Software Requirements Specification For <BoYing>

1854081 FU CHENG

1853781 DAI JIAHUI

1851486 FANG HAO

Course: Software Engineering

Instructor: HUANG JIE

Teaching Assistant: PAN CHAO

Contents

1 Introduction

Software requirements specification (SRS) refers to the specification written by software engineers or analysts after completing feasibility analysis and investment benefit analysis on the basis of studying user requirements. It defines in detail the information flow and interface, functional requirements, design requirements and restrictions, testing criteria, quality assurance requirements and so on. It is the SRS's function to serve as a technical agreement reached by users and software developers, as the basis and basis for starting design work, after the completion of the system development, it provides a basis for product acceptance.

1.1 Purpose

The purpose of this document is to present a detailed description of the designs of "BoYing". The SRS shows how the software system will be structured to satisfy the requirements identified in the software requirements specification. It is a translation of requirements into a description of the software structure, software components, interfaces and data necessary for the implementation phase. This document serves as a guide to implementing the project. At the same time, this document can also be used by designers trying to upgrade or modify the current design of the system.

1.2 Scope

The software system being produced is called "BoYing", which is used as an online-platform for users to buy movie tickets, drama tickets and so on. The system will run on a browser, avoiding the limitations of the operating system platform, and each user can interact with it through a web browser. Individual users can create accounts themselves.

The system will allow users to browse, search, select shows and purchase tickets. After purchasing tickets, the system will reduce the corresponding inventory. Users can view personal historical orders and refund tickets at the same time. Besides, users can modify personal information, such as mobile phone numbers, to receive verification codes.

BoYing will also allow managers to add shows, modify, and delete shows information, and will automatically do the process of getting the statistics daily to help managers better understanding the sales.

However, this system still has some limitations. For example, it does not have a complete payment subsystem. All the users will use a third-part service to do the paying. And the system will only use electronic tickets instead of physical tickets after the payment.

1.3 Definitions, acronyms, and abbreviations

SRS	Software Requirements Specification
BoYing	The system that will be produced, which is called "博影" in Chinese.
User	A person that is a customer of the system.
Manager	A single person that has the ability to create, update and delete show messages in the system, and view the statistics about the selling at the same time. This person cannot simultaneously act as a User and Manager.
Show	Performance activities held by performing units or individuals at a specific time and under a specific environment
Order	Documents that proves the trade.
Statistics	The collecting and analyzing of numerical data.
Ticket	A slip of paper or card that need to be admitted to a movie theater, a museum, which proves that the users have paid.
Checkout	The process a Customer goes through to purchase the tickets.
Refund	Give an amount of money back to the users, especially because the users are not satisfatic with a product or service that have bought.
CRUD	Create, Retrieve, Update, Delete.
Button	A user interface element that allows a User to submit the query to the system.

1.4 Organization

This Software Requirements Specification document is divided in to multiple subsections.

The first section includes explanations of the Purpose, Scope and Organization of the document. The first section also handles the description of project-specific words, acronyms and abbreviations that will be used in the document.

The second section of the document is separated into the following five different sections, each detailing specific details of system uses and their corresponding actions: Product Perspective, Product Functions, User Characteristics, Constraints, Assumptions and Dependencies, Apportioning of Requirements.

The third section is an enumerated listing of all of the requirements described for this system and encompasses all of the Use-case, Sequence, State and Class diagrams that model the system.

The fourth section contains a listing of all related reference materials used in this document.

2 Overall Description

This section includes details about what is and is not expected of the "BoYing" system in addition to which cases are intentionally unsupported and assumptions that will be used in the creation of the "BoYing" system.

2.1 Product Perspective

BoYing is a website which allows users to browse all kinds of shows, such as movies, drama and so on, and buy tickets online at the same time. Besides, it also allows the managers to add, update and delete shows, and get the selling details through the statistics provided by the system daily. Generally, the system serves as an online-ticket platform between users and managers to make the whole process of buying tickets more convenient.

The website must be available to anyone with a person computer-based browser and Internet. There are no hardware or software requirements beyond these including, but limited to, memory or specific software packages that need to be utilized nor software packages that need not be utilized.

2.2 Product Functions

This section will summarize the major functions the product must perform or must let the user perform. More details about functions will be showed in 3.2 Functional requirements.

Maintain data associated with the shows

A show has a name, time, price, address and other details about it.

The system will keep track of the quantity of the tickets.

Maintain records for users

A user should at least have a name and password.

The user could login either with password or with a verification code to the mobile phone.

Anyone may sign up for a user account.

The system will send a verification code to the user's mobile phone while registering.

Users can log out by themselves.

Show a listing of shows

Shows are to be displayed in the homepage of the users by different of types.

The information of the shows, including the name, address and the price will

be displayed briefly in the homepage.

The user could get the detailed information of the shows by clicking it to get into the show-detail page.

The show-detail page will include the details about the show, like the price, and the poster.

Buy tickets

Buying tickets is only available to logged-in users. A user that is not logged in is given a chance to log in.

The users can refund tickets before the show starting.

The users can view all the orders they have in the history-order page, including the status of them.

Allow managers to manage the information of shows

Managers can insert shows into the database by add the name, address, price and other details about them.

Managers can update the information of the shows.

Managers can delete the information of shows in the database.

Allow managers to view the orders

Managers are allowed to view all the history orders of all the users.

Allow managers to view the statistics

The system will do the process of computing the statistics of the sale daily, making the managers available to getting the states of the sales, and better do the busyness.

2.3 User Characteristics

The typical user is simply anyone that has access to the Internet and a web browser in the compute. It is assumed that the user is familiar enough with a computer to operate the browser, keyboard and mouse and is capable of browsing to, from and within simple websites. However, the administrator of the system should have the basic knowledge of the sales, with the ability to manage the whole system at the same time.

2.4 Constraints

1. Internet connection with enough bandwidth to fully render various displays.

2. CPU speed or RAM of the device is not a big concern.

3. Specific technologies, tools, and databases to be used: We will use Vue front-end framework Java back-end framework and MySQL database.

4. Hardware limitations (timing requirements, memory requirements): No limitations, our equipment performance is sufficient to support product operation.

5. The system may not behave correctly when used with Internet browsers other than Firefox and Chrome.

6. The database should not store passwords in plain text and doesn't need to be a password recovery feature nor lockout after numerous invalid login attempts.

7. The system should be able to send verification code to the users' mobile-phone to ensure the security.

8. The users' password should be strong, which means that it should contains a sufficient number of characters and contains not only lowercase letters but also capitals, numbers, and in some cases, symbols.

9. The system should use the third-part API to accomplish the payment, like WeChat or Alipay.

10. The system will use e-ticket instead of physical ticket.

2.5 Assumptions and Dependencies

We have assumed that all of the computer systems are in proper working condition and that the user is capable of operating these system's basic functions including but not limited to being able to power on the system, login and open either Chrome or Mozilla Firefox, and navigate the browser to the address of this system website.

We have assumed that the system will be running on a properly working web server and database system with an Internet connection that allows this system to perform all communications with clients.

We have assumed that the third-part payment API will work successfully at any time.

2.6 Apportioning of Requirements

The system is not responsible for the following:

1. Verifying that payment account is valid.

2. Verifying that the identity information of the user is valid.

3. Doing the payment.

Additionally, the system may need to later be extended to provide additional functions. One such example is added support for promotions about the sale, which will make the system more attractive to users.

3 Specific Requirements

3.1 External interface requirements

3.1.1 User interfaces

1. For users

Homepage: Display all the shows, users can search for them on the homepage.

Shows details page: show the shows' specific information, including the name, the price of each kind of ticket, the time to be present and so on, and provide users with options to purchase tickets.

The history order page: Shows all orders generated by the user to purchase the tickets.

Personal homepage: Shows the user's personal information and list, and users can legally modify these contents.

2. For managers

Statistics page: Shows the statistics of the sale daily, including the orders amount, and the sum of the turnover.

Shows all the products that the publisher is selling. The publisher can search for the game and modify the game information.

Shows management page: Shows all the shows' information. The manager can search for them, and add, update, delete the information for management.

Order list page: Shows all orders accepted by the manager.

3.1.2 Hardware interfaces

The software supports computers with Windows, Mac OS and Linux operating systems where a web browser can run on.

3.1.3 Software interfaces

Data Management System: MySQL Database

Operating system: Windows, Mac OS, Linux

Application type: Web-based application

3.1.4 Communications interfaces

We define the communication interface through the local network protocol, and the content format of the interface is explained in detail in the installation manual and user manual.

3.2 Entity Class Data Structure

User has these attributes

1. Unique ID (auto-increment starting at 1)

2. User name

3.Phone number

4.Password

5.Age

6.Gender

Manager has these attributes

1.Unique ID (auto-increment starting at 1)

2.Password

Show has these attributes

1. Unique ID (auto-increment starting at 1)

2.Name

3.Poster

4.Min-Price

5.Max-Price

6.Address

7.Time

8.Detail information

Ticket has these attributes

1. Unique ID (auto-increment starting at 1)

2.OrderId

3.Qr-code-url

4.Price

Order has these attributes

1. Unique ID (auto-increment starting at 1)

2.UserId

3.ShowId

4.Price

5.Ticket Count

6.Status

3.3 System

3.3.1 Browse Shows

Organization:

Shows listed on the homepage.

User can browse detail information in the show-detail page.

Each show has its own show-detail page.

The show-detail page including the following information:

Name

Time

Address

Price

Poster

Detail description

Interaction:

Users click at the show in the homepage and then jump into the show-detail page.

3.3.2 Search Shows

Search available by the name, address, time and price of the shows.

Search is exact-match only.

3.3.3 Create, Update and Destroy (CRUD) Functionality

Only managers are allowed to modify the data.

Managers have an interface to:

Create a show entry

Update a show entry

Destroy a show entry

Managers have full CRUD capabilities on each show entry.

3.3.4 Checkout procedure

User must log in successfully before beginning the procedure.

The checkout procedure will use a third-party API such as Alipay and WeChat.

There is a "checkout" button in the show-detail page and after the users clicking it, the checkout procedure begins.

The system should display the checkout result to the user.

The system will create an order after successful checkout.

3.3.5 Authentication System

Users Levels

Manager (single, hardcoded user, no orders)

Customer (unlimited, open creation, unlimited orders)

Account Creation

Everyone is allowed to create an account.

The mobile phone number is required to receive a verification code.

Account Modification

Users are not able to modify any aspect of their account after creation. ("it would be nice but not needed")

Login and Logout

There is no lost-password recovery.

Logging in allows one to logout.

Logging in allows checkout.

3.3.6 History Orders System

The user can view all his history orders in the orders page.

The orders have their own status.

3.3.7 Refund System

The user can refund tickets after checkout.

The orders that to be refunded must be valid.

The refund procedure will use a third-part API.

3.3.8 Statistics System

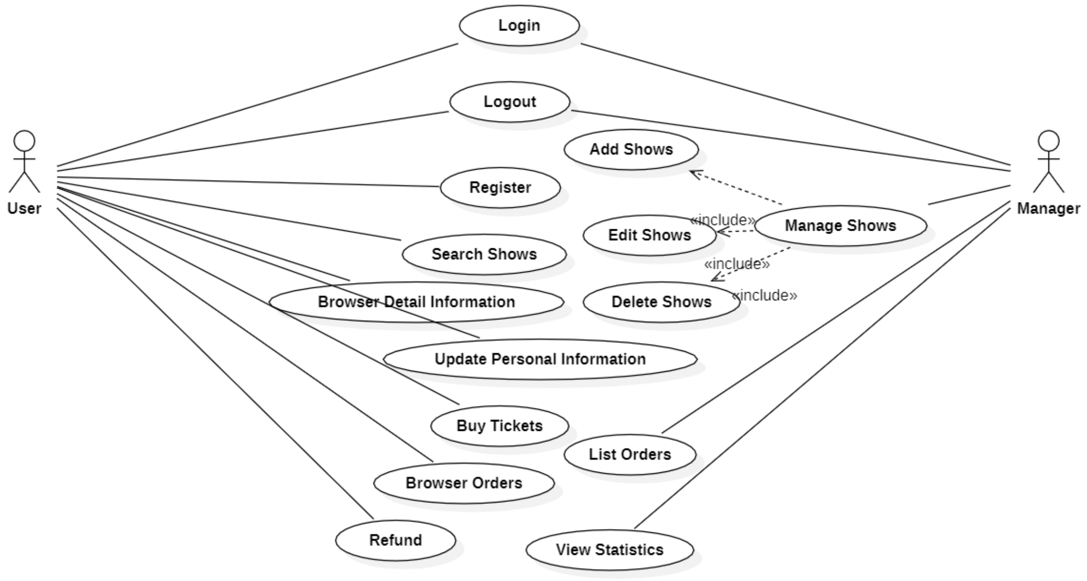
The system will automatically do the procedure of computing the statistics daily.

The statistics include the orders amount, and the sum of the turnover every day.

The manager can view the statistics in the statistics page.

3.4 Functional requirements

3.4.1 Uses Cases



Register

Actors: Users

Type: Primary and essential

Description: Anyone can get an account with a mobile phone number.

Includes: None

Extends: None

Use-Cases: None

Login

Actors: Managers, Users

Type: Primary and essential

Description: Initiated when a user attempts an action that is restricted. The user is then prompted to enter in their username and password in order to proceed.

Includes: None

Extends: None

Use-Cases: None

Logout

Actors: Managers, Users

Type: Primary and essential

Description: The managers and the users can log out of the system

Includes: None

Extends: None

Use-Cases: User must have completed the Log In use case.

Search shows

Actors: Users

Type: Primary and essential

Description: The users can search shows with the name, address and so on.

Includes: None

Extends: None

Use-Cases: None

Browser detail information

Actors: Users

Type: Secondly

Description: The users can browser the details of the shows.

Includes: None

Extends: None

Use-Cases: None

Update personal information

Actors: Users

Type: Primary and essential

Description: The users can update personal information.

Includes: None

Extends: None

Use-Cases: User must have completed the Log In use case.

Buy tickets

Actors: Users

Type: Primary and essential

Description: The users can check out.

Includes: None

Extends: None

Use-Cases: User must have completed the Log In use case.

Browser orders

Actors: Users

Type: Primary and essential

Description: The users can view their own history orders.

Includes: None

Extends: None

Use-Cases: User must have completed the Log In use case.

Refund

Actors: Users

Type: Primary and essential

Description: The users can refund after checking out.

Includes: None

Extends: None

Use-Cases: User must have completed the Log In use case and had some orders.

Manage shows

Actors: Managers

Type: Primary and essential

Description: The managers can manage the shows.

Includes: Add shows. Edit shows. Delete shows.

Extends: None

Use-Cases: Managers must have completed the Log In use case.

Add shows

Actors: Managers

Type: Primary and essential

Description: The managers can add shows.

Includes: None

Extends: None

Use-Cases: Managers must have completed the Log In use case.

Edit shows

Actors: Managers

Type: Primary and essential

Description: The managers can update the information of shows.

Includes: None

Extends: None

Use-Cases: Managers must have completed the Log In use case.

Delete shows

Actors: Managers

Type: Primary and essential

Description: The managers can delete shows.

Includes: None

Extends: None

Use-Cases: Managers must have completed the Log In use case.

List orders

Actors: Managers

Type: Primary and essential

Description: The managers can view all the orders in the system.

Includes: None

Extends: None

Use-Cases: Managers must have completed the Log In use case.

View statistics

Actors: Managers

Type: Primary and essential

Description: The managers can view the statistics served by the system every day.

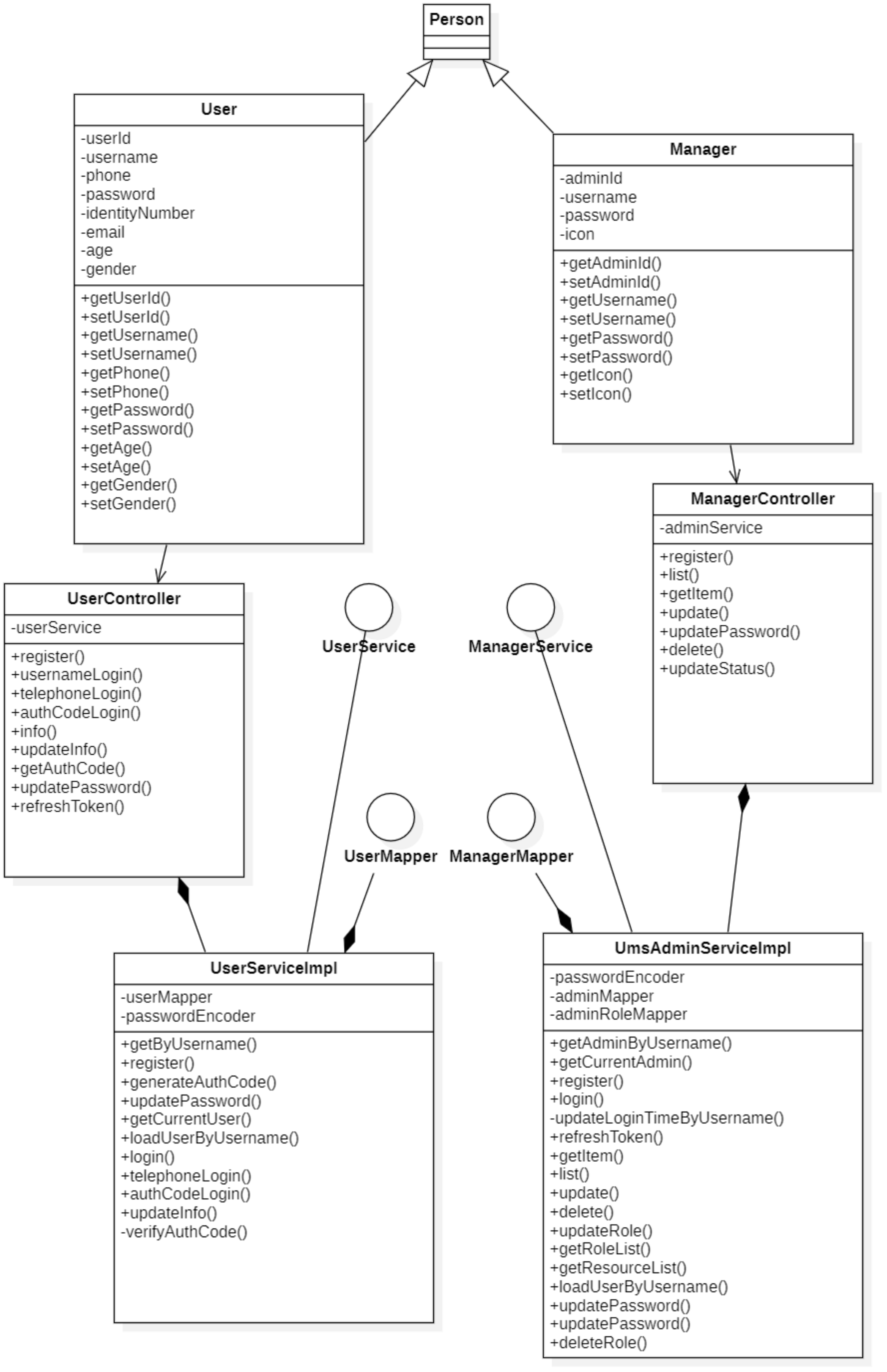
Includes: None

Extends: None

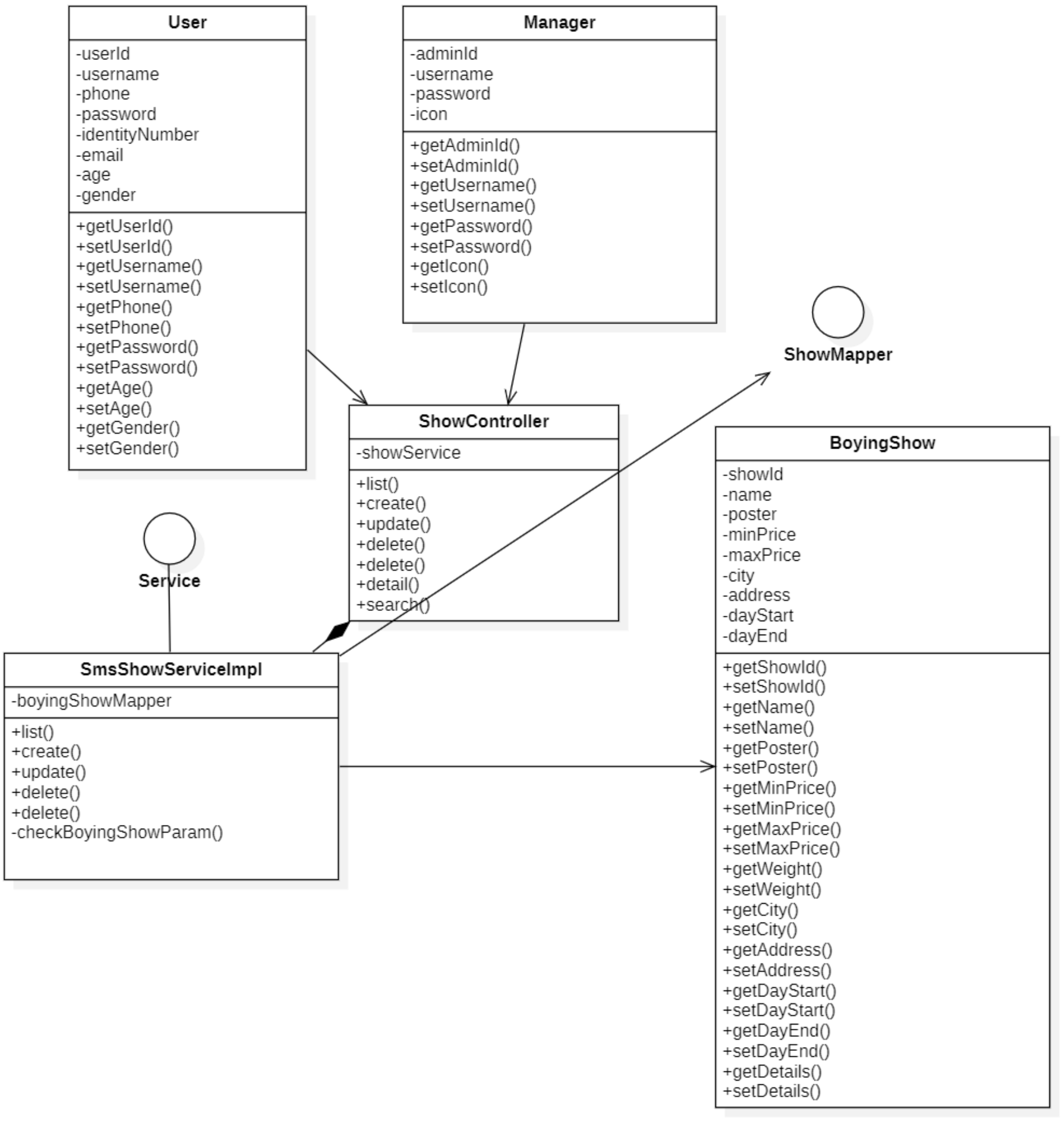
Use-Cases: Managers must have completed the Log In use case.

3.4.2 Class Diagram

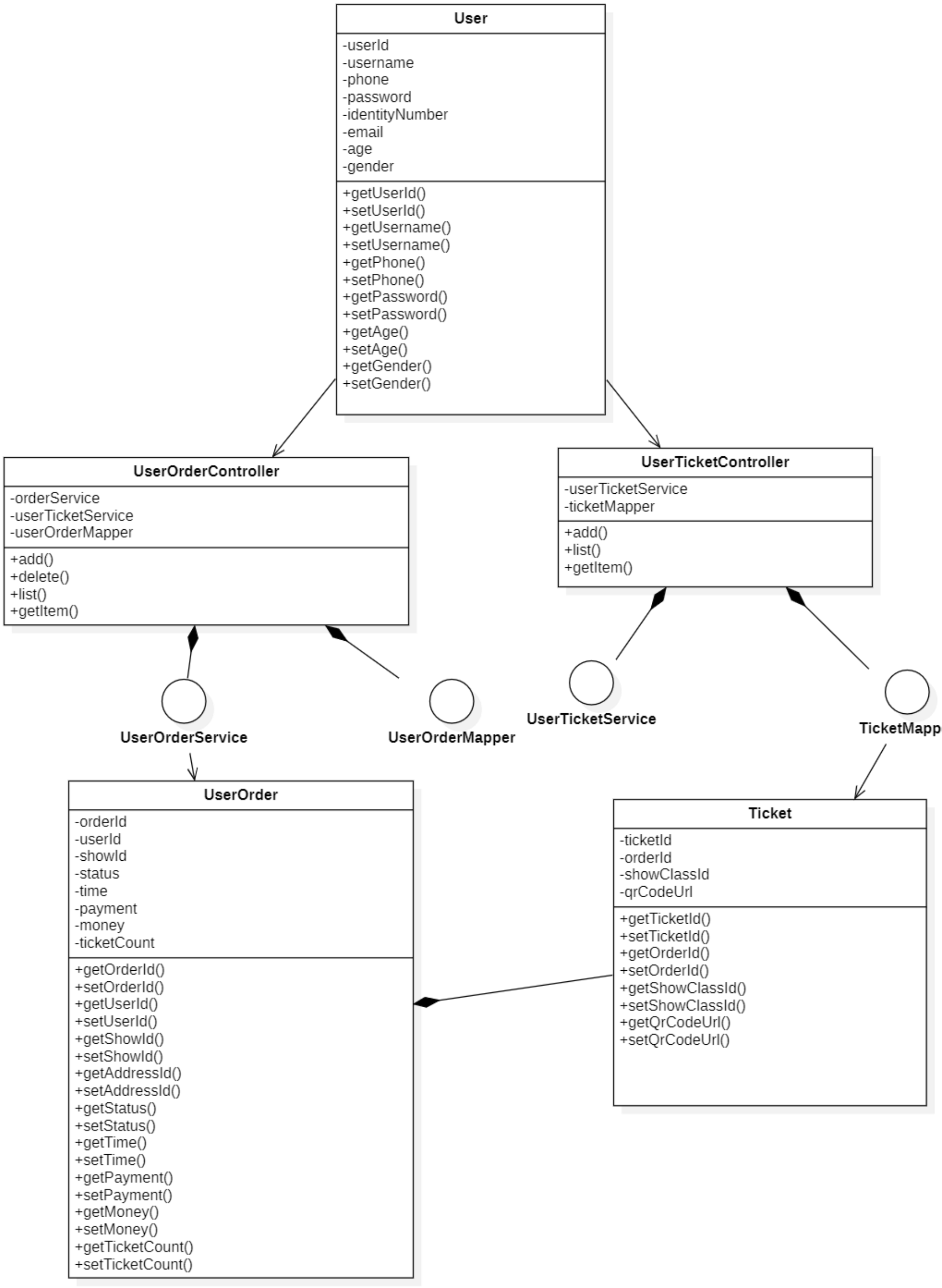
Account Module


Person
User
-userId
-username
-phone
-password
-identityNumber
-email
-age
-gender
+getUserId()
+setUserId()
+getUsername()
+setUsername()
+getPhone()
+setPhone()
+getPassword()
+setPassword()
+getAge()
+setAge()
+getGender()
+setGender()
Manager
-adminId
-username
-password
-icon
+getAdminId()
+setAdminId()
+getUsername()
+setUsername()
+getPassword()
+setPassword()
+getIcon()
+setIcon()
ManagerController
-adminService
+register()
+list()
+getItem()
+update()
+updatePassword()
+delete()
+updateStatus()
UserController
-userService
+register()
+usernameLogin()
+telephoneLogin()
+authCodeLogin()
+info()
+updateInfo()
+getAuthCode()
+updatePassword()
+refreshToken()
UserService
ManagerService
UserMapper
ManagerMapper
UserServiceImpl
-userMapper
-passwordEncoder
+getByUsername()
+register()
+generateAuthCode()
+updatePassword()
+getCurrentUser()
+loadUserByUsername()
+login()
+telephoneLogin()
+authCodeLogin()
+updateInfo()
-verifyAuthCode()
UmsAdminServiceImpl
-passwordEncoder
-adminMapper
-adminRoleMapper
+getAdminByUsername()
+getCurrentAdmin()
+register()
+login()
-updateLoginTimeByUsername()
+refreshToken()
+getItem()
+list()
+update()
+delete()
+updateRole()
+getRoleList()
+getResourceList()
+loadUserByUsername()
+updatePassword()
+updatePassword()
+deleteRole()

Shows Module



Order Module



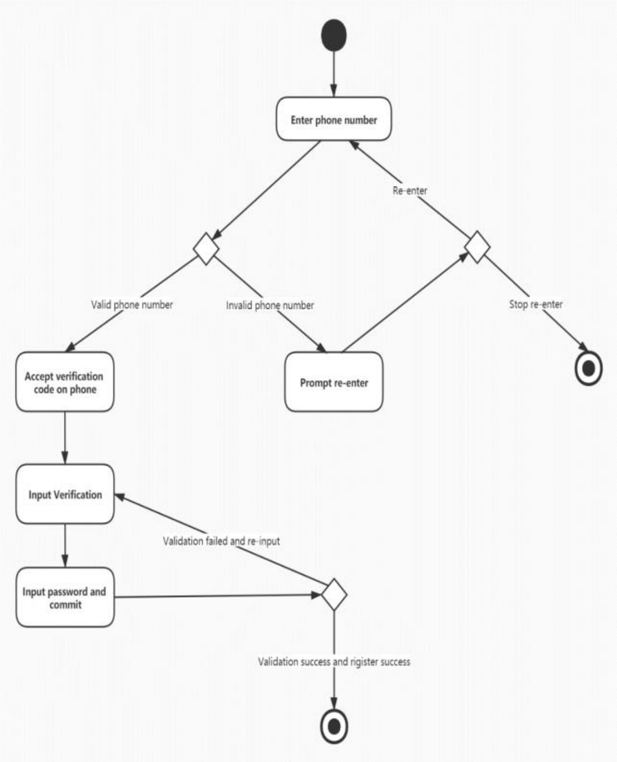
3.4.3 User Mode

3.4.3.1 Users register

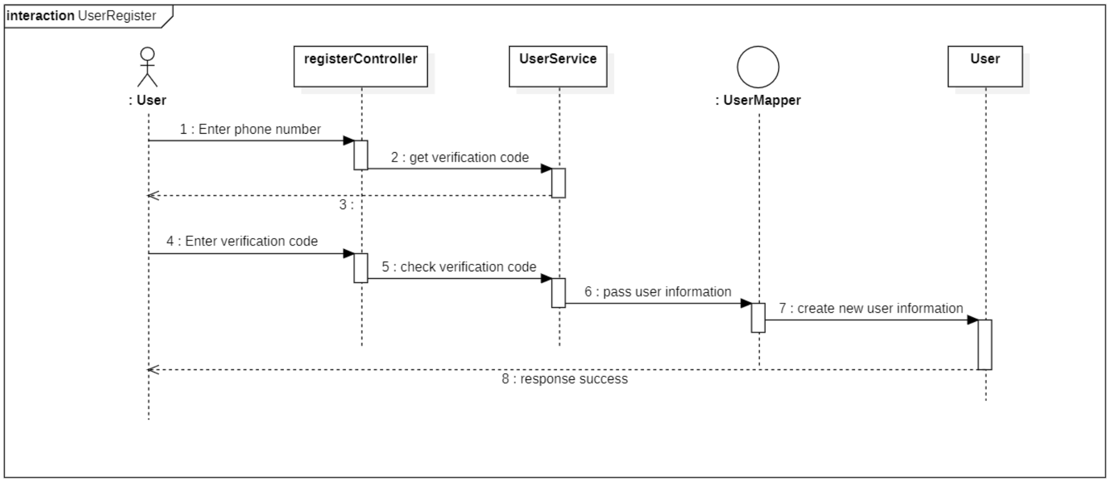
Use case	The user register to the system.
Main participant	General user.
Situational goals	Users can register to the system with a phone number.
Prerequisite	Users must have a valid phone number.
Trigger	Users decide to register at some point.
Scenes	1.The user opens the BoYing register interface. 2.The user input the phone number. 3.The system sends a verification code. 4.The user get the code and input it in the system register page. 5.The user input the password. 6.The user registers successfully.
Exception handling	1.The verification code the user inputs are incorrect--The system displays the appropriate error message and prompts the user to re-enter it.
Priority	User register is the most basic function of the system and has the highest priority.
When it is valid	The first increment.
Usage frequency	Higher frequency.
Connection channels for participants	Connect to the BoYing system website through a personal computer-based browser and the Internet.
Secondary participant	1.System administrator
Connection channels for secondary participants	1.System administrator: A personal computer-based system.

Unsolved problem	1.Is it safe enough? Hacking the function will invade the system and user's private information and cause property threats. 2. How many concurrent visits are there?

Activity diagram


Enter phone number
Re-enter
Valid phone number
Invalid phone number
Stop re-enter
Accept verification code on phone
Prompt re-enter
Input Verification
Validation failed and re-input
Input password and commit
Validation success and rigister success

Sequence diagram

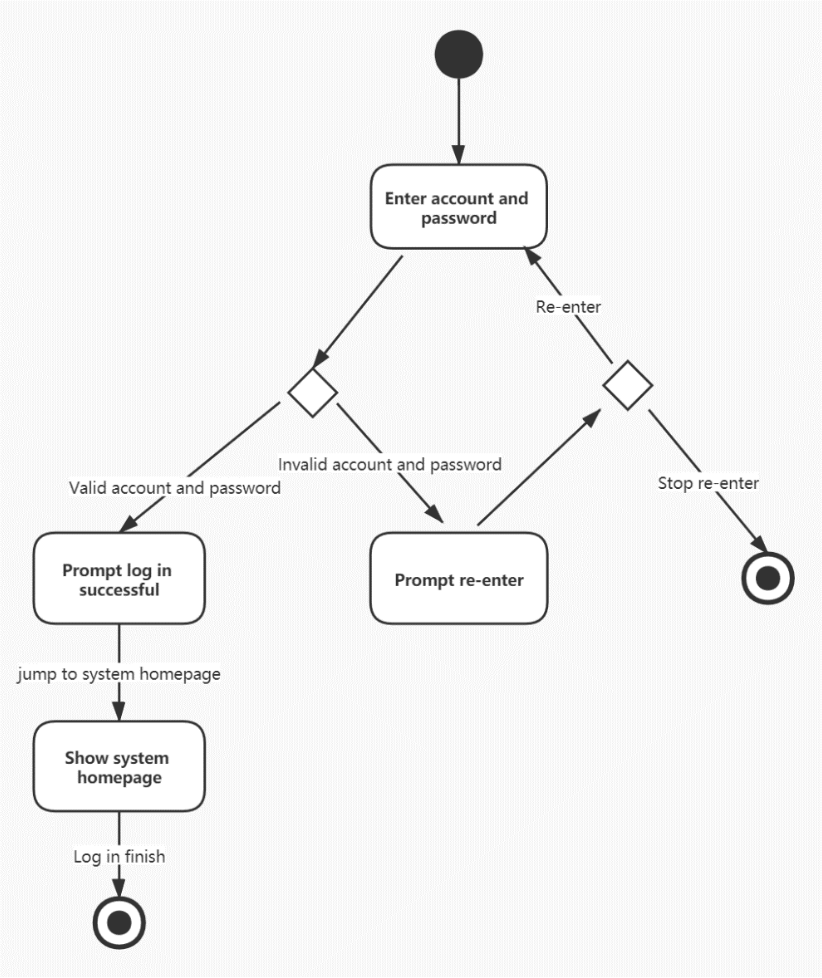

interaction UserRegister
: User
registerController
UserService
: UserMapper
User
1 : Enter phone number
2 : get verification code
3 :
4 : Enter verification code
5 : check verification code
6 : pass user information
7 : create new user information
8 : response success

3.4.3.2 User log in

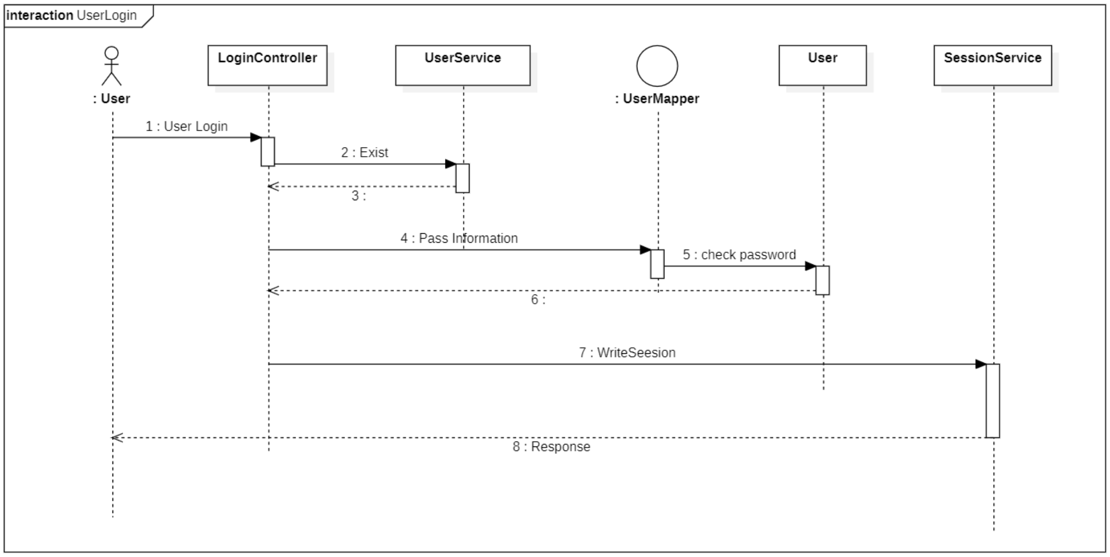
Use case	The user logs in to the system.
Main participant	General user.
Situational goals	Users can log in to the system with a phone number.
Prerequisite	Users must have a valid account and password.
Trigger	Users decide to log in to the system at some point.
Scenes	1.The user opens the system login interface. 2.The user enters an account. 3.The user enters the password. 4.The user select "Login" to submit a login application. 5.The system detects that the account password entered by the user is valid. 6.The system prompts the user login successfully. 7.The user jumps to system homepage.
Exception handling	1.The user account or password is incorrect or not confirmed—The system displays the appropriate error message and prompts the user to re-enter it. 2.System failure—The system displays appropriate error messages.
Priority	User login is the most basic function of the system and has the highest priority.
When it is valid	The first increment.
Usage frequency	Higher frequency.
Connection channels for participants	Connect to the BoYing system website through a personal computer-based browser and the Internet.

Secondary participant	1.System administrator.
Connection channels for secondary participants	1.System administrator: A personal computer-based system.
Unsolved problem	1.Is it safe enough? Hacking the function will invade the system and user’s private information and cause property threats. 2.How often does this system log in? How many concurrent visits are there? 3.Does this system need other login methods such as e-mail login?

Activity diagram



Sequence diagram

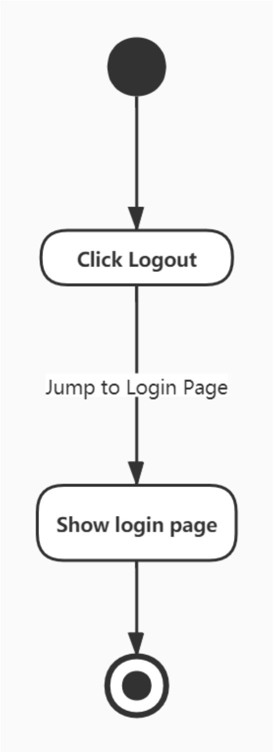


3.4.3.3 User log out

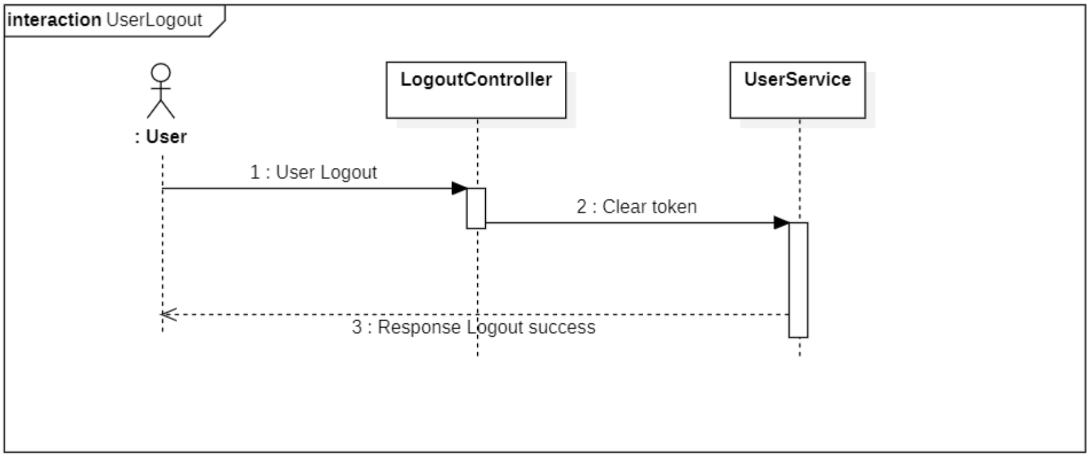
Use case	The user logs out of the system.
Main participant	General user.
Situational goals	Users can log out of the system.
Prerequisite	None.
Trigger	Users decide to log out of the system at some point.
Scenes	1.The user opens the system homepage. 2.The user click the “Logout” button. 3.The system goes to the login page.
Exception handling	None
Priority	User login is the most basic function of the system and has the highest priority.
When it is valid	The first increment.

Usage frequency	Higher frequency.
Connection channels for participants	Connect to the BoYing system website through a personal computer-based browser and the Internet.
Secondary participant	1.System administrator.
Connection channels for secondary participants	1.System administrator: A personal computer-based system.
Unsolved problem	1.Should the system clear all the cache in the browser?

Activity diagram



Sequence diagram

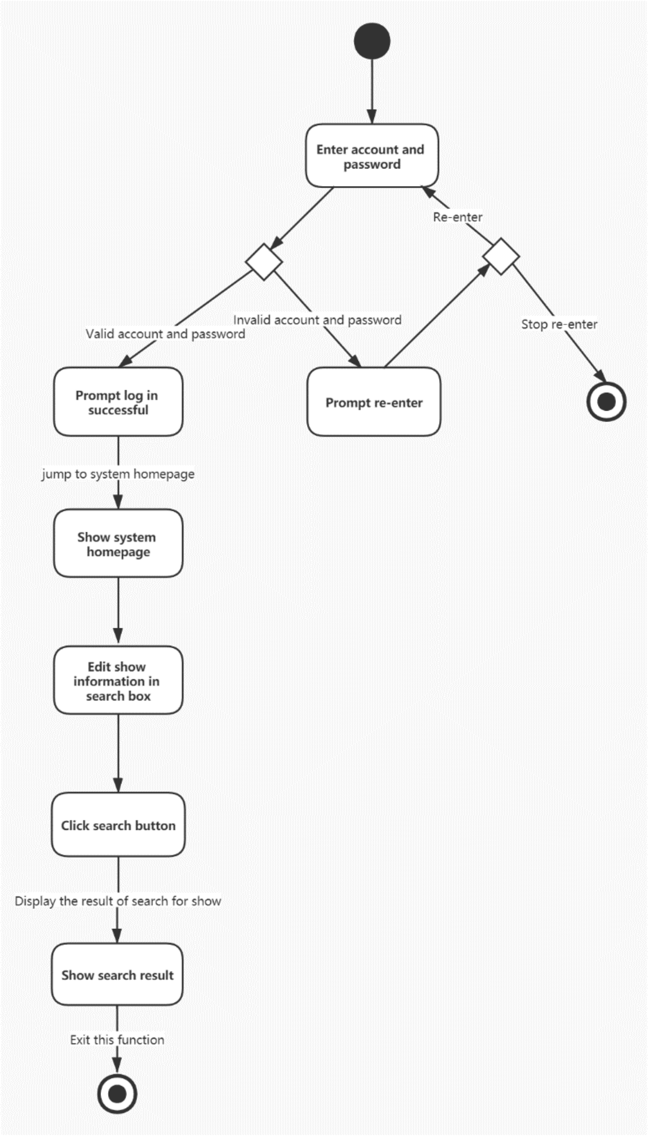


3.4.3.4 The user search for shows

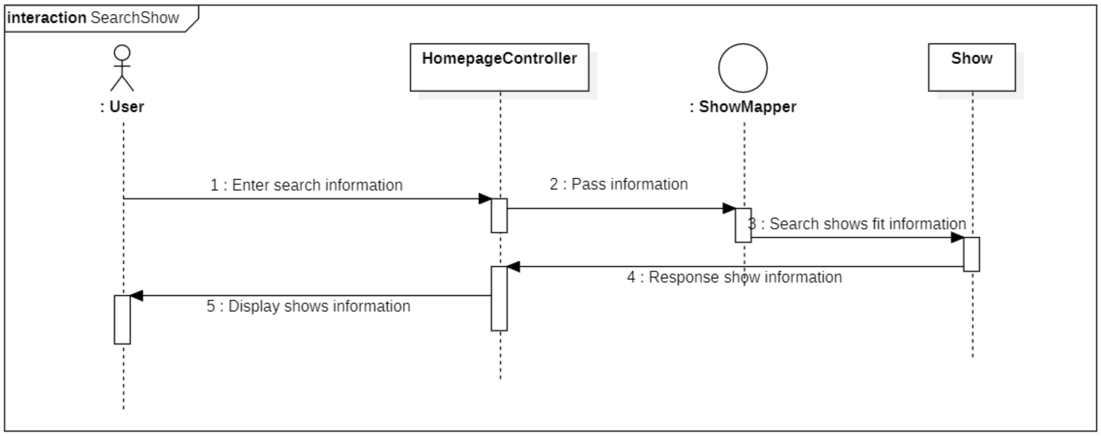
Use case	The user searches for shows in the system.
Main participant	General users.
Situational goals	Users can search for shows keywords on the page and find all shows related.
Prerequisite	Users must log in to the system.
Trigger	Users decide to browser the detail information.
Scenes	1.The user opens the login interface. 2.The user enters an account. 3.The user enters the password. 4.The user selects “Login” to submit a login application. 5.The system detects that the account password entered by the user is valid. 6.The system prompts the user login successfully. 7.The user jumps to the homepage. 8.The user enters what he is looking for in the search box. 9.The user selects the “Search” button.

	10.The system searches for related shows and displays them based on the content entered by the user.
Exception handling	The user account or password is incorrect or not confirmed—The system displays the appropriate error message and prompts the user to re-enter it. The system failure—The system displays appropriate error messages. The content entered by the user is illegal—The system displays an appropriate error message and prompts the user to enter it again.
Priority	User search is a relatively common function of the system and has medium priority.
When it is valid	The third increment.
Usage frequency	Medium frequency.
Connection channels for participants	Connect to the system website through a personal computer-based browser and the Internet.
Secondary participant	1.System administrator.
Connection channels for secondary participants	1.System administrator: A personal computer-based system.
Unsolved problem	1.When the searching result is in large account, should the system display only part of it? 2.Should the system order the results?

Activity diagram



Sequence diagram

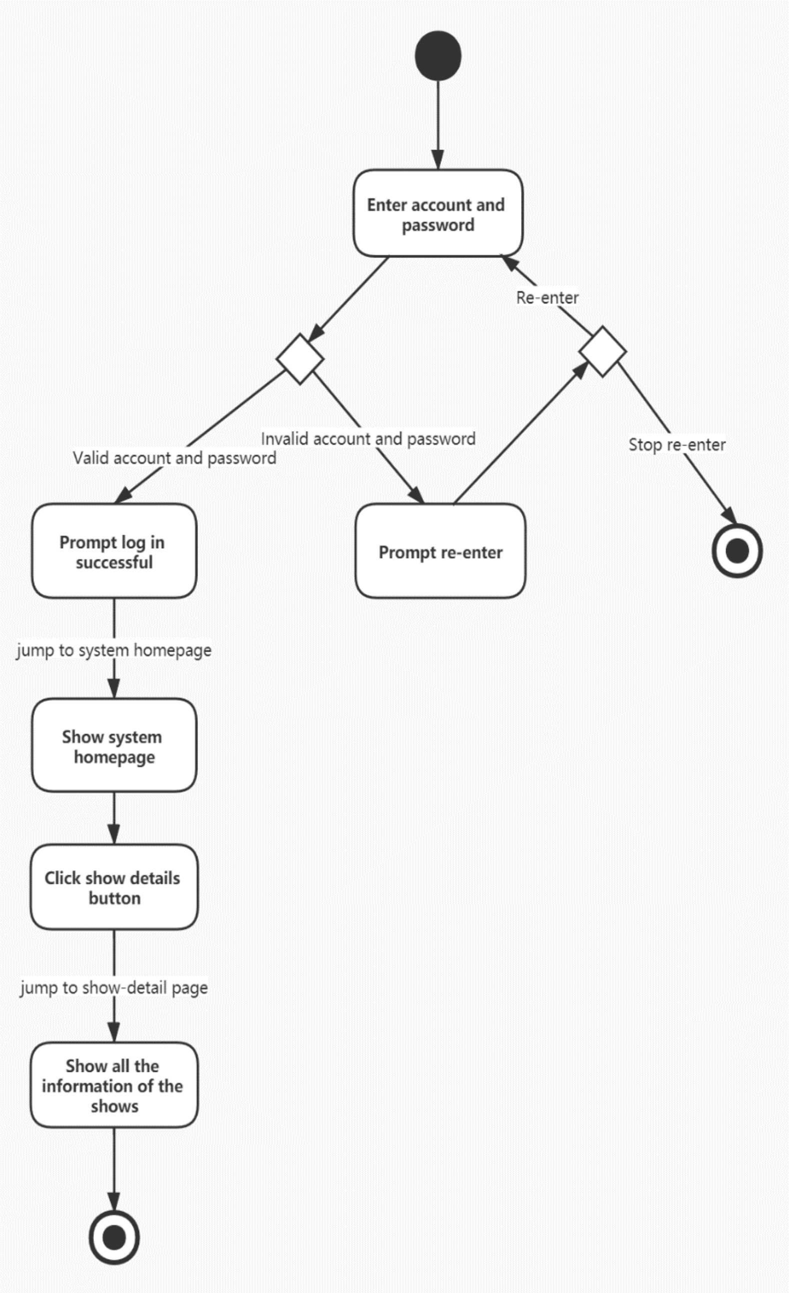

interaction SearchShow
: User
HomepageController
: ShowMapper
Show
1 : Enter search information
2 : Pass information
3 : Search shows fit information
4 : Response show information
5 : Display shows information

3.4.3.5 The user views the shows' details

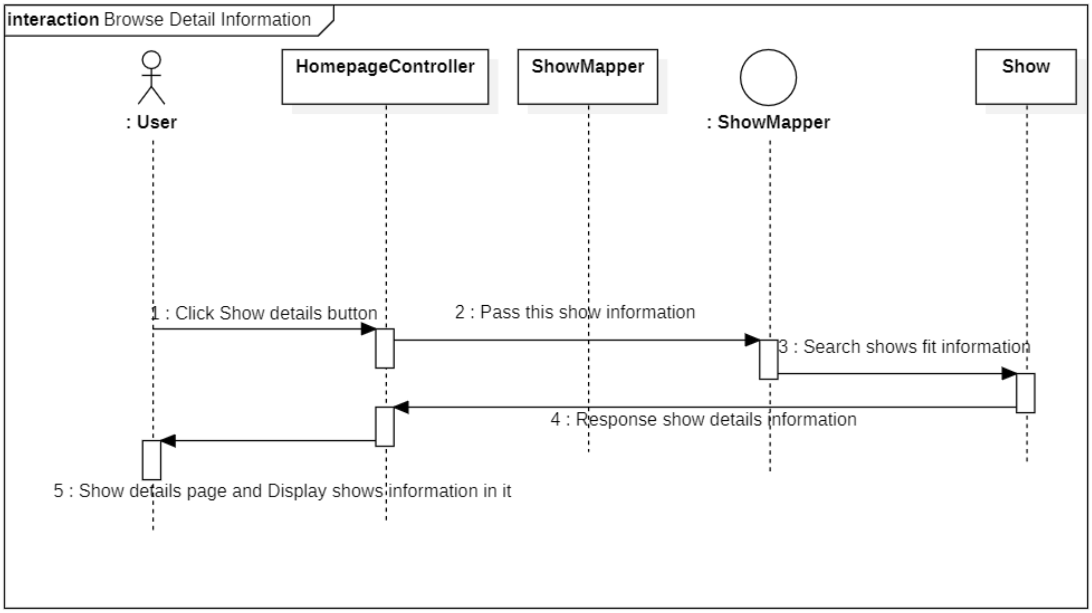
Use case	The user views the detail information of the shows in the system.
Main participant	General users.
Situational goals	Users can search for shows keywords on the page and find all shows related.
Prerequisite	Users must log in to the system.
Trigger	Users decide to search for shows.
Scenes	1.The user jump into the home page after login. 2.The user click the left button above the brief information of the shows. 3.The system jump into the show-detail page. 4.The system display all the information of the shows. 5.The user browsers all the information and decided whether to buy tickets.
Exception handling	The system failure—The system displays appropriate error messages.
Priority	Browser detail information is the most basic function of the system and has the highest priority.
When it is valid	The first increment.
Usage frequency	Most frequency.
Connection channels for participants	Connect to the system website through a personal computer-based browser and the Internet.
Secondary participant	1.System administrator.
Connection channels for secondary	1.System administrator: A personal computer-based

participants	system.
Unsolved problem	1.Should the system only display part of the detail information to avoid the big data flow between the frontend and the backend?

Activity diagram


Enter account and password
Re-enter
Invalid account and password
Valid account and password
Stop re-enter
Prompt log in successful
Prompt re-enter
jump to system homepage
Show system homepage
Click show details button
jump to show-detail page
Show all the information of the shows

Sequence diagram

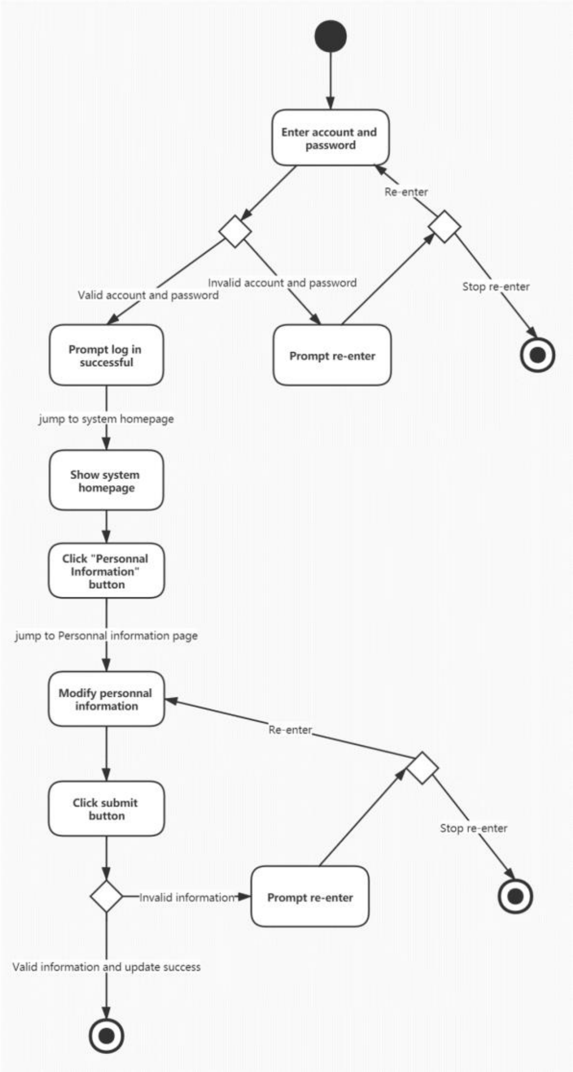

interaction Browse Detail Information
: User
HomepageController
ShowMapper
: ShowMapper
Show
1 : Click Show details button
2 : Pass this show information
3 : Search shows fit information
4 : Response show details information
5 : Show details page and Display shows information in it

3.4.3.6 The user updates personal information

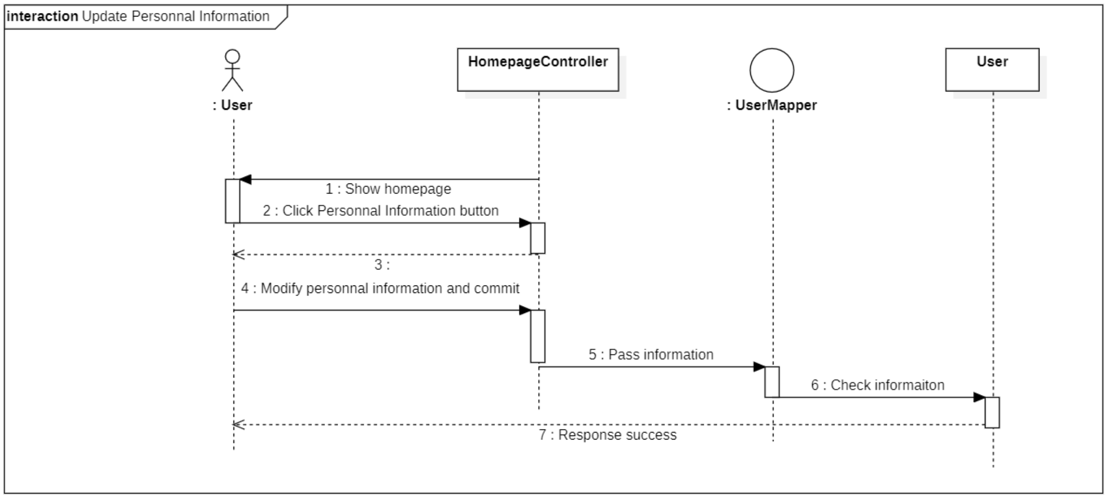
Use case	The user updates personal information in the system.
Main participant	General users.
Situational goals	Users can update personal information.
Prerequisite	Users must log in to the system.
Trigger	Users decide to update person information
Scenes	1.The user jump into the home page after login. 2.The user click the button of "Person Information". 3.The system jump into the personal-information page. 4.The user update the information, including the mobile phone number, the password. 5.The user click the submit button. 6.The system send a verification code to the mobile phone. 7.The user enters the code.

	8.The system update the information in the database. 9.The system display the successful message.
Exception handling	The system failure—The system displays appropriate error messages. The verification code the user entered is invalid—The system displays appropriate message and the user enters again.
Priority	Update personal information of the users is the most basic function of the system and has the highest priority.
When it is valid	The first increment.
Usage frequency	Most frequency.
Connection channels for participants	Connect to the system website through a personal computer-based browser and the Internet.
Secondary participant	1.System administrator.
Connection channels for secondary participants	1.System administrator: A personal computer-based system.
Unsolved problem	1.Should the system check if the password after updating is strong enough? 2.Should the system send the message to the past mobile phone to make identification?

Activity diagram


Enter account and password
Re-enter
Valid account and password
Invalid account and password
Stop re-enter
Prompt log in successful
Prompt re-enter
jump to system homepage
Show system homepage
Click "Personnal Information" button
jump to Personnal information page
Modify personnal information
Re-enter
Click submit button
Stop re-enter
Invalid information
Prompt re-enter
Valid information and update success

Sequence diagram

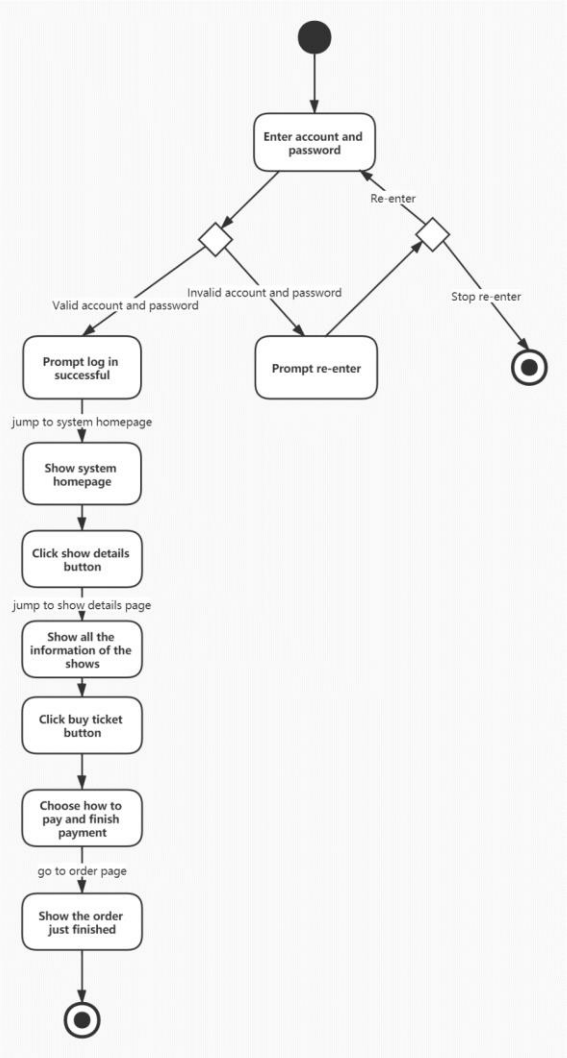

interaction Update Personnal Information
: User
HomepageController
: UserMapper
User
1 : Show homepage
2 : Click Personnal Information button
3 :
4 : Modify personnal information and commit
5 : Pass information
6 : Check informaiton
7 : Response success

3.4.3.7 The user buys tickets in the system.

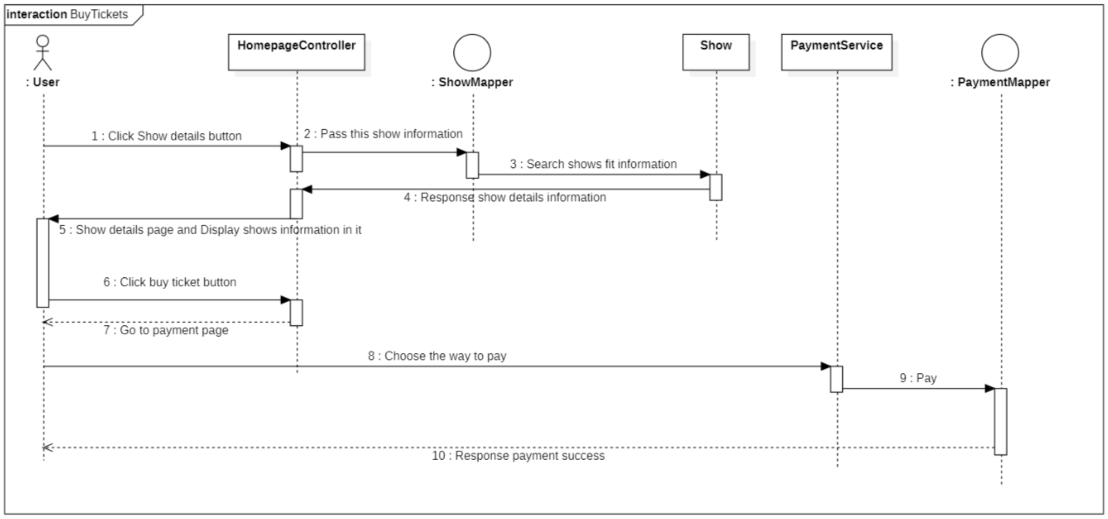
Use case	The user buys tickets in the system.
Main participant	General user.
Situational goals	Users can buy tickets in to the system.
Prerequisite	Users must have a valid account and password.
Trigger	Users decide to buy tickets in the system at some point.
Scenes	1.The user jumps into the show-detail page. 2.The user click the “Buy” button. 3.The user choose the count of tickets. 4.The user select “Submit” to submit the application. 5.The system do the sales and decrease the count of tickets in the database. 6.The system informs the user successfully. 7.The user jumps back to the show-detail page.
Exception handling	1.System failure—The system displays appropriate error messages. 2.The third-part API of payment failure—The system displays appropriate error messages.
Priority	User payment is the most basic function of the system and has the highest priority.
When it is valid	The first increment.
Usage frequency	Higher frequency.
Connection channels for participants	Connect to the BoYing system website through a personal computer-based browser and the Internet.

Secondary participant	1.System administrator.
Connection channels for secondary participants	1.System administrator: A personal computer-based system.
Unsolved problem	1.Is it safe enough while do the payment? 2.What if the system does a large account of payment in a very short time? Will the system fail? 3.Could the speed of paying be increased to improve the experience of users?

Activity diagram


Enter account and password
Re-enter
Invalid account and password
Valid account and password
Stop re-enter
Prompt log in successful
Prompt re-enter
jump to system homepage
Show system homepage
Click show details button
jump to show details page
Show all the information of the shows
Click buy ticket button
Choose how to pay and finish payment
go to order page
Show the order just finished

Sequence diagram

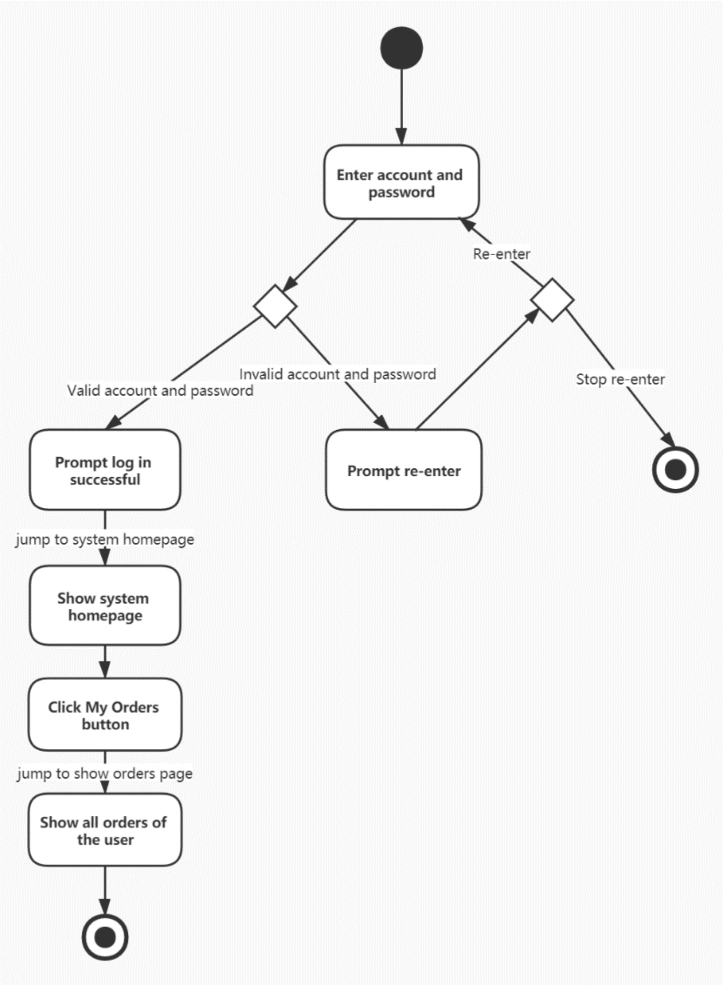

interaction BuyTickets
: User
HomepageController
: ShowMapper
Show
PaymentService
: PaymentMapper
1 : Click Show details button
2 : Pass this show information
3 : Search shows fit information
4 : Response show details information
5 : Show details page and Display shows information in it
6 : Click buy ticket button
7 : Go to payment page
8 : Choose the way to pay
9 : Pay
10 : Response payment success

3.4.3.8 The user browser history orders

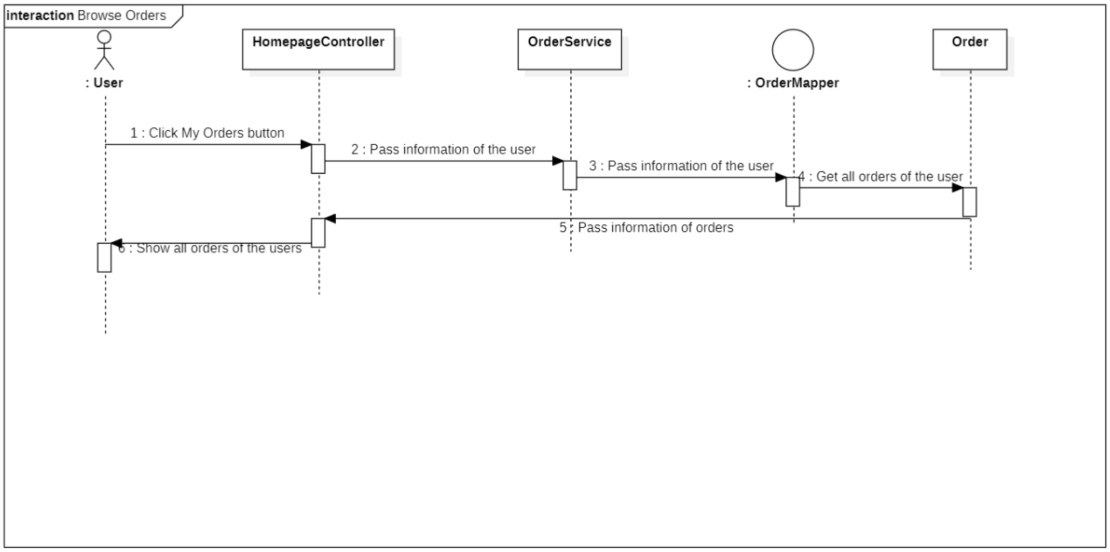
Use case	The user browser history orders.
Main participant	General user.
Situational goals	Users can browser history orders in to the system.
Prerequisite	Users must have a valid account and password. Users must have some orders.
Trigger	Users decide to browser history orders in the system at some point.
Scenes	1.The user jumps into the homepage after login. 2.The user click the “History Orders” button. 3.The system jumps into the history-order page. 4.The system lists all the orders the user has.
Exception handling	1.System failure—The system displays appropriate error messages. 2.The user has no orders—The system displays that there

	is no order.
Priority	User payment is relatively common function of the system and has medium priority.
When it is valid	The second increment.
Usage frequency	Common frequency.
Connection channels for participants	Connect to the BoYing system website through a personal computer-based browser and the Internet.
Secondary participant	1.System administrator.
Connection channels for secondary participants	1.System administrator: A personal computer-based system.
Unsolved problem	1.Should the system display the status of the orders? 2.If there are too many orders, should the system display only a part of them?

Activity diagram



Sequence diagram

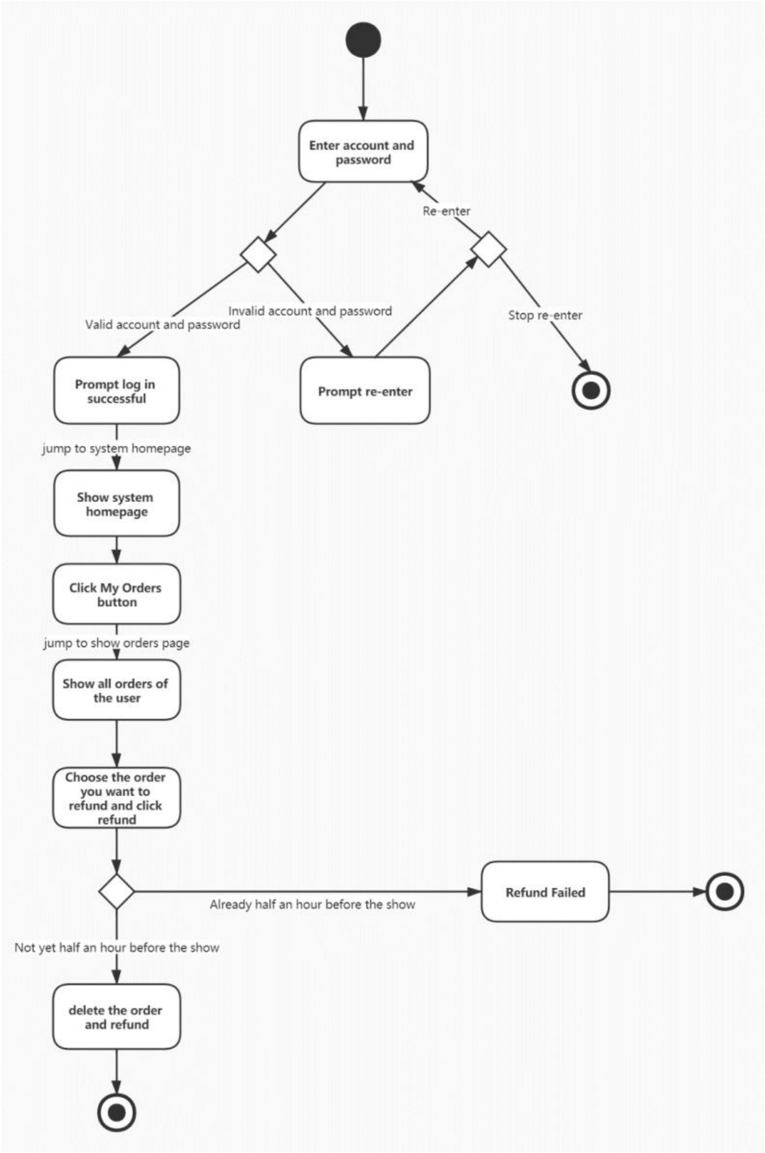

interaction Browse Orders
: User
HomepageController
OrderService
: OrderMapper
Order
1 : Click My Orders button
2 : Pass information of the user
3 : Pass information of the user
4 : Get all orders of the user
5 : Pass information of orders
6 : Show all orders of the users

3.4.3.9 The user refunds tickets after payment.

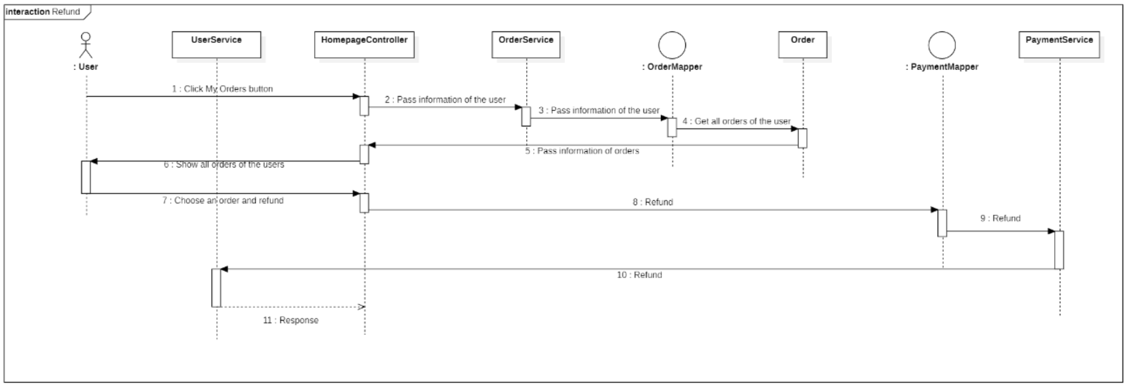
Use case	The user refunds tickets after payment.
Main participant	General user.
Situational goals	Users can refund tickets after payment.
Prerequisite	Users must have a valid account and password. Users must have some orders.
Trigger	Users decide to refunds tickets after payment.
Scenes	1.The user jumps into the homepage after login. 2.The user click the “History Orders” button. 3.The system jumps into the history-order page. 4.The system lists all the orders the user has. 5.The user clicks at the ticket that he wants to refund. 6.The system displays the detail information of the order. 7.The user clicks the “Refund” button. 8.The system does the refunding and updates the database. 9.The system display the success information to the user.
Exception handling	1.System failure—The system displays appropriate error messages. 2.The user has no orders—The system displays that there is no order. 3.The third-part payment API failure—The system displays appropriate error messages.
Priority	User refund is most basic function of the system and has highest priority.
When it is valid	The first increment.

Usage frequency	Common frequency.
Connection channels for participants	Connect to the BoYing system website through a personal computer-based browser and the Internet.
Secondary participant	1.System administrator.
Connection channels for secondary participants	1.System administrator: A personal computer-based system.
Unsolved problem	1.Should the system display the status of the orders? 2.If there are too many orders, should the system display only a part of them? 3.Should the system be improved to speed up the refund?

Activity diagram


Enter account and password
Re-enter
Valid account and password
Invalid account and password
Stop re-enter
Prompt log in successful
Prompt re-enter
jump to system homepage
Show system homepage
Click My Orders button
jump to show orders page
Show all orders of the user
Choose the order you want to refund and click refund
Already half an hour before the show
Refund Failed
Not yet half an hour before the show
delete the order and refund

Sequence diagram


interaction Refund
: User
UserService
HomepageController
OrderService
: OrderMapper
Order
: PaymentMapper
PaymentService
1 : Click My Orders button
2 : Pass information of the user
3 : Pass information of the user
4 : Get all orders of the user
5 : Pass information of orders
6 : Show all orders of the users
7 : Choose an order and refund
8 : Refund
9 : Refund
10 : Refund
11 : Response

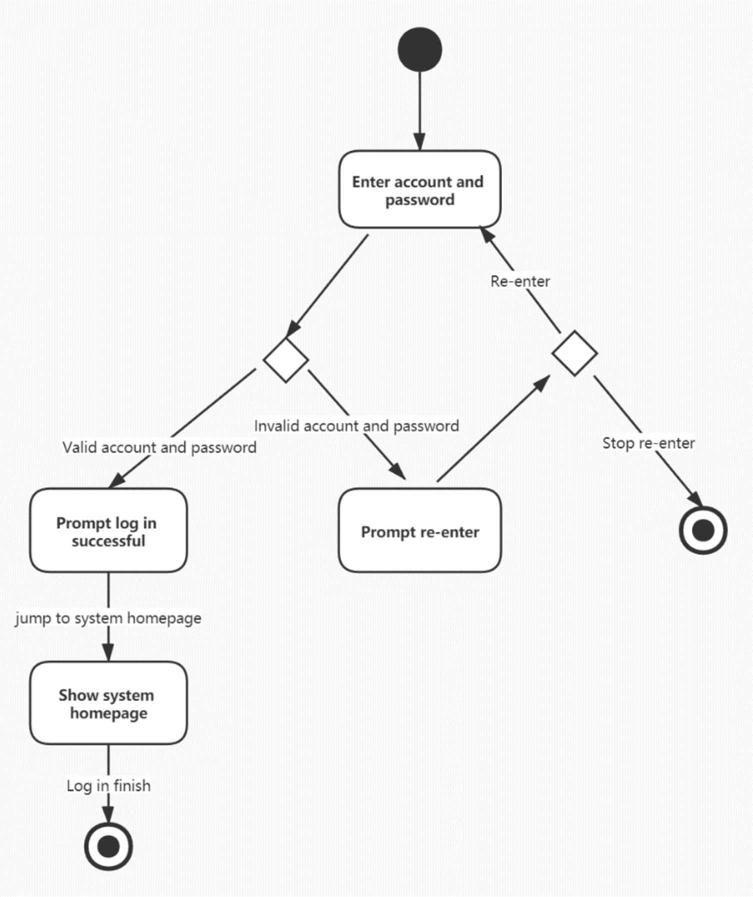
3.4.4 Manager Mode

3.4.4.1 The manager logs in to the system.

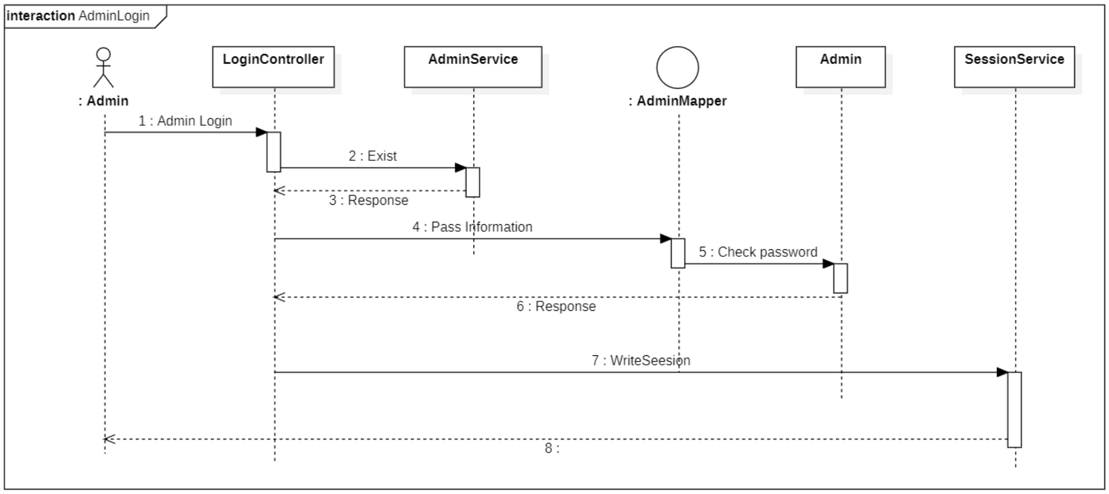
Use case	The manager logs in to the system.
Main participant	Manager.
Situational goals	Managers can log in to the system with a phone number.
Prerequisite	Managers must have a valid account and password.
Trigger	Managers decide to log in to the system at some point.
Scenes	1.The managers opens the system login interface. 2.The managers enters an account. 3.The managers enters the password. 4.The managers select “Login” to submit a login application. 5.The system detects that the account password entered by the user is valid. 6.The system prompts the managers login successfully. 7.The managers jump to system homepage.
Exception handling	1.The managers’ account or password is incorrect or not confirmed—The system displays the appropriate error message and prompts the user to re-enter it. 2.System failure—The system displays appropriate error messages.
Priority	Managers login is the most basic function of the system and has the highest priority.
When it is valid	The first increment.
Usage frequency	Higher frequency.
Connection channels	Connect to the BoYing system website through a personal

for participants	computer-based browser and the Internet.
Secondary participant	None
Connection channels for secondary participants	None
Unsolved problem	1.Is it safe enough? Hacking the function will invade the system and user's private information and cause property threats. 2.How often does this system log in? How many concurrent visits are there? 3.Does this system need other login methods such as e-mail login?

Activity diagram



Sequence diagram

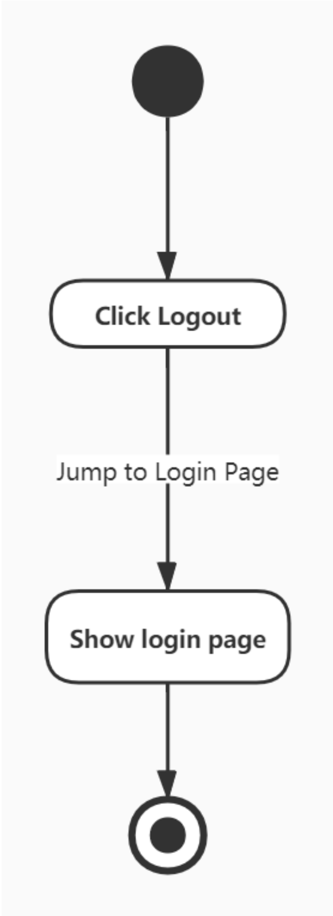


3.4.4.2 The manager logs out of the system.

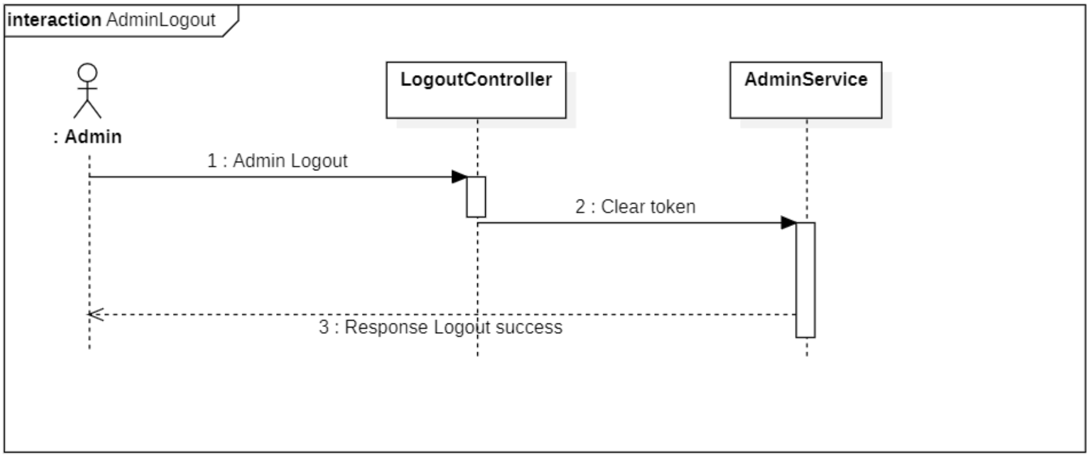
Use case	The manager logs out of the system.
Main participant	Managers
Situational goals	Managers can log out of the system.
Prerequisite	None.
Trigger	Managers decide to log out of the system at some point.
Scenes	1.The manager opens the system homepage. 2.The manager click the “Logout” button. 3.The system goes to the login page.
Exception handling	None
Priority	manager logout is the most basic function of the system and has the highest priority.
When it is valid	The first increment.
Usage frequency	Higher frequency.

Connection channels for participants	Connect to the BoYing system website through a personal computer-based browser and the Internet.
Secondary participant	None
Connection channels for secondary participants	None
Unsolved problem	1.Should the system clear all the cache in the browser?

Activity diagram



Sequence diagram

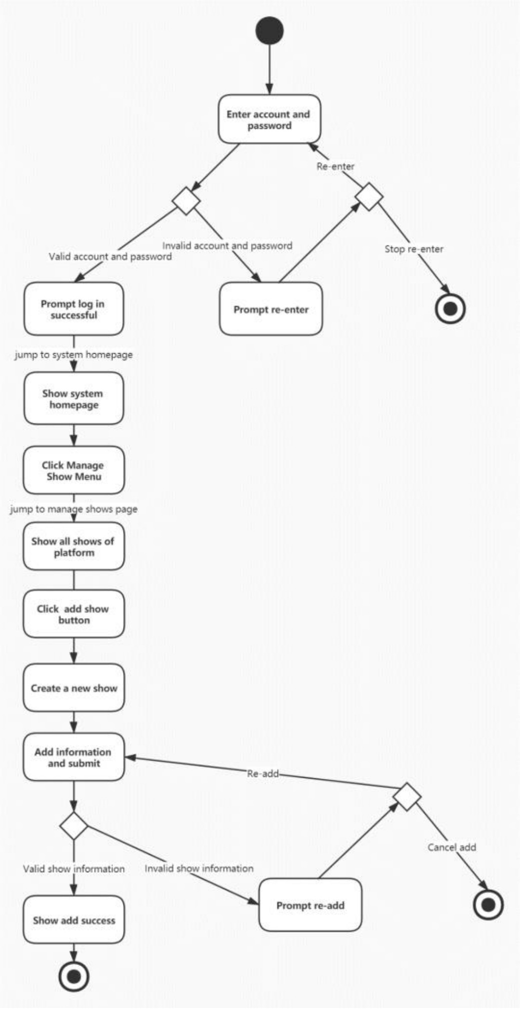


3.4.4.3 The manager adds shows to the system.

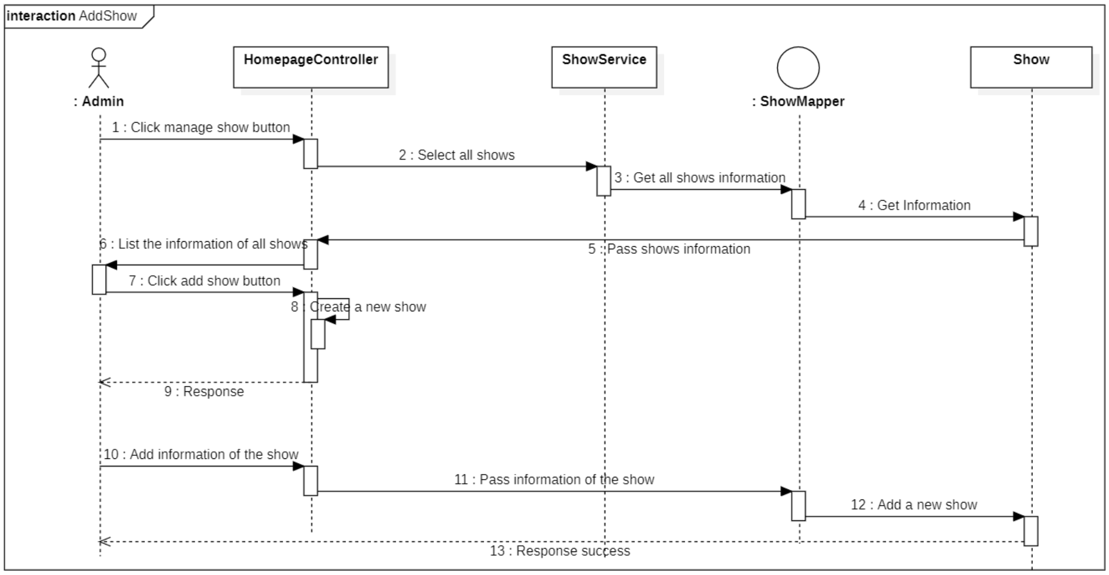
Use case	The manager adds shows to the system.
Main participant	Managers
Situational goals	The manager adds shows to the system.
Prerequisite	The manager must have been logged in.
Trigger	The manager decides to add shows to the system.
Scenes	1.The manager jumps into the homepage after logging in. 2.The manager click the “Add Shows” button. 3.The system jumps into the show-detail page. 4.The manager enters the detail information of the shows, including the name, the time to be present, the address and so on. 5.The manager submit the form. 6.The system inserts the data into the database. 7.The system display the success message.
Exception handling	1.The system fails to insert the data into the database—The system displays appropriate error messages.

	2.The information of the shows is invalid—The system displays appropriate error messages and allow the manager to re-enter it.
Priority	Adding shows is most basic function of the system and has highest priority.
When it is valid	The first increment.
Usage frequency	Common frequency.
Connection channels for participants	Connect to the BoYing system website through a personal computer-based browser and the Internet.
Secondary participant	None.
Connection channels for secondary participants	None.
Unsolved problem	1.Should the information of the shows be more restrict? Like the name of them must be special. 2.Should the system be improved to speed up the insertion of the data?

Activity diagram



Sequence diagram

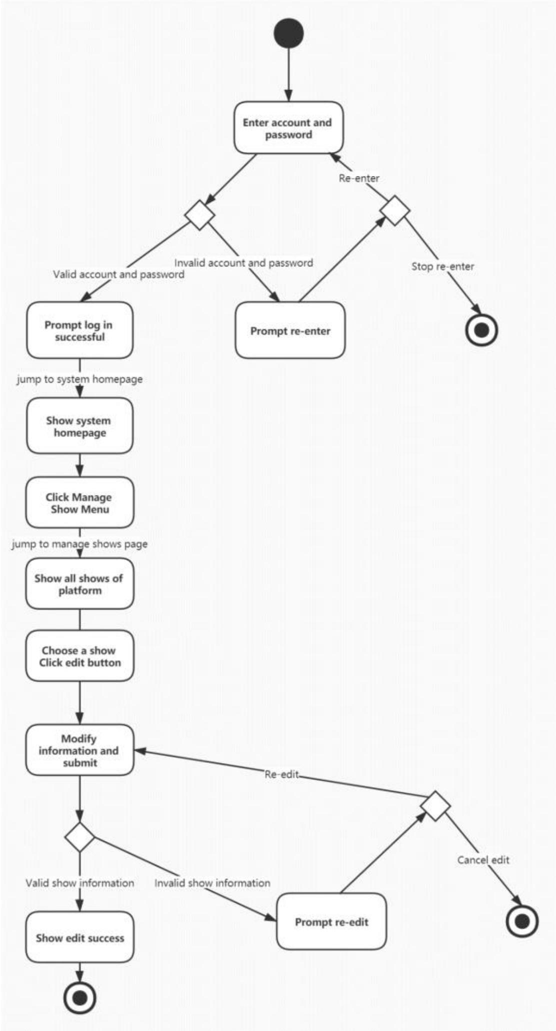

interaction AddShow
: Admin
HomepageController
ShowService
: ShowMapper
Show
1 : Click manage show button
2 : Select all shows
3 : Get all shows information
4 : Get Information
5 : Pass shows information
6 : List the information of all shows
7 : Click add show button
8 : Create a new show
9 : Response
10 : Add information of the show
11 : Pass information of the show
12 : Add a new show
13 : Response success

3.4.4.4 The manager updates the shows in the system.

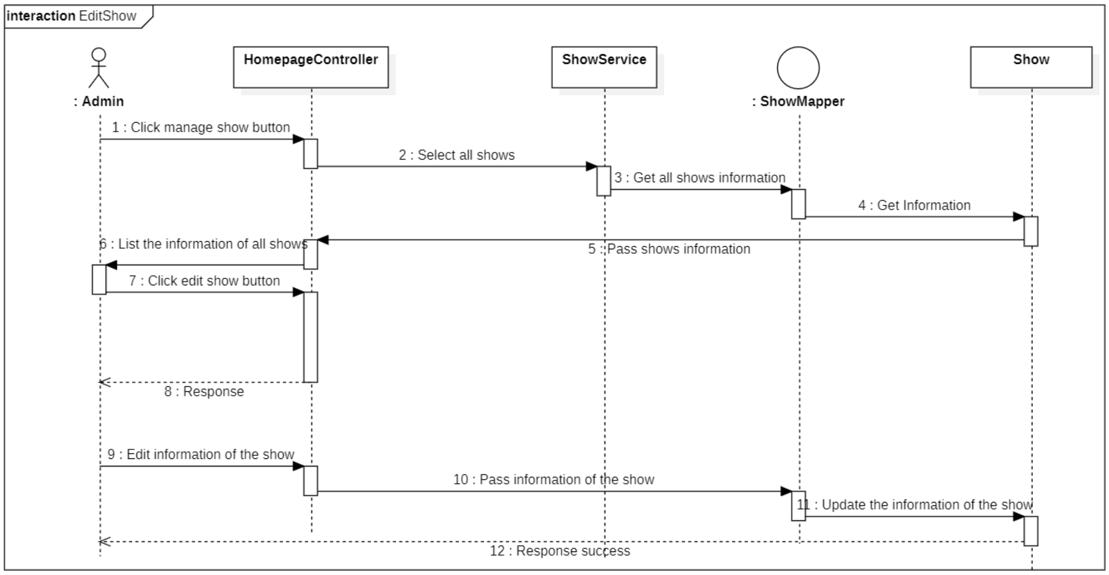
Use case	The manager updates the shows in the system.
Main participant	Managers
Situational goals	The manager updates the shows in the system.
Prerequisite	The manager must have been logged in.
Trigger	The manager decides to update the information of shows in the system.
Scenes	1.The manager jumps into the homepage after logging in. 2.The manager click the “Shows” button. 3.The system jumps into the show-detail page. 4.The manager enters the detail information of the shows, including the name, the time to be present, the address and so on. 5.The manager submit the form. 6.The system update the data into the database. 7.The system display the success message.
Exception handling	1.The system fails to update the data into the database—The system displays appropriate error messages. 2.The information of the shows is invalid—The system displays appropriate error messages and allow the manager to re-enter it.
Priority	Updating shows is most basic function of the system and has highest priority.
When it is valid	The first increment.
Usage frequency	Common frequency.
Connection channels	Connect to the BoYing system website through a personal

for participants	computer-based browser and the Internet.
Secondary participant	None.
Connection channels for secondary participants	None.
Unsolved problem	1.Should the information of the shows be more restrict? Like the name of them must be special. 2.Should the system be improved to speed up the insertion of the data?

Activity diagram



Sequence diagram

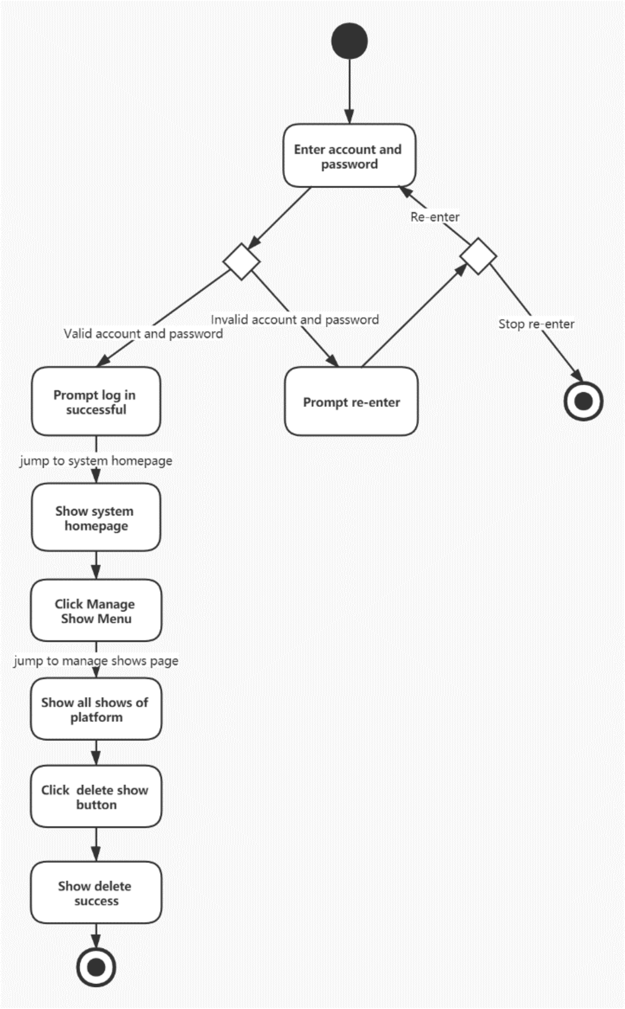

interaction EditShow
: Admin
HomepageController
ShowService
: ShowMapper
Show
1 : Click manage show button
2 : Select all shows
3 : Get all shows information
4 : Get Information
5 : Pass shows information
6 : List the information of all shows
7 : Click edit show button
8 : Response
9 : Edit information of the show
10 : Pass information of the show
11 : Update the information of the show
12 : Response success

3.4.4.5 The manager deletes the shows in the system.

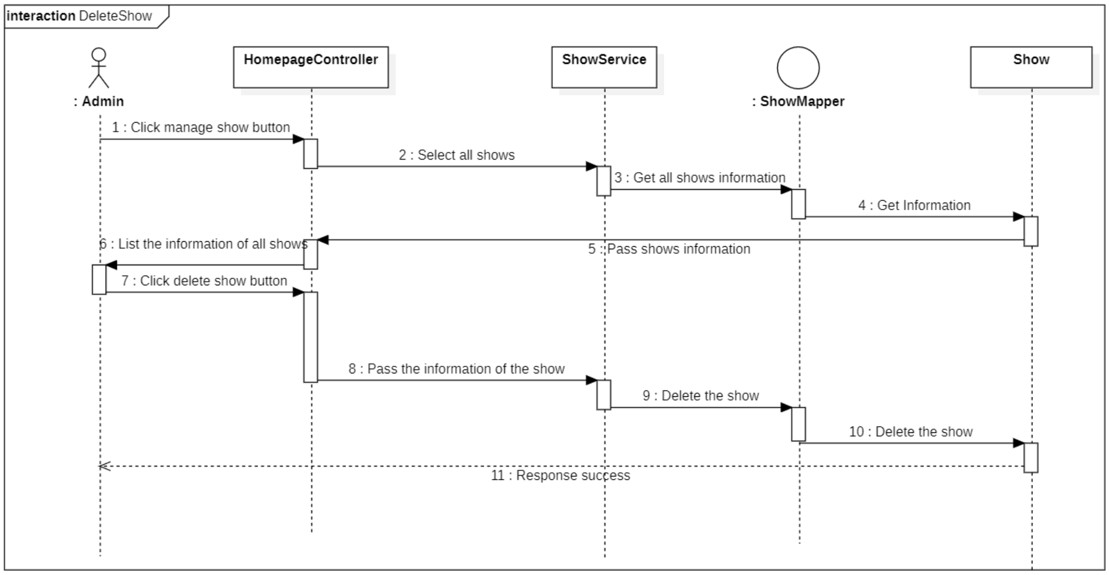
Use case	The manager deletes the shows in the system.
Main participant	Managers
Situational goals	The manager deletes the shows in the system.
Prerequisite	The manager must have been logged in.
Trigger	The manager decides to deletes the information of shows in the system.
Scenes	1.The manager jumps into the homepage after logging in. 2.The manager click the “Shows” button. 3.The system jumps into the show-detail page. 4.The manager click the “Delete” button. 5.The system deletes the data into the database. 6.The system display the success message.

Exception handling	1.The system fails to delete the data into the database—The system displays appropriate error messages.
Priority	Deleting shows is most basic function of the system and has highest priority.
When it is valid	The first increment.
Usage frequency	Common frequency.
Connection channels for participants	Connect to the BoYing system website through a personal computer-based browser and the Internet.
Secondary participant	None.
Connection channels for secondary participants	None.
Unsolved problem	1.Should the system be improved to speed up the deletion of the data?

Activity diagram



Sequence diagram

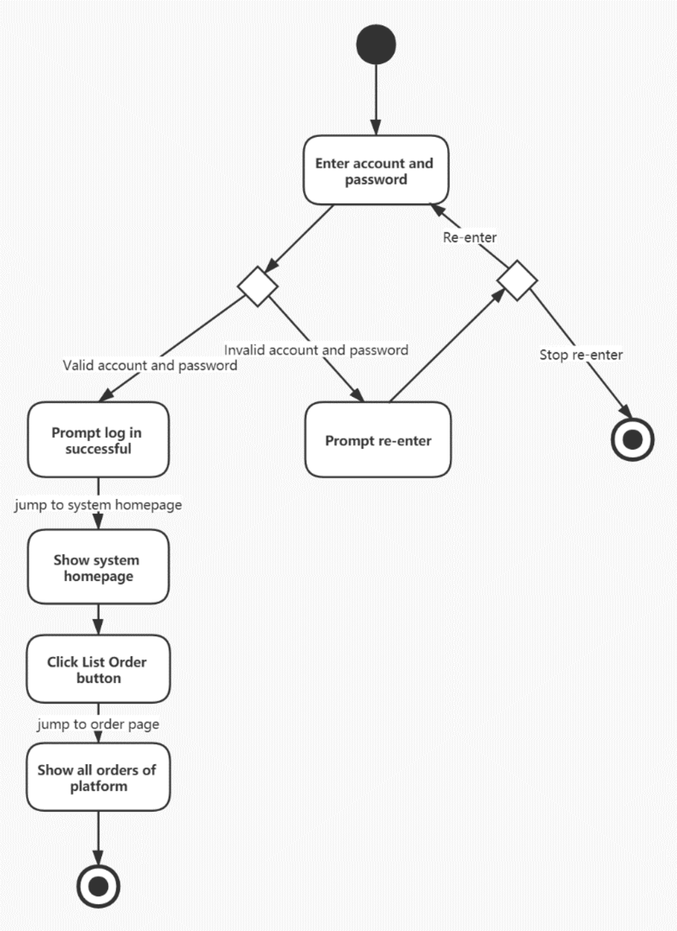

interaction DeleteShow
: Admin
HomepageController
ShowService
: ShowMapper
Show
1 : Click manage show button
2 : Select all shows
3 : Get all shows information
4 : Get Information
5 : Pass shows information
6 : List the information of all shows
7 : Click delete show button
8 : Pass the information of the show
9 : Delete the show
10 : Delete the show
11 : Response success

3.4.4.6 The manager views all the orders in the system.

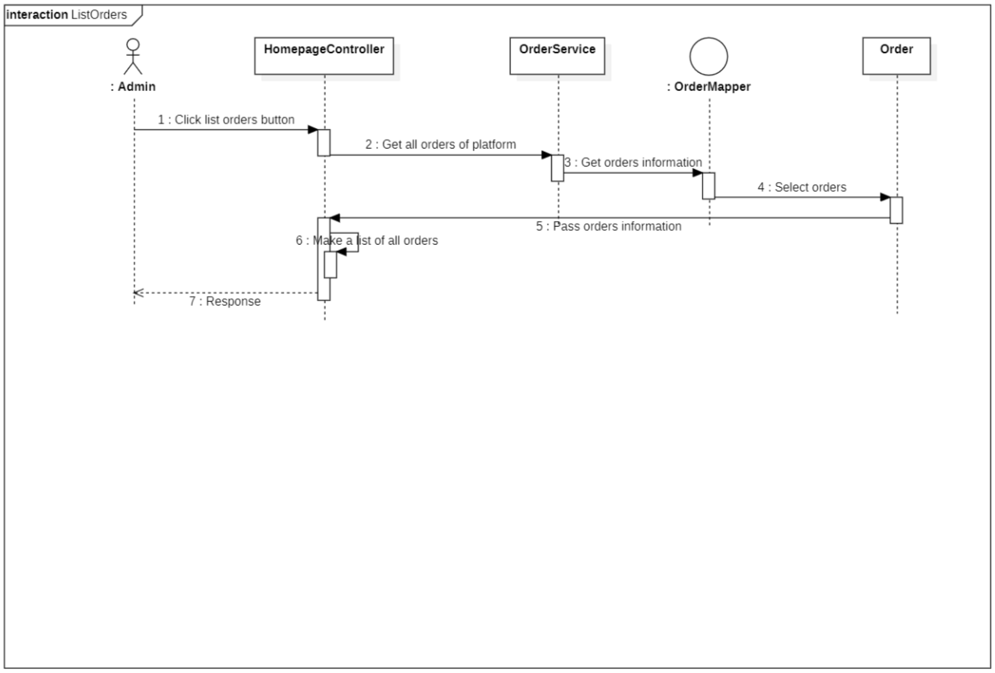
Use case	The manager views all the orders in the system.
Main participant	Managers
Situational goals	The manager views all the orders in the system.
Prerequisite	The manager must have been logged in.
Trigger	The manager decides to views all the orders in the system.
Scenes	1.The manager jumps into the homepage after logging in. 2.The manager jumps into the orders page after clicking on the "Orders" button. 3.The system lists all the orders in the system. 4.The manager browsers the information of the orders.
Exception handling	1.The system fails to get the data from the database—The system displays appropriate error messages.
Priority	Viewing orders is most basic function of the system and has highest priority.
When it is valid	The second increment.
Usage frequency	Common frequency.
Connection channels for participants	Connect to the BoYing system website through a personal computer-based browser and the Internet.
Secondary participant	None.
Connection channels for secondary participants	None.
Unsolved problem	1.Should the system be improved to speed up the query of

	the data? 2.Should the system only display a part of the orders when there is a large amount of orders?

Activity diagram



Sequence diagram

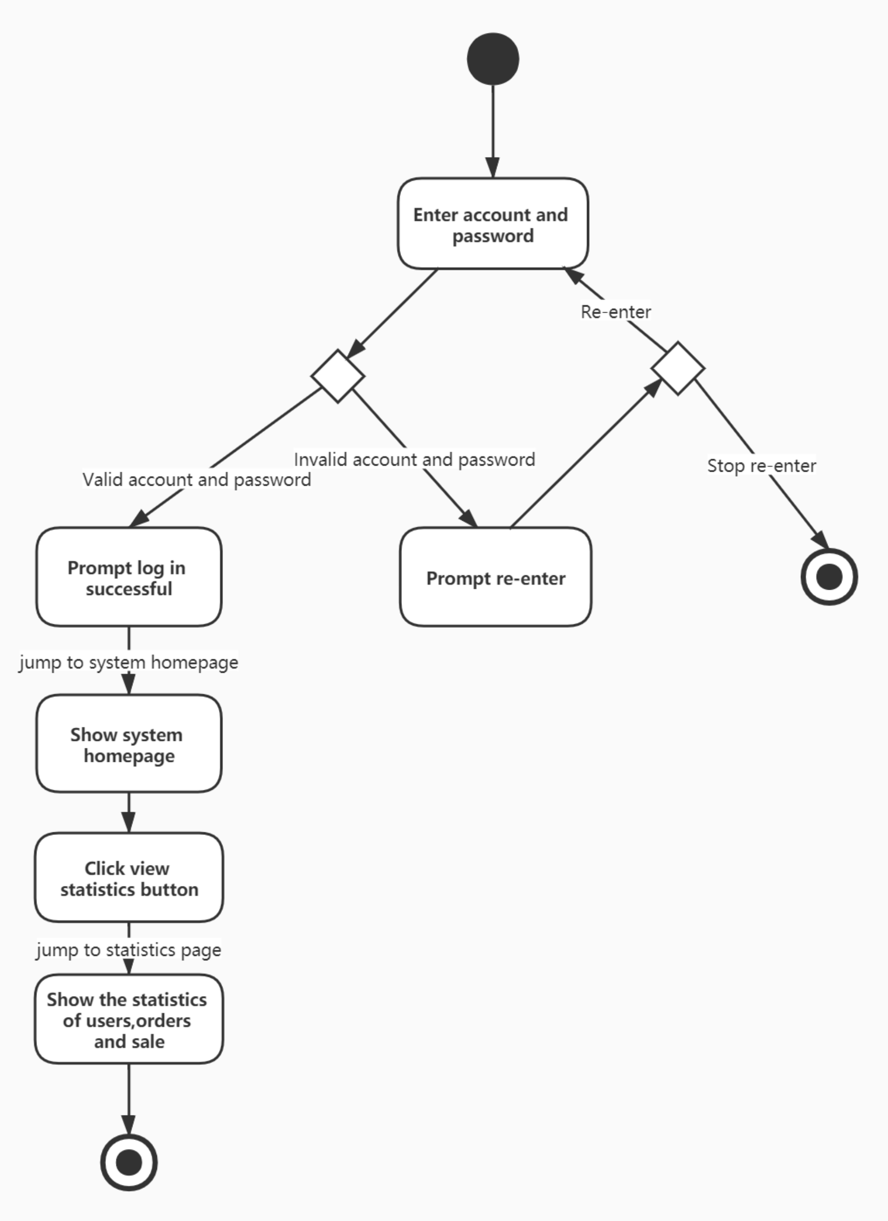

interaction ListOrders
: Admin
HomepageController
OrderService
: OrderMapper
Order
1 : Click list orders button
2 : Get all orders of platform
3 : Get orders information
4 : Select orders
5 : Pass orders information
6 : Make a list of all orders
7 : Response

3.4.4.7 The manager views the statistics in the system.

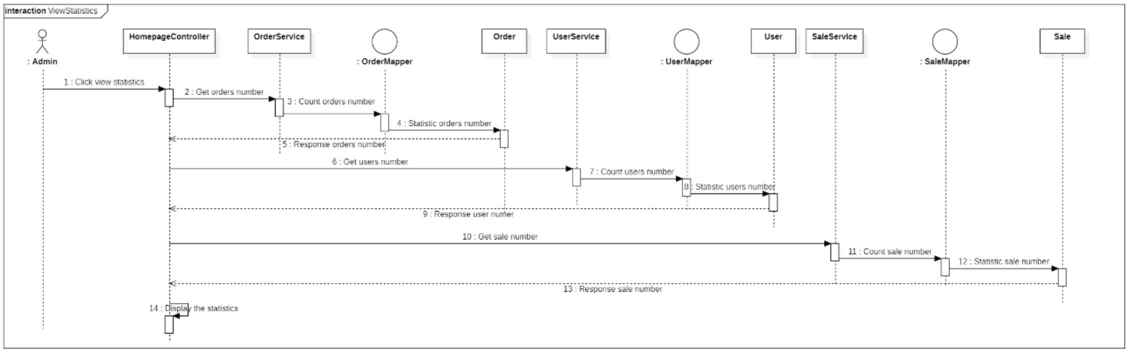
Use case	The manager views statistics served by the system.
Main participant	Managers
Situational goals	The manager views statistics served by the system.
Prerequisite	The manager must have been logged in.
Trigger	The manager decides to views statistics served by the system.
Scenes	1.The manager jumps into the homepage after logging in. 2.The system displays all the statistics in the system daily 4.The manager browsers the statistics of the sales.
Exception handling	1.The system fails to get the data from the database—The system displays appropriate error messages.
Priority	Viewing statistics is most basic function of the system and

	has highest priority.
When it is valid	The second increment.
Usage frequency	Common frequency.
Connection channels for participants	Connect to the BoYing system website through a personal computer-based browser and the Internet.
Secondary participant	None.
Connection channels for secondary participants	None.
Unsolved problem	1.Should the system be improved to speed up the query of the data? 2.What should be computing as a statistics about the sales?

Activity diagram


Enter account and password
Re-enter
Valid account and password
Invalid account and password
Stop re-enter
Prompt log in successful
Prompt re-enter
jump to system homepage
Show system homepage
Click view statistics button
jump to statistics page
Show the statistics of users,orders and sale

Sequence diagram



3.5 Performance requirements

Here are some basic performance requirements for the product under various circumstances.

Accessing Speed:

Sometimes the system will get a lot of web queries at a very short time caused by a large number of users buy tickets at the same time, which will slow down the requesting speed and bring bad experience to the users. The MySQL database will be deployed in this system to handle the large queries at a short time, which is known for its high performance. And at the same time, the system will increase the proportion of SQL language programming to achieve high speed.

Reduce Memory Occupancy:

Sometimes shows may have a lot of detailed information, so when the user getting into the show-detail page, the data flow between frontend and backend will be greatly high, which will lead to high occupancy of memory and slow loading. The way to handle this is to query part of the data when the page first loading, and dynamically do the query of other data when needed.

Algorithm Improvement:

The system should create indexes in the database to speed up the query of the users. Besides, the system should use more efficient algorithm, both in time and in space.

3.6 Safety Requirements

In this part, we'll specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of this product.

Avoid database failure:

The database may fail when getting a large number of queries at a very short time. The Redis will be deployed to avoid the failure of MySQL database. When the frontend doing the query, it will first get into the Redis to do the query and return if the data is available, and if not, it will then get into the MySQL for data and update the Redis's information. In this way, the MySQL database will get less query and the possibility of failures will decrease.

Robustness:

To make sure all data request from the frontend is legal, all the functions in the backend should check the session, which is where all the essential information of the front user are stored, and then run the function if the session suggest the request is legal, while if not, the function should return an error message to the frontend. To make sure all data flow into the database is by standards valid, the product will check all input information from the frontend which is where the data come from, to the database, which is where the data arrives. In this way, all data we gain is valid and can fit in our database correctly.

3.7 Security Requirements

This is the part that specifies requirements regarding security or privacy issues surrounding use of product or protection of data used or created by the product.

Specific Data Access Authority:

The system must authorize the users and the managers specifically. Different groups of users have different access to different data. And the users can't access the managing system to make sure the data in the system stay in the absolute control of the managers. Besides, the users can only refund tickets before the shows started. There are also other regulations to keep our data safe.

Prevent Injection SQL Query:

Applying string splicing to form a SQL query is the easiest way to do SQL query in the backend. However, this method brings about the problem of injection SQL query, and basically makes it an opportunity for users to gain illegal access to data that are not supposed to be accessed by regular users. In this system we choose to apply MyBatis framework that allows us to encapsulate the SQL query with java functions. This perfectly avoid the situation of illegal access.

3.8 Software system attributes

In this part, we will specify some additional quality characteristics for the product that will be important to either the users and administrators.

Adaptability:

We expect our web application to run smoothly and correctly on most mainstream web explorers (Chrome, Firefox, Edge, Opera, etc.) and operating systems (Windows, MacOS, Linux, etc.).

Maintainability:

We expect our product to be able to record all data flow since released and requests of any kind of those data can be done in no time. Besides, the regular maintaining and updating operations have no influence on our previous data.

Interoperability:

Our applications on different platforms are supposed to easily communicate with each other. They all run on the same server and database.

Robustness:

Any kind of data query and any kind of data inserts, whether legal or not, will never result in system crush. All errors will be properly handled and represented.

3.9 Other requirements

Data Dictionary Requirement

Data Item:

The specific description of the data item in the data structure is required. It should contain the following information: the item name, simple explanation of the item, data type, data length, range of value, meaning of certain values and logical relation with other data items. This will bring convenience to the developers.

Data Structure:

Data structure suggests the combination relation between data. The description of data structure should contain the data structure name, specific explanation and content (data items or substructures).

Data Flow:

Data flow describes the route of data transportation. The description should include: data flow name, specific explanation, data flow source, data flow direction, content (data structures), average flow, and peak flow.

Data Storage:

It's the place that hold the data structures. It also serves as one of the sources or directions of data flows. Description of the follow contend is needed: data storage name, specific explanation, ID number, flowing in data, flowing out data, content (data structure), data amount, access mode.

Data Process:

In the data dictionary, we only need description of those information: process name, specific explanation, input (data flow), output (data flow), process method.

4. References

[1] Prof. Betty H.C. Cheng, "Homework 2", CSE 435, East Lansing, MI, September 2007.

[2] Prof. Betty H.C. Cheng, "Requirements Assignment", CSE 435, East Lansing, MI, September 2007.

[3] Mr. Borzoo Bonakdarpour, Elicitation Meeting, September 25th 2007.

[4] IEEE-SA Standards Board, “IEEE Recommended Practice for Software Requirements Specifications”, Software Engineering Standards Committee of the IEEE Computer Society, June 25th 1998

[5] Prof. Betty H.C. Cheng, “Intro to Specifications”, CSE 435, East Lansing, MI, September 2007.

[6] Blossom, Emelander, Gebhard, Meyer, “CSE 435: Group 4”, http://code.google.com/p/cse435-group4/, October 2007.

[7] Constance L. Heitmeyer, “Software Cost Reduction”, Naval Research Laboratory, Washington, DC, 2002.

[8] Prof. Betty H.C. Cheng, “Operational vs Descriptive Models (including SCR)”, CSE 435, East Lansing, MI, October 2007.