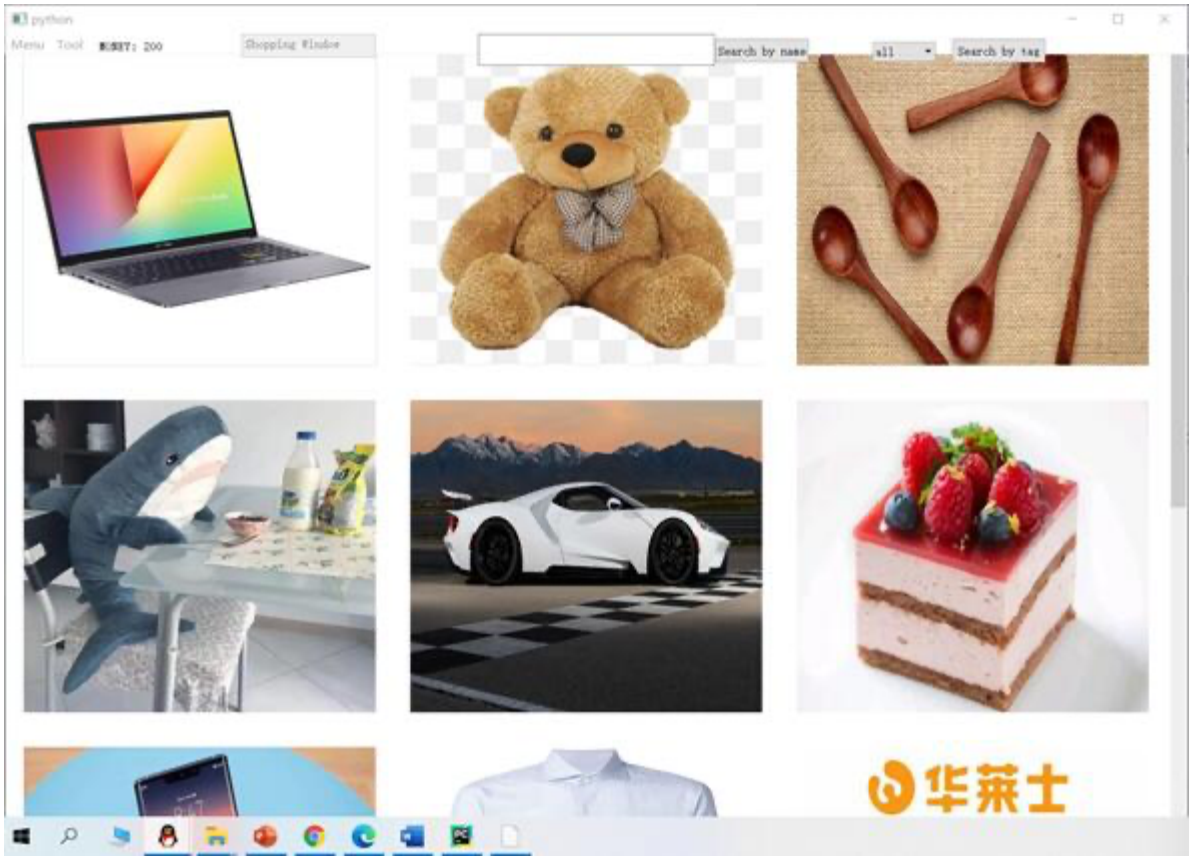


- S1.1.1: Select login mode
  1. Select login if you already have an account or register if you haven't.
  2. Press 'OK' button
     2.1 Username, email, password and verify password are all non-empty string. Email is an unique variable.
     2.2 A class User used to access database which need to take parameter email and password in login mode.
     2.3 After login successfully, global variable user which type is class User will store the basic information including id, name, email.

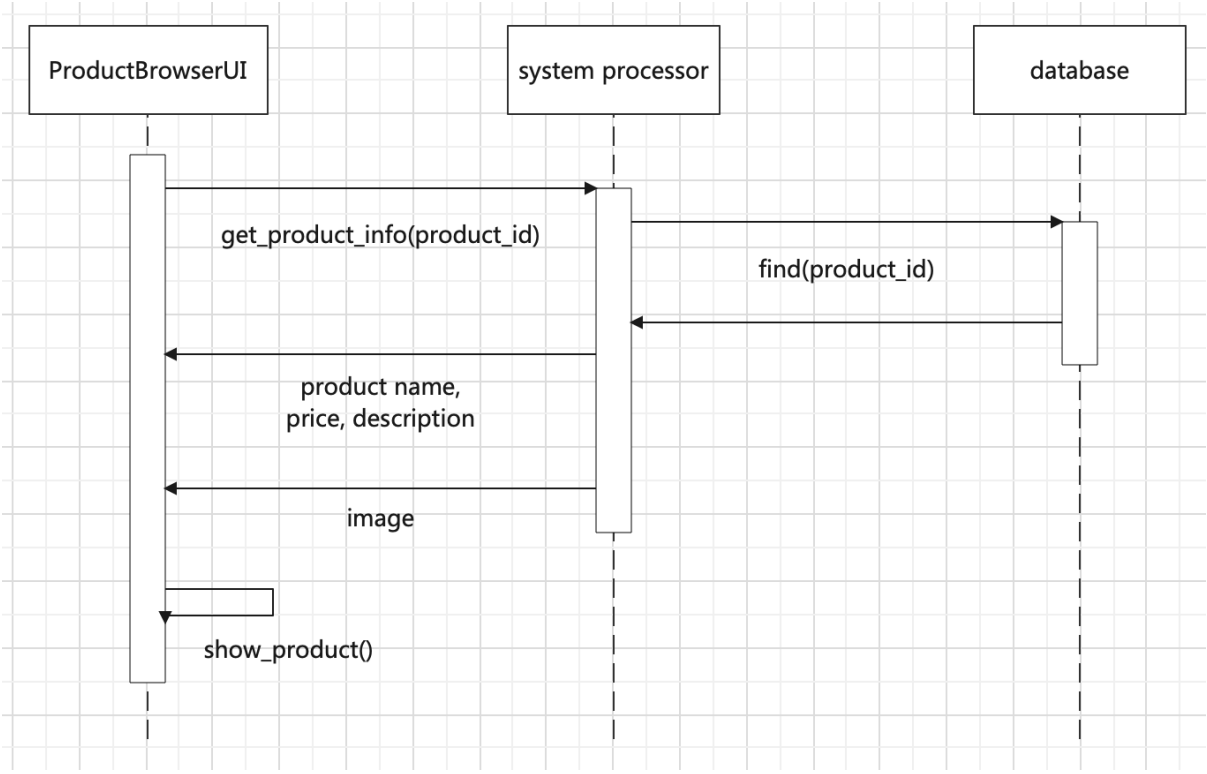
2.4 A list is used to store products user liked. A list is used to store products user watched. A list is used to store products user bought.

- S1.1.2: Select register mode
  3. Select register if you use for the first time.
  4. Press ‘OK’ button
     4.1 Username, email, password and verify password are all non-empty string. Email is an unique variable.
     4.2 A class User used to access database which need to take parameter email and password and username additionally in register mode.
     4.3 Verify password need to be the same as password.
     4.4 After register successfully, global variable user which type is class User will store the basic information including id, name, email.
     4.5 A list is used to store products user liked. A list is used to store products user watched. A list is used to store products user bought.

S2: Product browser

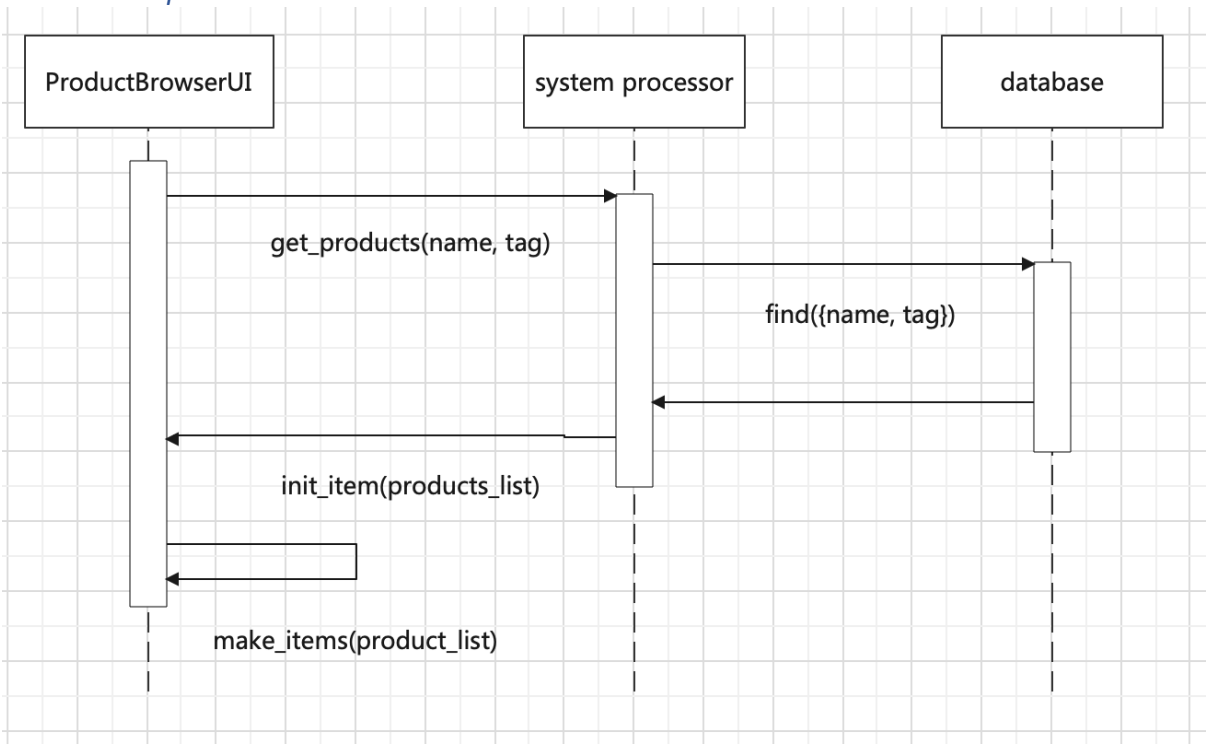


S2.1: Select a product



- S2.1.1: Select a product
  1. Get product id which was chosen.
  2. Find product information by its id from database.
  3. Find product image path by its id from database.
  4. Generate a new window to show product image, name, description, price, remaining amount, tag.

S2.2: Search product

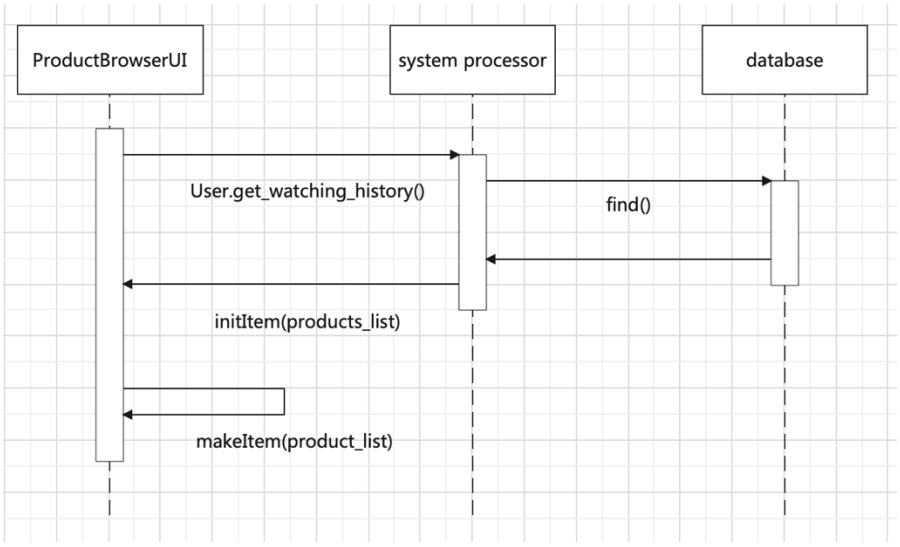


- S2.2.1: Press ‘search by name’ button

1. Get text in the box and send to system processor.
2. Find products which name matches the given text and return a list.
3. Refresh the showing products with the list returned.

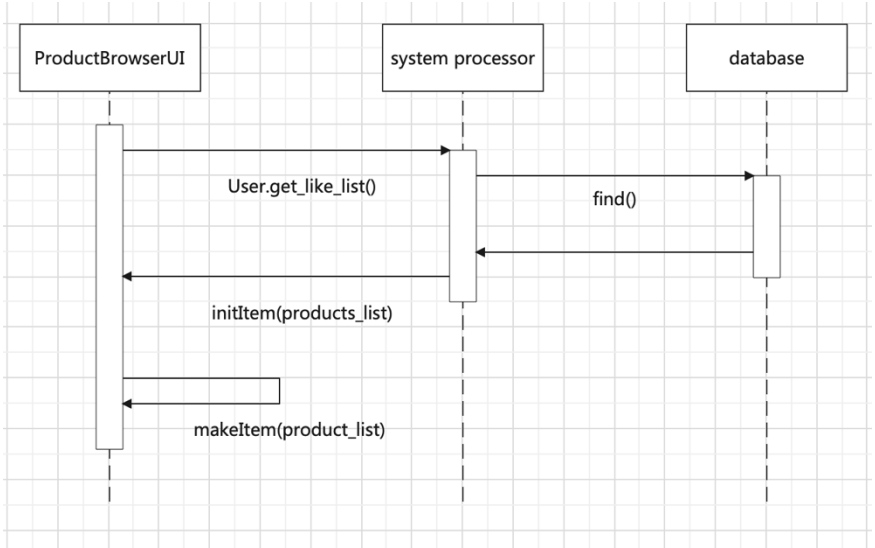
- S2.2.2: Press ‘search by tag’ button
  1. Get tag that selected and send to system processor.
  2. Find products which tag matched the given tag and return a list.
  3. Refresh the showing products with the list returned.

S2.3: Check watching history



- 2.3.1 Watching history
  1. A list is used to save product id every time user visit then save in a csv file after user logout.
  2. Load csv file by user id then return the products list.
  3. Refresh the showing products with the list returned.

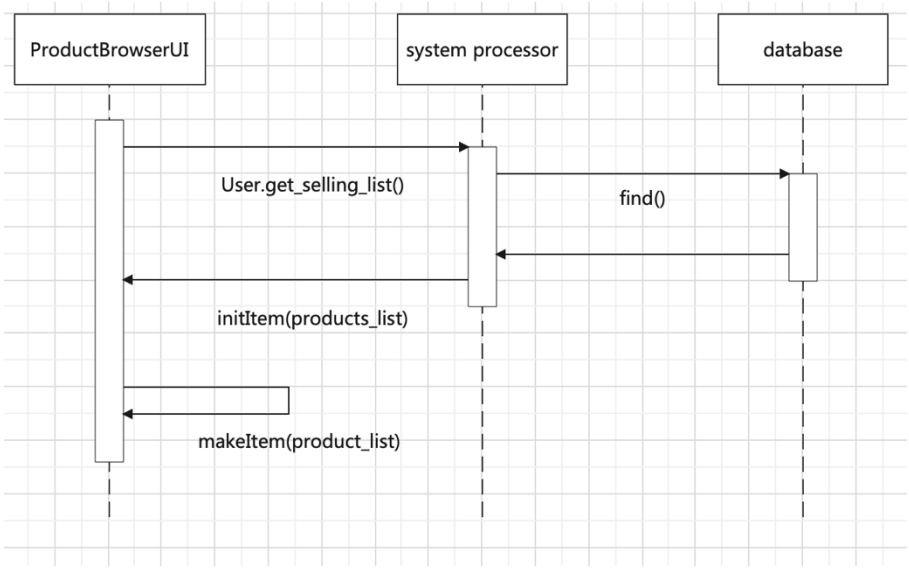
S2.4: Check like list



- 2.4.1 Watching history

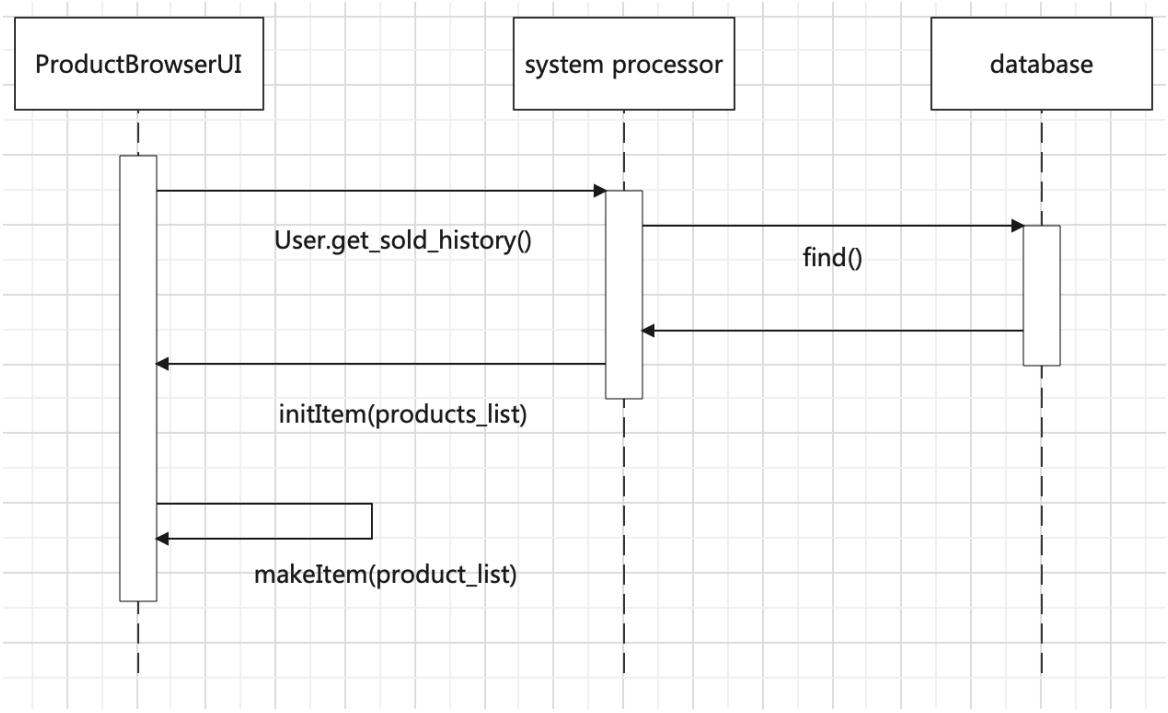
1. A list is used to save product id every time user click ‘like’ button in the product page then save in a csv file after user logout.
2. Load csv file by user id then return the products list.
3. Refresh the showing products with the list returned.

S2.5: Check selling list



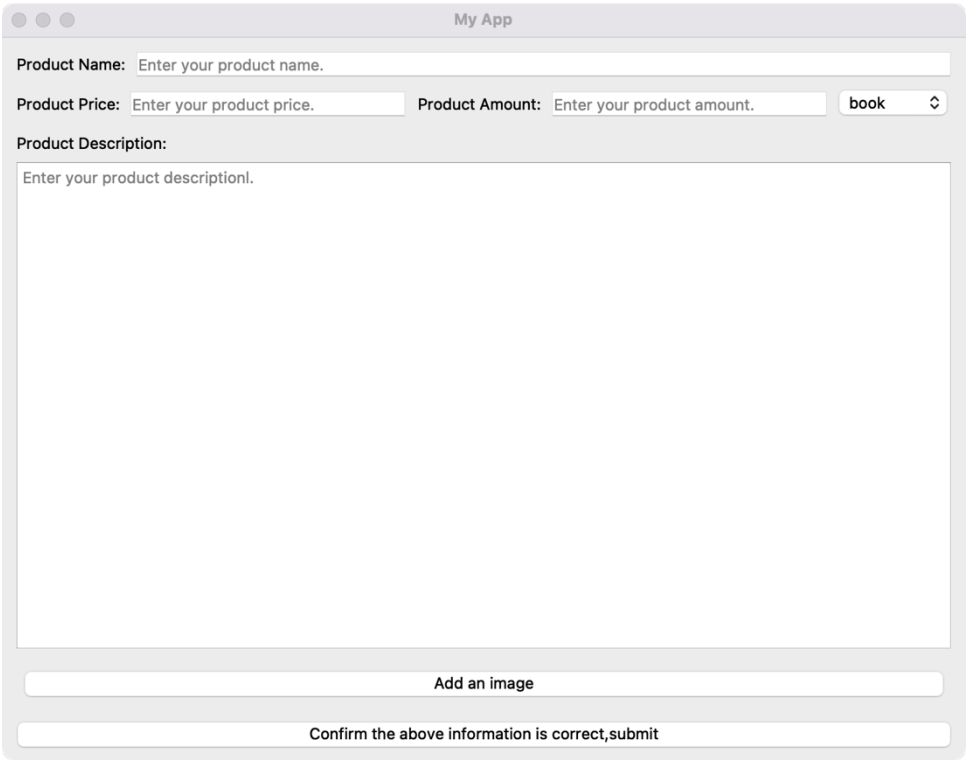
- 2.5.1 Watching history
  1. A list is used to save product id every time user uploaded an item then save in a csv file after user logout.
  2. Load csv file by user id then return the products list.
  3. Refresh the showing products with the list returned.

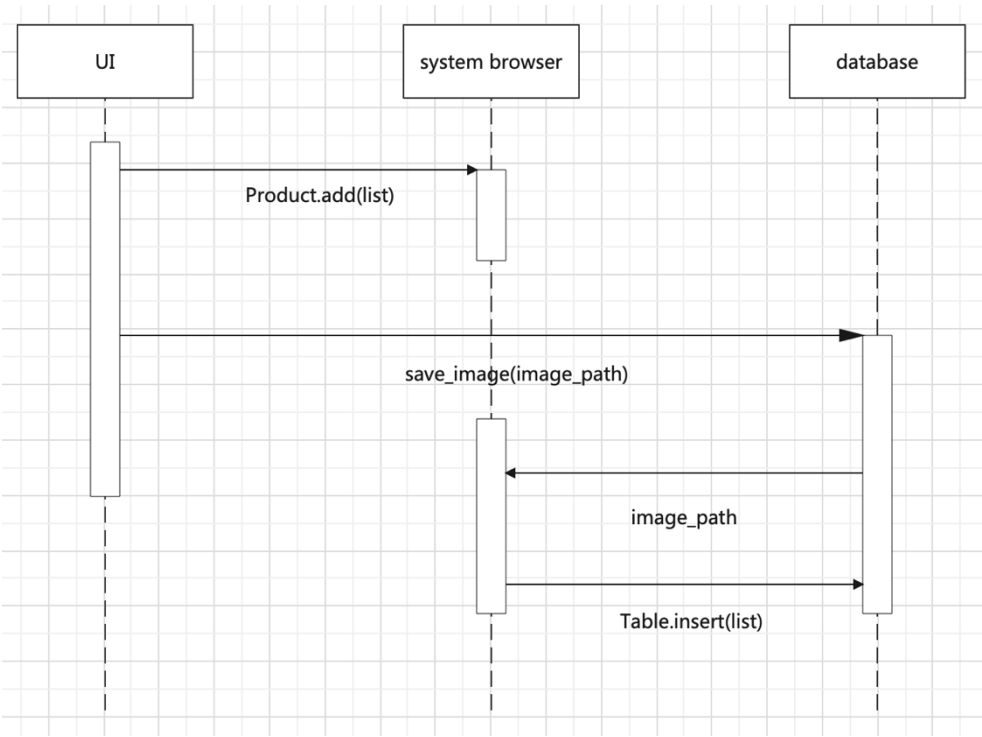
S2.6: Check sold history



- 2.6.1 Watching history
  1. A list is used to save product id every time user sold an item then save in a csv file after user logout.
  2. Load csv file by user id then return the products list.
  3. Refresh the showing products with the list returned.

S3: Upload product

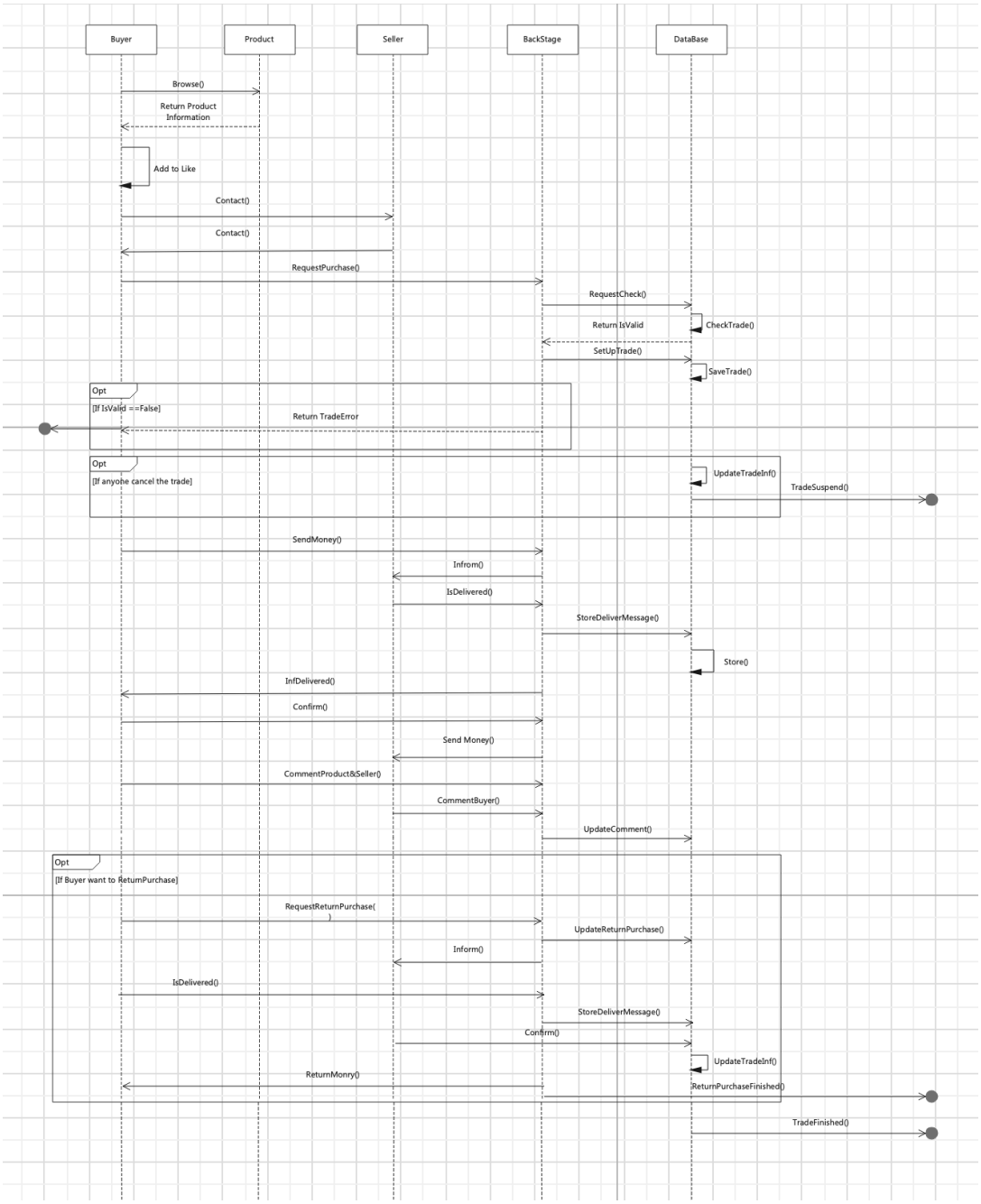




S3.1: Upload product information

- S3.1.1: Upload product information
  1. Product name, description are non-empty strings. Tag is a string but must choose tags from the given set i.e. ['book', 'toy', 'phone', 'food', 'PC', 'tool', 'shirt', 'cooker']. Price and amount are signed int variables, this will check after user press ‘submit’ button.
  2. After user choose an image, copy the image to folder ‘./Code/pictures’, return the new picture path.
  3. After user press ‘submit’ button, a list is used to save all product information, then database part will save it into a csv file.

S4: Trade



S4.1: Buy product

- S4.1.1: Buy product
  1. A trade is created when buyer press 'buy' button in the product page and his/her money is enough. A trade is a list contains buyer id, seller id, product id, trade date, trade state.
  2. Trade status:
     0: buyer: buy, buyer--money-->admin
     1: seller: deliver
     1.5: product on the way, seller--product-->buyer
     2: buyer: receive, admin--money-->seller
     3: buyer: return purchase with no reasons(i.e. in 7 days), bank--money->buyer
     4: buyer: require for return purchase
     5: seller: accept the requirement, bank--money->buyer
     6: seller: reject the requirement
     6.5: product on the way, buyer--product-->seller
     7: seller: receive the product, seller--money->bank
     8: trade done!

S4.2: Return product

- S4.2.1: Return product
  1. Check weather trade within 7 days.
  2. Trade within 7 days will directly change its state to 3, after send back money, state will change to 7 and it will change to 8 after seller receive the product.
  3. Trade out of 7 days will be more complicated. The state will be directly change to 4 when buyer want to return the product. If seller reject the require, state will change to 6 then to 8. If seller accept the require, state will change to 5 and then 7 after send back money. Also state will change to 8 after seller receive the return product.