成都理工大學中英合作办学
CDUT Sino-British Collaborative Education

OXFORD
BROOKES
UNIVERSITY

UNDERGRADUATE PROJECT REPORT

Project Title:	**A Smart Web-based Toy Customization System**
Surname:	**Yiting**
First Name:	**Wang**
Student Number:	**201918020103**
Supervisor Name:	**Dr.Joojo Walker**
Module Code:	**CHC 6096**
Module Name:	**Project**
Date Submitted:	**May 5, 2023**

Chengdu University of Technology Oxford Brookes College

Chengdu University of Technology

BSc (Single Honours) Degree Project

Programme Name: **Software Engineering**

Module No.: **CHC 6096**

Surname: Yiting

First Name: Wang

Project Title: A Smart Web-based Toy Customization System

Student No.: 201918020103

Supervisor: Dr.Joojo Walker

2ND Supervisor (if applicable): **Not Applicable**

Date submitted: **May 5, 2023**

A report submitted as part of the requirements for the degree of BSc (Hons) in Software Engineering

At

Chengdu University of Technology Oxford Brookes College

Declaration

Student Conduct Regulations:

Please ensure you are familiar with the regulations in relation to Academic Integrity. The University takes this issue very seriously and students have been expelled or had their degrees withheld for cheating in assessment. It is important that students having difficulties with their work should seek help from their tutors rather than be tempted to use unfair means to gain marks. Students should not risk losing their degree and undermining all the work they have done towards it. You are expected to have familiarised yourself with these regulations.

https://www.brookes.ac.uk/regulations/current/appeals-complaints-and-conduct/c1-1/

Guidance on the correct use of references can be found on www.brookes.ac.uk/services/library, and also in a handout in the Library.

The full regulations may be accessed online at https://www.brookes.ac.uk/students/sirt/student-conduct/

If you do not understand what any of these terms mean, you should ask your Project Supervisor to clarify them for you.

I declare that I have read and understood Regulations C1.1.4 of the Regulations governing Academic Misconduct, and that the work I submit is fully in accordance with them.

Elsa

Signature Date 23/05/03

REGULATIONS GOVERNING THE DEPOSIT AND USE OF OXFORD BROOKES UNIVERSITY MODULAR PROGRAMME PROJECTS AND DISSERTATIONS

Copies of projects/dissertations, submitted in fulfilment of Modular Programme requirements and achieving marks of 60% or above, shall normally be kept by the Oxford Brookes University Library.

I agree that this dissertation may be available for reading and photocopying in accordance with the Regulations governing the use of the Oxford Brookes University Library.

Elsa

Signature Date 23/05/03

Acknowledgment

First of all, allow me to express my gratitude to Dr. Joojo Walker. He supplied me with sound guidance as I developed the project, went through the revision process, and carefully reviewed and fixed any flaws in the final report. This undertaking would not have been finished without him. Second, I would like to express my appreciation to my parents for their unwavering love, belief in me, and tolerance. Finally, I would like to recognize all the instructors and classmates who have motivated me to acknowledge my weaknesses and keep growing. I want to thank everyone for leaving me with priceless memories throughout my time at Oxford Brookes University, from my first to my final year. I will also take on the future with a grateful heart.

Table of Contents

Abstract

A rising number of toy companies are leveraging the explosive growth of e-commerce to develop online platforms promoting brand development, offering branded goods and features. As a result, a web application enabling customizations of branded toys has been developed to cater to consumers' specific and general needs. This website facilitates basic online sales services for toy brands, such as registration, log in, browsing and purchasing toys, and provides introductions to the brand. In addition, customers are granted access to a personalized toy service, where they can interactively customize the baseline model (teddy bears) using fashion accessories, then purchase the decorated product. The smart toys customization system has three primary development stages, comprising the construction of the database and back-end functionalities employing SQL management server tools and Springboot, programming of front-end functionalities using Vue, and the creation and rendering of toy models utilizing 3dsmax and Three.js. Overall, the project implementation was successful, with thorough testing and evaluation conducted to determine the web application's useful functionalities..

Keywords: **Online Toy Brand, Web-based Toy Customization System, Vue, Springboot**

Abbreviations

Abbreviations	Definition
Vue	Vue.js
HTML	Hyper Test Markup Language
CSS	Cascading Style Sheets
IDE	Integrated Development Environment
IDEA	IntelliJ IDEA. A tool to develop software and an integrated development environment with improvement suggestions and precise explanations of the Java language
VSCode	Visual Studio Code
GUI	Graphical User Interface
UI	User Interface
SQL	Structured Query Language
BDD	Behavior Driven Development. A strategy written use cases and expectations prior to the creation of a program, used in agile development to describe the behavior of the application and narrow the gap between expectations and outcomes
ERD	Entity Relationship Diagram
URL	The address of a unique internet resource or file
IT	Information Technology
HTTP	HyperText Transfer Protocol
DOM	Document Object Model

CPU	Central Processing Unit
XML	Extensible Markup Language
MVC	Model View Controller

Glossary

Glossary	Definition
Front-end	The user-visible interface, such as effects, layout, images, video, audio, and other content on a web page
Back-end	The majority of the time, site data saving and reading, as well as data interaction with the front-end engineer, are invisible to the user.
Vue	Vue is an open-source MVVM front-end JavaScript framework for creating user interfaces, and a web application framework for creating single-page applications
Sprintboot	Springboot is a rapid development framework that enables rapid integration of third-party frameworks
render	Rendering is the process of building a 3D model, mapping it, establishing the lighting, and utilizing software to match and blend these elements with the model to produce a realistic-looking image that resembles a photograph.
Three.js	A JavaScript library or API is utilized by the cross-browser script known as Three.js to produce and show dynamic 3D computer visuals.
3dsMax	3D Studio Max. 3dsMax is the mainstream and mature software to construct models.

Chapter 1 Introduction

1.1 Background

The e-commerce era began in the 1990s, with Amazon and eBay paving the way for e-commerce in the early 1990s. The ensuing decades witenessed a continued expansion and development of e-commerce. With today's rapid developments in the IT sector, as well as the expansion of the internet and the globalization of e-commerce, an increasing number of industries are offering online platforms that allow customers to purchase products from around the world without leaving their homes. Currently, in the toy industry, a growing number of brands such as Lego[1] and Jellycat[2] have created their online websites to showcase the brands' culture, history, and merchandise, and provide consumers with sophisticated online sales services by understanding consumers' shopping habits, spending tendencies and behavioral patterns. Besides, customized products are growing in popularity with the general public at a time when people are more aware of the experience of the products they buy and use daily and value and support originality. According to [3], under the impact of covid-19, customized goods have a considerably lower return rate than regular ones. Customers will establish a psychological attachment to a product and are less likely to return products if marketers encourage consumers to spend more time and experience with products, making products an extension of themselves. Examples include the release of sneakers from Nike[4] that can be customized online, the ability to add text to Coke cans from Coca-Cola, and the introduction of personalized headphone cases from Apple[5]. There is no denying that these customized services encourage more consumers to purchase customized products and receive positive reviews. Customized services not only address the requirements of specific customers but also boost customer satisfaction and loyalty. The toy business has yet to introduce a web application where consumers can both purchase toys online and customize a toy with fashion accessories. A web-based toy store has been considered, given that customers cannot visit a real store to customize or purchase toys for a variety of reasons, including the distance to offline stores and the limited time available to choose fashion accessories, and so forth.

1.2 Aim

The web application will be principally designed for users who want to design their exclusive toy for themselves or others such as parents, lovers, children, and so forth. Additionally, the web application provides a wide selection of non-customized toys for customers to browse and order. Besides, the web application provides the culture, history, contact information, and other related information of the brand for consumers.

1.3 Objectives

The objectives of this project are as follows:

1. Carry out a comprehensive background study and investigation of existing online toy stores
2. Carry out a comprehensive background study and investigation of existing online brands providing services for customizing
3. Design the backend components (database and backend logic) of the web application
4. Design the frontend components (User Interface and JavaScript) of the web application
5. Design the basic toy model and related fashion accessories
6. Testing and maintenance of the web application

1.4 Project Overview

1.4.1 Scope

The present study focuses on a web application built as an online toy brand aimed at offering customers basic online shopping services, non-customized toy products, and customized toy options. The Customized Toys module provides guidelines for customers to follow while designing their own toys and is equipped with graphical user interfaces that enable modifications to the baseline teddy bear model using different accessory buttons such as hats, headgear, hoods, and glasses. After completing this customization process, customers can instantly place orders for the new design. Moreover, the web application facilitates efficient management of toy products and their details which includes diverse categorizations of toy types, promotions, and detailed product information. These features enable users to browse product offerings, locate target toys quickly, view complicated product profiles, and place orders with ease. Customers can add toy products to their respective carts and make changes as necessary after registering and logging into their accounts. Furthermore, the address management subsystem aids customers in updating, adding new or deleting existing information on addresses during the ordering process. After logging in, the system simultaneously updates both the information related to the order and the logistics status displayed on the customer's account, allowing customers to check the order and logistics information conveniently. With global users in mind, the website's language will be set to English, allowing people from all over the world to browse and design toys.

The finished web application will allow users to browse, customize, and order toys using graphical and textual descriptions. Changes to relevant data, such as modifying personal information, adding and removing delivery addresses, setting default addresses, adding items to the shopping cart, and so on, will be sent to the back end via HTTP requests and processed as needed. The back end will also return the operation's results in Jason format to the front end for user feedback on the operation.

1.4.2 Audience

Audiences can be of the following categories:

1. Parents who require toys to accompany their children
2. Parents who require to reward their children
3. People who require toys as gifts
4. Children who require toys
5. People who require customized toys as a gift
6. People who require to have their unique toy

Chapter 2 Background Review

2.1 Summary of existing approaches

2.1.1 Summary of the existing mainstream online toy/shoes stores

Web application / Function	Jellycat [2]	POP MART[6]	NIKE[4]	LEGO[1]	Mattel[7]	Vans[8]
Make an order	√	√	√	N/A	√	√
Check logistic	√	√	√	N/A	√	√
Customize toys/shoes	N/A	N/A	√	N/A	N/A	√
Sign up and log in	√	√	√	√	√	√
Shopping Cart	√	√	√	N/A	√	√
Navigation	√	√	√	√	√	√

Table 1 The existing online toy/shoes stores

Previous online shopping stores[1], [2], [4], [6]–[8] are the mainstream brands to sell products online to customers around the world. The following is a comparison and analysis of the functional and design aspects of each website. With regard to functionality, Previous online shopping stores[1], [2], [4], [6]–[8] provide services including browsing products, searching for aimed products, categorizing products to make them easier to find, and signing up and logging in or off customers' accounts. LEGO[1] provides a variety of languages, currencies, and nations for the convenience of its international customers, a large selection of toys, including themed sets, limited editions, and more, as well as in-depth product information, reviews, and ratings that help customers gauge the efficacy and quality of the products. For purchasers to understand the brand, POPMART[6] offers a variety of product categories and greater brand culture. Excellent customer service is provided by Mattel [7], including after-sales support and delivery assurances.

Web application	**Jellycat**[2]	**POP MART**[6]	**NIKE**[4]	**LEGO**[1]	**Mattel**[7]	**Vans**[8]
Front-end Technologies	HTML+CSS+JavaScript	HTML+CSS+JavaScript	HTML+CSS+JavaScript	HTML+CSS+application/Jason	HTML+CSS+JavaScript	HTML+CSS+JavaScript

Table 2 The Front-end technologies used in the existing online toy/shoes stores

According to Table 2, all the websites utilize HTML and CSS to structure and style the layout of the pages, and javascript and jquery to implement the interaction and animation effects of the pages.

Web application	**NIKE**[4]	**Vans**[8]
Redesign	N/A	√
Withdrawn	N/A	√
Virtual Reality shoes	N/A	√
Share design	√	√
Save design	√	√

Table 3 The comparison of customizing function

In the context of shoe customization, NIKE[4] and Vans [8] adopt different approaches to cater to the diverse needs of users. NIKE[4] offers an extensive range of shoes, along with a wide range of customization options such as tongue, laces, and box variants. Likewise, Vans[8] offers its users the ability to choose custom details randomly, adding a layer of fun to the customization process. Moreover, Vans[8] provides options for redesigning and withdrawing, which enables users to restart or cancel previous operations without re-entering the design page.

Incorporating virtual reality during the shoe design process is a unique feature available with Vans[8], wherein users can control the arrows to view the complete range of shoe designs. However, NIKE[4] has a limitation in terms of visualizing their shoes, as users can only browse pictures of each part of the shoes rather than viewing a spatial shoe model.

Both online shopping stores, NIKE[4] and Vans[8], have the option to share designed shoes, allowing users to share ideas and design with others. Overall, NIKE[4] and Vans [8] have distinctive features that offer users varying levels of customization, making it possible to find the right fit for individuals' preferences in shoe design.

Chapter 3 Methodology

3.1 Requirements and specification

The requirements of the web application are outlined by use cases. The primary purpose of a use case diagram is to show what subsystem in the web application is performed for which actor. Considering the features of online toy brands, the web application is divided into four subsystems to perform.

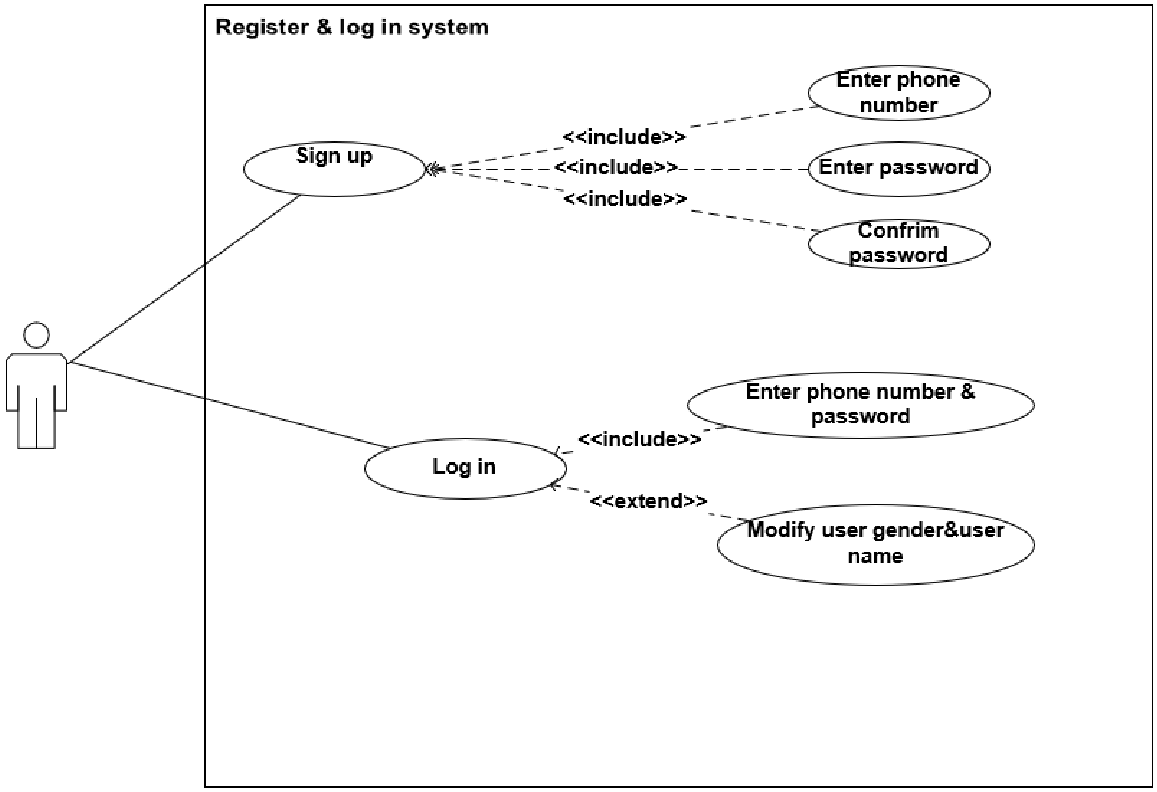


Figure 1 Use case of Register and Login system

Figure 1 is shown a subsystem of register and login of the web application, where users register their account by inputting their telephone number and password. After registering an account successfully, users could log in successfully by inputting the registered telephone number and the corresponding password. Users could complete, and modify their profiles.

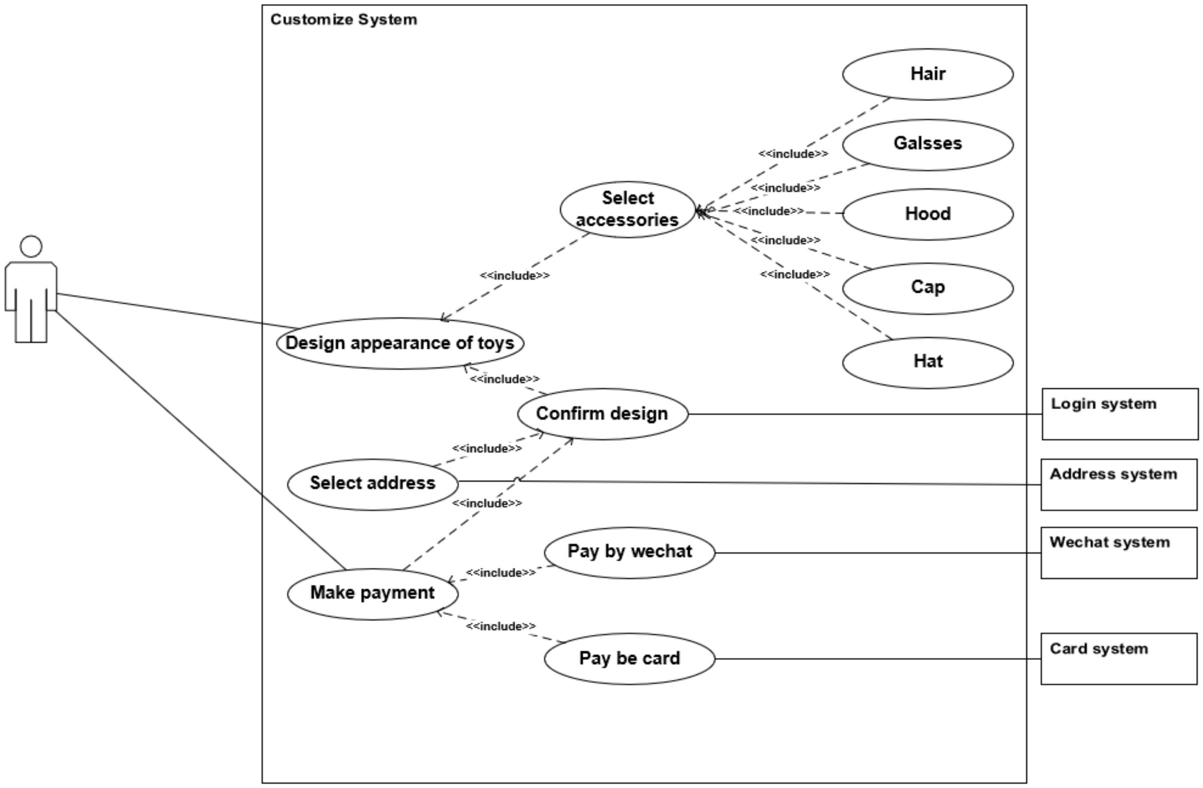


Figure 2 Use case of Customizing Toys system

Figure 2 is shown a subsystem of customization of this web application, where users will be given a basic toy model as well as several attachments. With the push of a button, the user can render the accessories all at once. After pairing is complete, the user must log in and confirm the pairing before proceeding with the payment.

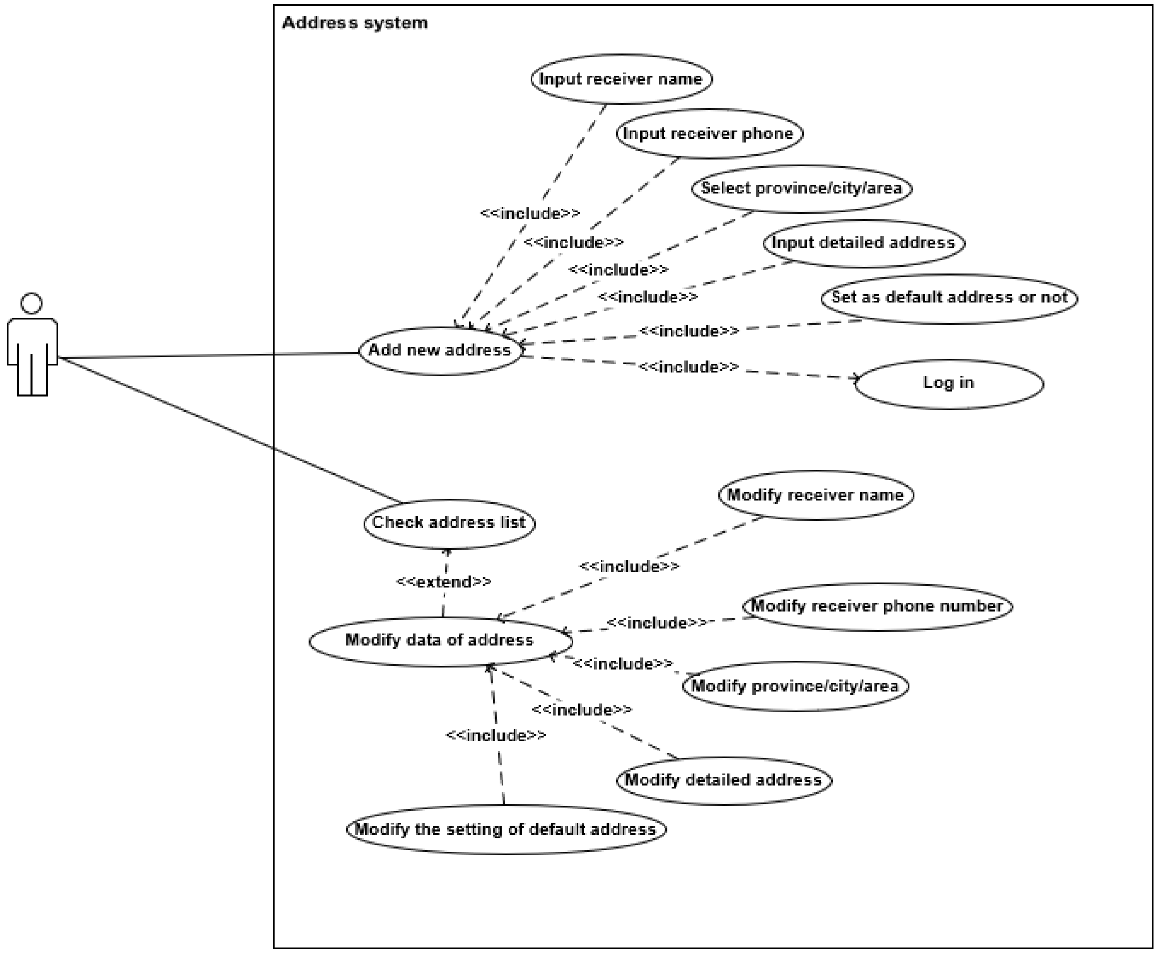


Figure 3 Use case of Address system

Figure 3 is shown a subsystem of addresses of the web application, where users could add new addresses by inputting the receiver’s name, telephone number, and address and set as the default address or not after logging in account. In addition, a list of delivery addresses can be displayed for users and users have the ability to change, delete, and set the default address in the collecting list.

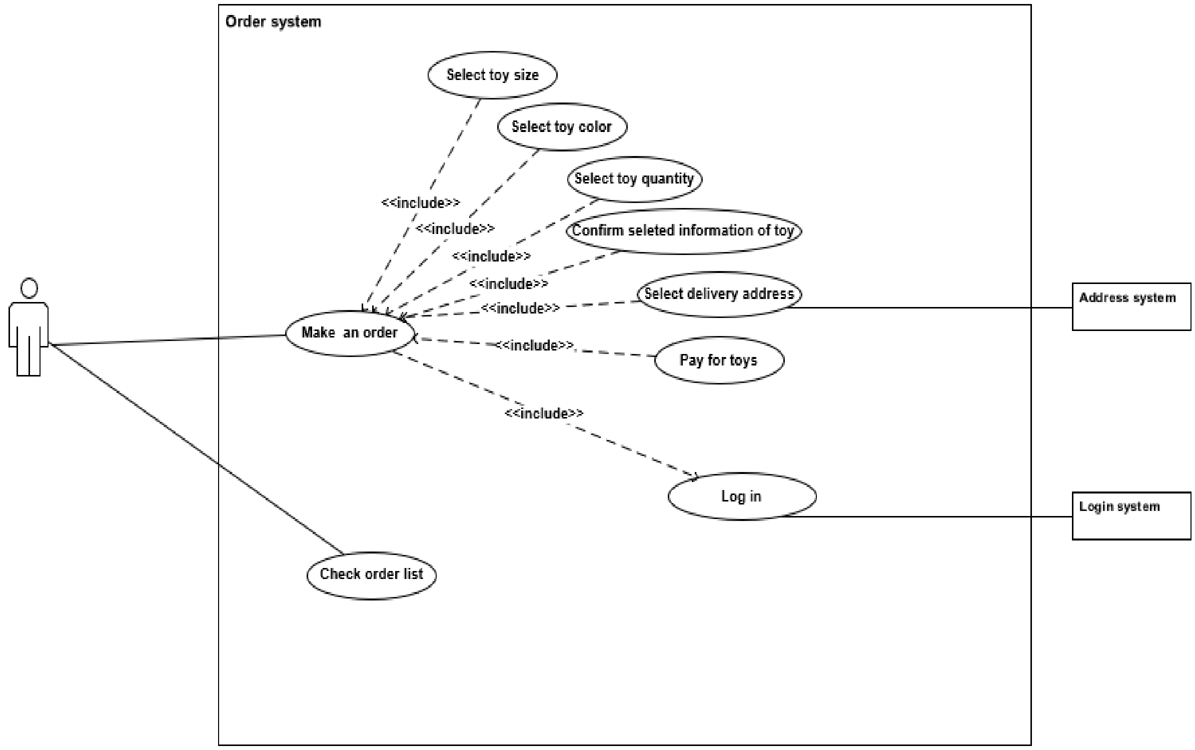


Figure 4 Use case of Order system

Figure 4 is shown a subsystem of the order of the web application, where users can make an order after logging in. Once users select the necessary data for the order, users could pay for toys.

3.2 Approach

The approach of this project is as follows:

Research and development methodology	Approach
Software development model	Agile: considering the insufficient experience in developing software, Agile software development could adopt changes from any stage of the whole

	software development so that it could decrease the cost [9]
Requirement gathering and specification method	Use cases: stories are described the requirements from the users' perspective following people around Prototyping: Gathering initial requirements that users require, showing the initial version of the design to users who provide additional requirements, changing the web application, and cycling around with the users again. Brainstorming: the basic model and fashion accessories of customized toys are to be started with creative brainstorming.

Table 4 The related approach of the project

3.3 Technology

This section discusses the applied technologies for developing and testing the website.

3.3.1 Development environment of back-end: IDEA

Considering this back-end of the web application will be principally based on Java language to develop and I had the experience of using IDEA-programmed coursework and set-related configurations, IDEA[10] will be selected to develop the back end of this web application.

3.3.2 Development environment of front-end: VScode

Considering the front-end will be principally based on Vue, HTML, and CSS language, VScode[11] is an excellent and mature editor which supports to install extensions to add Vue and serverless computing, relational and document-based data storage and querying, and deployment and hosting of Vue.

3.3.3 Database

This web application that is being developed will require a database to store the data of products, orders, and users and a tool to manage each table of data. Navicat Premium[12] compares and synchronizes data between databases/schema through a thorough analysis process, making data migration easier and quicker through data transfer, data synchronization, and structure synchronization. Considering the above advantages, Navicat Premium[12] would be used for producing and managing the database

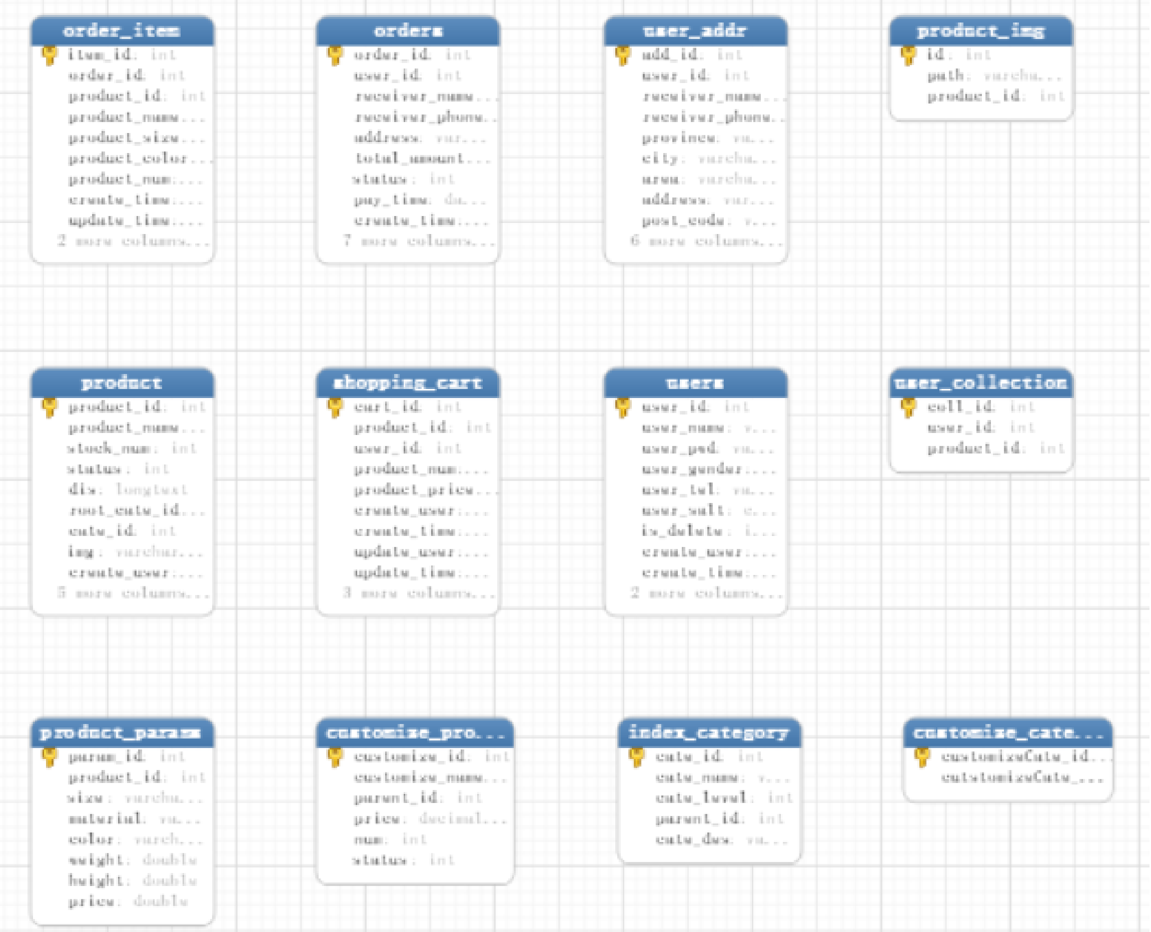


Figure 5 ERD of the web application

3.3.4 Development Methodologies

3.2.4.1 Development method between front and back ends

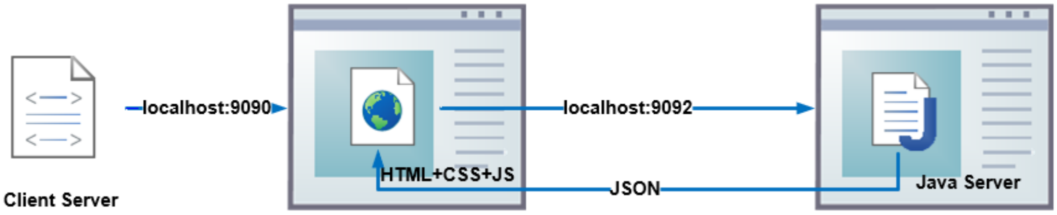


Figure 6 The structure of separation of front and back ends

Figure 6 shows the method of connecting the front and back ends. The process of passing data from client-server to Java Server. The front-end page serves as the client's visualization page, receives client requests, executes the associated request methods, and sends the data to the back end when it has been normalized. As necessary, the back end of the website calls the appropriate methods, and adds, deletes, or updates data in the database after receiving requests from the front end.

Development Method	Reason
Separation of front and back end	1. It simplifies the procedure of locating bugs by facilitating the ability to determine if they are in the front or back ends. 2. It reduces load pressure on back-end servers 3. The back-end server down or not running or timeout will not affect the front-end page running and loading (except for data in the database) 4. It increased development efficiency due to the parallel development of front and back ends

Table 5 The reason for selecting the separation of front and back end

3.2.4.2 Development technologies of back-end system

Back-end technology	Reason
Spring	The dependencies between objects are given to Spring for easy decoupling and simplified development. Spring provides integration support for other good open-source frameworks

Mybatis	Mybatis separates the SQL statements from the Java source program and writes them in a separate XML file, making it much easier to maintain the program
SpringMVC	SpringMVC develops web applications cleaner and connects Spring better

Table 6 Back-end technologies used in the web application

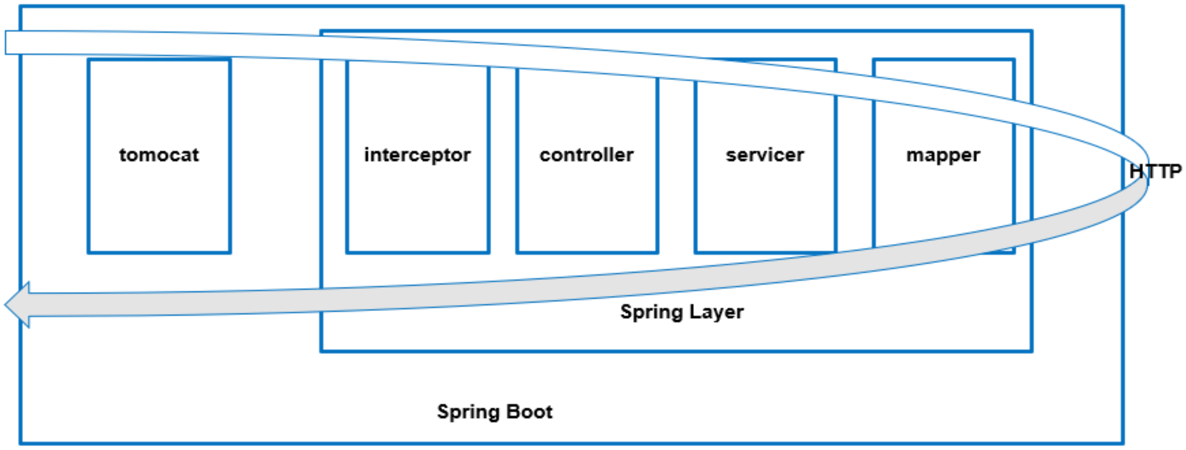


Figure 7 The process of delivering HTTP request

Figure 7 shows the process of delivering HTTP requests in the back end of the web application. During the development of the back end of the web application, the functions of each layer will be implemented.

Components name of Spring Boot	Description
Tomcat	It is a free and open-sourced web application server that is used to deploy this web application
Interceptor	It is employed to intercept user requests and respond appropriately including user authentication, user permission verification, etc.

Controller	It takes the data from the HTTP request, separates the parameters, and routes them to various processing services (service layer), then transmits the processed data (JSON data) back to the front end.
Servicer	It stores business logic processing, avoids using the database directly, and contains interfaces and interface implementation classes with functions that the controller layer can access.
Mapper	It connects the database and allows to operation of the database directly including addition, deletion, update, and other operations.

Table 7 The components of the Spring Boot

3.2.4.3 Development technologies front-end system

Front-end technology	Reason
Vue[13]	Vue is a single-page application, making the page partially refreshed without requiring all data and DOM to be requested each time the page is jumped, speeding up development, and access speed and improving the user experience.
3dsMax[14]	3dsMax can use Vary render toy models so that toys look more vivid.
Visio[11]	Visio is the mainstream software for drawing flowcharts. It will be used to draw use cases, system architecture diagrams, activity diagrams and so forth.

Bootstrap 5[15]	Bootstrap is the most popular CSS framework for developing responsive websites. Bootstrap 5 will be used to design the front end of the website.
Balsmiq Wrieframes[16]	Balsmiq Wrieframes is an excellent tool to design the UI of a web application, which provides many components and icons for users to design.
Photoshop[17]	Photoshop is the most popular photo manipulation software, which can use to crop or create images and icons for the design of the website
Vant 3[18]	Vant is a lightweight and reliable mobile component library. Vant wraps components that Bootstrap does not provide such as address selectors, stepper and so forth.
Iconfont[19]	Iconfont is a powerful and extensive library of vector icons that offer capabilities such as format conversion, online storage, and vector icon download.

Table 8 The front-end technologies used in this website

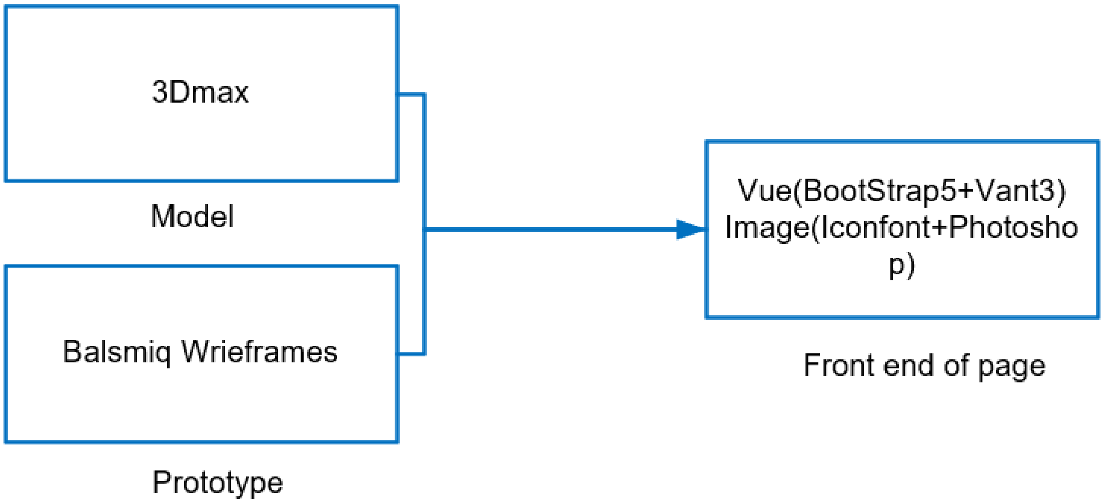


Figure 8 The process of applying technologies in the web application

The process of how to accomplish the front end of the web application will follow in Figure 8. The prototype of the project page is being designed in Balsmiq Wrieframes. The toy model and its accessories are being built by 3dsMax. The front-end pages are being built using frameworks and components provided by Vant 3 and Bootstrap5, with the front-end data sending HTTP requests to the back-end via Axios. For the block of customizing bear development, the model will be exported from 3dsMax and rendered into the page using three.js

3.3.5 Testing Tools

Test Tool	Description
Selenium IDE	Selenium IDE is a plugin for Firefox to help those just starting with automated testing for testing.
Spring Boot Test	Spring Boot provides a dedicated Spring Boot Test code and runs for each layer of the application (interceptor, controller, server, mapper, as shown in Figure 7) to quickly locate bugs and resolve them to ensure the code is robust.

Table 9 Testing tools used in the web application

Spring Boot Test	Supported categories of tests
Unit tests	Generally method-oriented and more expensive to test when writing general business code. The annotations involved are @Test.
Slice-and-dice tests	Generally oriented towards hard-to-test boundary functions, between unit tests and functional tests. The annotations involved are @RunWith @WebMvcTest etc.
Functional testing	Generally oriented towards a complete business function, involving annotations such as @RunWith @SpringBootTest.

Table 10 Supported categories of tests by Spring Boot Test

3.3.6 Testing Types

Testing types	Description
Unit testing	Verifying that the smallest testable component of a web application operates as intended.
Functional testing	Verifying that a web application will work correctly for the target user.
Integration testing	Verifying that the requirements are executed properly, combining the unit test modules one at a time, and examining the combined performance.

Table 11 The list of chosen testing which will be used in the process of testing the web application

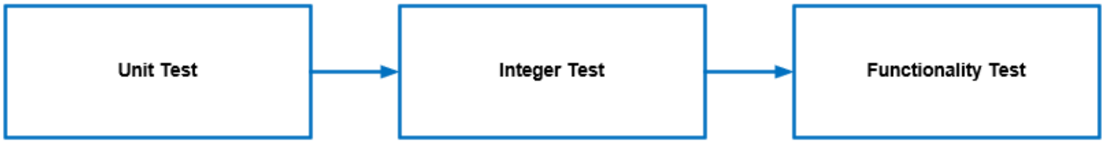


Figure 9 The step of testing the web application

The testing sequence of the web application is following Figure 9 in order to accomplish the above goals. Unit Tests and Integer Tests are more focused on the back end of the web application and the connection between the front and back ends. Functionality Tests are more focused on the usability of features as a test for different users

3.3.7 Test Driven Development

Considering the software development of the web application is Agile, BDD could enhance automated testing capabilities with defined user criteria. To avoid scenarios where all tests pass but the system performs incorrectly, BDD ensures that when modifications are performed due to updates in business requirements, the final product behaves as expected[20]. Therefore, BDD will be applied in the web application.

3.4 Project Version Management

The GitHub platform would be used for the version management of this project to record the progress of the project. Here is the URL of the GitHub account to be used:
https://github.com/Wangyiting1234/project

Chapter 4 Results

The primary goals of testing this web application can be divided into the following points:

1. Improving the software's quality

2. Verifying the software's security

3. Reducing the price of software development

4.1 Deployment Details Environment

This section mainly discusses the deployment details of software, hardware, and testing environment for the web application.

4.1.1 Software and hardware environment

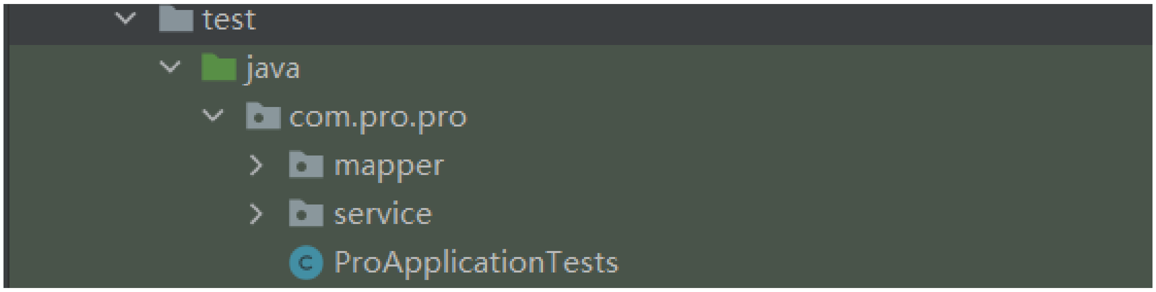
Software Environment	
Server Computer	Operating System: Windows 10 x64 Database: MYSQL 8.0
Client Computer	Operating System: Windows 10 Supporting browser: Chrome Latest, Firefox Latest, Opera Latest, Edge 13 +, IE 11+
Hardware Environment	
Server Computer	CPU: Intel(R) Core (TM) i7-9750H CPU @ 2.60GHz 2.59 GHz Memory: 16.0 GB
Client Computer	CPU: Intel(R) Core (TM) i7-9750H CPU @ 2.60GHz 2.59 GHz Memory: 16.0 GB

Table 12 Deployment of software and hardware environment

4.1.2 Testing environment

4.1.2.1 Spring Boot Test

The introduction and supported categories of tests of Spring Boot Test are shown in Tables 6 and 7. In the process of testing the web application, unit test and integration test will be implemented by Spring Boot Test. The structure of the test folder is constructed, as shown in Figure 10.



```
test
  java
    com.pro.pro
      mapper
      service
      ProApplicationTests
```

Figure 10 Structure of test folder of Spring Boot Test

Package Name	Function
com.pro.pro	Manage all test files
mapper	Manage all test classes of the mapper layer
service	Manage all test classes of the service layer

Table 13 Function of each package name in the test folder

4.1.2.2 Selenium

The introduction of selenium is shown in Table 6. In the process of testing the web application, Selenium IDE will be downloaded, inserted into Firefox, and used to test each link whether can jump to the correct page.

4.2 Unit Tests

This section displays the process and results of unit tests in the mapper, servicer, controller, and interceptor layers of the web application. Considering the priority of functions, the process of testing the block of User and Address will be displayed.

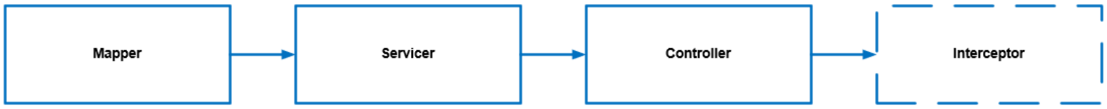


Figure 11 Test sequence of unit test

Considering the function and logic of each layer, the back-end unit tests will be divided into four sections and tested accordingly.

Test Section	Purpose
Mapper	Test that the mapper interface correctly corresponds to the corresponding configuration file and that the interface implementation class can perform the corresponding operations on the database
Servicer	Test whether the service's interface's implementation class complies with the required business logic and whether the relevant exception can be captured
Controller	Test that the request is forwarded, that the parameters of the accepted page are accurately passed to the service for processing, and that the service layer interface is appropriately called to manage the business process and receive the right return response from the service.
Interceptor	Test whether the interceptor can intercept user requests and process them accordingly

Table 14 Test the purpose of each layer in the back end of this web application

4.2.1 Unit Test of Mapper Layer

Unit Test List of Mapper Layer

Id	Package Name	Class Name	Method Name	Priority	Description
1	com/pro/pro/mapper	UserMapper.java	Insert()	medium	Insert data from the user's telephone and password into the database
2	com/pro/pro/mapper	UserMapper.java	findByUserTel()	medium	Find user information by user telephone number
3	com/pro/pro/mapper	UserMapper.java	findByUserId()	medium	Find user information by user telephone number
4	com/pro/pro/mapper	UserMapper.java	updatePasswordById()	low	Update user password by user Id
5	com/pro/pro/mapper	UserMapper.java	updateUserInfo()	low	Update user gender, and user name by user Id

Table 15 Unit Test List of Mapper Layer

4.2.1.1 Unit Test Case and Result of User Mapper Layer

<table>
<tr><th colspan="6">Unit Test Case</th></tr>
<tr><td><b>Test Case Id</b></td><td colspan="5">1</td></tr>
<tr><td><b>Test Case Purpose</b></td><td colspan="5">Test data of user telephone and password whether could be inserted into database accurately and correspondingly</td></tr>
<tr><td><b>Input Action</b></td><td>User telephone: "15604781380"
User password:”123123”</td><td><b>Expected Output/Actions</b></td><td>Data inserted into the database accurately and correspondingly</td><td><b>Actual Output/ Actions</b></td><td>Success (See Figure 30)</td></tr>
</table>

Table 16 Unit test case 1 in user mapper layer

<table>
<tr><th colspan="6">Unit Test Case</th></tr>
<tr><td><b>Test Case Id</b></td><td colspan="5">2</td></tr>
<tr><td><b>Test Case Purpose</b></td><td colspan="5">Test user information whether could be found by user telephone number and return list of user information correctly</td></tr>
<tr><td><b>Input Action</b></td><td>User telephone: "15604781380"</td><td><b>Expected Output/Actions</b></td><td>User information return to the consoler accurately</td><td><b>Actual Output/Actions</b></td><td>Success (See Figure 31)</td></tr>
</table>

	User password:”123123”		and correspondingly		

Table 17 Unit test case 2 in user mapper layer

Unit Test Case					
Test Case Id	3				
Test Case Purpose	test user information whether could be found by the user's telephone number and return a list of user information correctly				
Input Action	User telephone: "15604781380" User password:”123123”	**Expected Output/Actions**	User information return to the consoler accurately and correspondingly	**Actual Output/Actions**	Success (See Figure 32)

Table 18 Unit test case 3 in user mapper layer

Unit Test Case					
Test Case Id	4				
Test Case Purpose	test user information whether could be found by the user Id and return a list of user information correctly				
Input Action	User Id: 18	**Expected Output/Actions**	User information	**Actual Output/Actions**	Success (See

			return to the consoler accurately and correspondingly		Figure 33)

Table 19 Unit test case 4 in user mapper layer

Unit Test Case					
Test Case Id	5				
Test Case Purpose	Test the user password whether could be updated by the user Id				
Input Action	User Id: 18 User Current Password:"123123" User Updated Password:"123456	**Expected Output/Actions**	User password could be updated from current password to updated password by user Id accurately	**Actual Output/Actions**	Success (See Figure 34)

Table 20 Unit test case 5 in user mapper layer

Unit Test Case

Test Case Id	6				
Test Case Purpose	Test user gender and name whether could be updated by user Id				
Input Action	User Id: 18 User Gender:1 User Name ” Elsa”	**Expected Output/Actions**	User gender and name could be updated by user Id accurately	**Actual Output/Actions**	Success (See Figure 35)

Table 21 Unit test case 6 in user mapper layer

4.2.2 Unit Test of User Servicer Layer

Unit Test List of Servicer Layer					
Id	**Package Name**	**Class Name**	**Method Name**	**Priority**	**Description**
1	com/pro/pro/service/impl	UserServiceImpl.java	reg ()	heigh	Insert the user's telephone number and encrypted password into the database
2	com/pro/pro/service/impl	UserServiceImpl.java	login ()	heigh	Check the user’s telephone and match

					the telephone number with the corresponding password
3	com/pro/pro/service/impl	UserServiceImpl.java	ChangePassord()	medium	Check user's current password is correct and modify the user's password by user Id
4	com/pro/pro/service/impl	UserServiceImpl.java	getByUserId()	low	Check user id whether exists and return the user gender and name
5	com/pro/pro/service/impl	UserServiceImpl.java	ChangeInfo()	low	Update user gender, and user name by user Id

Table 22 Unit test list in servicer layer

4.2.2.1 Unit Test Case and Result of Servicer Layer

Unit Test Case

Test Case Id	1	
Test Case Purpose	1. Test user telephone number and password whether could be inserted into data successfully 2. Test password whether could be encrypted 3. Test if an existing phone number can be re-inserted into the database	
Input Action	**Expected Output/Actions**	**Actual Output/Actions**
User Telephone Number:”15604781212” User Password:”121212”	User data inserts database successfully	Success (See Figure 36)
User Telephone Number:” 15604781380” User Password:”123123”	User data inserts database failingly	Success (See Figure 37)

Table 23 Unit test case 1 in user servicer layer

Unit Test Case		
Test Case Id	2	
Test Case Purpose	1. Test if an unregistered user telephone number can be found 2. Test if an errored password can be matched with the telephone 3. Test if an encrypted password can be decrypted correctly	
Input Action	**Expected Output/Actions**	**Actual Output/Actions**
User Telephone Number:”15604781212” User Password:”121212”	The user’s telephone number and password match successfully	Success (See Figure 38)

User Telephone Number:” 15604781212” User Password:”123123”	The user‘s telephone number and password match failingly and throw an exception	Success (See Figure 39)
User Telephone Number:” 15604785555” User Password:”123123”	The user's telephone number can be not found and throw an exception	Success (See Figure 40)

Table 24 Unit test case 2 in user servicer layer

Unit Test Case		
Test Case Id	3	
Test Case Purpose	1. Test if an unregistered user modify password can throw an exception 2. Test if the current password can be not matched with the input password 3. Test new password can be encrypted 4. Test new password can replace the current password	
Input Action	**Expected Output/Actions**	**Actual Output/Actions**
User Telephone Number:”15604781212” User Current Password:”121212” User New Password:”123123”	The user's current password can be replaced with the new password	Success (See Figure 41)
User Telephone Number:” 15604781212” User Current Password:”123123”	The user's current password is not matched with password from the database	Success (See Figure 42)

User New Password:”121212”		
User Telephone Number:” 15604785555” User Current Password:”123123” User New Password:”121212”	The user’s telephone number can be not found and throw an exception	Success (See Figure 43)

Table 25 Unit test case 3 in user servicer layer

Unit Test Case		
Test Case Id	4	
Test Case Purpose	1. Test if an existing user id can be found and return the user’s gender and name 2. Test if a non-existent user can be not found and throw an exception	
Input Action	**Expected Output/Actions**	**Actual Output/Actions**
User Id:10	The user id can be found and return user gender and name	Success (See Figure 44)
User Id:130	The user id can be not found and	Success (See Figure 45)

Table 26 Unit test case 4 in user servicer layer

Unit Test Case	
Test Case Id	5
Test Case Purpose	1. Test if an existing user id can be found and update the user’s gender and name

	2. Test if a non-existent user can be not found and throw an exception	
Input Action	**Expected Output/Actions**	**Actual Output/Actions**
User Id:8 User Gender:0(female) User Name: ”Alie”	The user id can be found and the user gender and name can be updated	Success (See Figure 46)
User Id:130	The user id can be not found	Success (See Figure 47)

Table 27 Unit test case 5 in user servicer layer

With the above unit tests, the logic associated with each unit and the operation of the database is the same as the desired result.

4.3 Integration Tests

Integration tests will be carried out below to confirm that the interfaces between the units connect and operate on the database properly based on the outcomes of the unit tests carried out above.

4.3.1 Integration Tests of User

Integration Test List of User					
Id	**Package Name**	**Class Name**	**Method Name**	**Priority**	**Description**
1	com/pro/pro/controller	UserController.java	signin()	heigh	Set the router for registering and delivering the user's telephone and password to the

					Server layer
2	com/pro/pro/controller	UserController.java	login()	heigh	Set the router to log in and deliver the user’s telephone and password to the Server layer
3	com/pro/pro/controller	UserController.java	ChangePassord()	medium	Set the router of changing the password and deliver the user telephone and password to the Server layer
4	com/pro/pro/controller	UserController.java	getByUserId()	low	Set the router of receiving user gender and name by

					user id and deliver the user id to the Server layer
5	com/pro/pro/controller	UserController.java	ChangeInfo()	low	Set the router of updating the user gender and name and deliver the user id, gender, and name to the Server layer

Table 28 Integration Test List in User

Integration Test Case			
Test Case Id	1		
Test Case Purpose	1. Test the user telephone number and password whether could be inserted into data successfully by the HTTP request 2. Test if an existing phone number can be re-inserted into the database by the HTTP request		
Input Action	**Expected State**	**Expected Message**	**Actual Output/Actions**

User Telephone Number:”15604781213” User Password:”121212”	200	User data inserts database successfully	Success (See Figure 48)
User Telephone Number:” 15604781212” User Password:”123123”	4000	The telephone number has been registered	Success (See Figure 49)

Table 29 Integration test case 1 in User

Integration Test Case			
Test Case Id	2		
Test Case Purpose	1. Test if an unregistered user telephone number can be found by the HTTP request 2. Test if an errored password can be matched with telephone by the HTTP request		
Input Action	**Expected State**	**Expected Message**	**Actual Output/Actions**
User Telephone Number:”15604781213” User Password:”121212”	200	Requested data of the user match with user data on the database	Success (See Figure 50)
User Telephone Number:” 15604781213” User Password:”123123”	5002	Password is wrong	Success (See Figure 51)
User Telephone Number:” 15604785555” User Password:”123123”	5001	The telephone number has not been registered	Success (See Figure 52)

Table 30 Integration test case 2 in User

<table>
<tr><th colspan="5">Integration Test Case</th></tr>
<tr><td colspan="2"><b>Test Case Id</b></td><td colspan="3">3</td></tr>
<tr><td colspan="2"><b>Test Case Purpose</b></td><td colspan="3">1. Test if an unregistered user modify password can return an error message by HTTP request
2. The test back end will return an error message if the current password can be not matched with the input password by the HTTP request
3. The test back end will return success state that the new password can replace the current password by the HTTP request</td></tr>
<tr><td><b>Previous Action</b></td><td><b>Input Action</b></td><td><b>Excepted State</b></td><td><b>Expected Message</b></td><td><b>Actual Output/Actions</b></td></tr>
<tr><td>Log in</td><td>User Telephone Number:”15604781212”
User Current Password:”121212”
User New Password:”123123”</td><td><b>200</b></td><td><b>Requested data of the user match with user data on the database</b></td><td>Success (See Figure 53)</td></tr>
<tr><td>Log in</td><td>User Telephone Number:” 15604781212”
User Current Password:”123123”
User New Password:”121212”</td><td><b>5002</b></td><td><b>Password is wrong</b></td><td>Success (See Figure 54)</td></tr>
<tr><td>Not Log in</td><td>User Telephone Number:” 15604781212”
User Current Password:”123123”</td><td><b>5003</b></td><td><b>The user has not logged in</b></td><td>Success (See Figure 55)</td></tr>
</table>

	User New Password:”121212”			

Table 31 Integration test case 3 in User

Integration Test Case			
Test Case Id	4		
Test Case Purpose	1. The test back end will return a successful state if an existing user id can be found and return the user gender and name		
Input Action	**Excepted State**	**Expected Message**	**Actual Output/Actions**
User Id:10	200	Requested data of the user match with user data on the database	Success (See Figure 56)

Table 32 Integration test case 4 in User

Integration Test Case				
Test Case Id	5			
Test Case Purpose	1. The test back end will return success state if an existing user id can be found and update the user gender and name 2. The test back end will return an error message if the user id has not been stored in session before updating the user gender and name by HTTP request			
Previous Action	**Input Action**	**Excepted State**	**Expected Message**	**Actual Output/Actions**
Log in	User Id:8 User Gender:0(female)	User id can be found and user gender and name can be updated	State code:200	Success (See Figure 57)

	User Name:" Alie"			
Not Log in	Log in	User Id:8 User Gender:0(female) User Name:" Alie"	The user has not logged in	Success (See Figure 58)

Table 33 Integration test case 5 in User

Each unit is precisely matched to the called interface through the integration tests mentioned above, and the back end could implement the desired classes and perform the corresponding operations on the requested data in HTTP.

4.4 Functionality Tests

Considering the website is an online toy brand, registering, logging in, adding new addresses and customizing toys are primary functions to test.

4.4.1 Registration Test

Functionality Test Case				
Test Case Name	Registration			
Test Case Purpose	1. Test whether the new user can successfully register 2. Test whether existing users can successfully register 3. Test whether the front end can validate inputting data 4. Test whether corresponding data from the back end can be delivered to the front end			
Id	**Test Items**	**Procedure**	**Input Data**	**Excepted Output/Actions**

1	**Interface Test**	1. Check the title of the page 2. Check the link to the page 3. Check whether the tab works correctly 4. Check whether the input bar focus correctly 5. Check whether the link of log in displays 6. Check whether the navigation displays at the top of the interface	Register URL: http://localhost:9090/#/signin	1. Title: Sign up 2. Link page: Page of sign up 3. Press the tab can submit the form 4. Click the input bar can focus on excepted bar 5. “already sign up?” shows on this page 6. Navigation is shown on this page
2	Telephone Validation Test	1. Input telephone number	Empty telephone number:”	“Please check the format of the telephone number”
			Less than 11 bits of telephone	“Please check the format of the telephone number”

			number:”1560478133”	
			More than 11 bits of telephone number:”156047813321”	“Please check the format of the telephone number”
			Wrong telephone number segment:”12345678911”	“Please check the format of the telephone number”
			Illegal symbols: “1560478111。”	“Please check the format of the telephone number”
			The correct format of the telephone number:”15604781111”	Success (See Figure 60)
3	Password Validation Test	1. Input the password	Empty password:”	“Password cannot be empty”
			Less than 6 bits of	“Please check the format of the password”

	2. Input confirmed password	password:" 12312"	
		More than 6 bits of password but the first password and confirmed password are not the same: first password: " 123123" and confirmed password:" 123122"	"Confirm passwords are the same"
		More than 6 bits of password and first password and confirmed password are same: first Password:" 123123" Confirmed	Success (See Figure 60)

			password:” 1231233”	
4	Request Test	Submit the form	Legal telephone number:”15604781999” Password:” 123123” Confirmed password:” 1231233”	Success (See Figure 61) Response: {"state":200,"message":null,"data":null}

Table 34 Functionality test of registering

Interface Name	Title of Page	Link of Page	Tab Works Correctly	Input bar focus correctly	“aleardy sign up” shows	Navigation Displays On The Top Of Page
Sign Up	Sign up	http://localhost:9090/#/signin	Success (See Figure 59)	Success (See Figure 59)	Success (See Figure 59)	Success (See Figure 59)

Table 35 The actual result of the user interface of registering

4.4.2 Log in Test

Functionality Test Case

Test Case Name	Log in			
Test Case Purpose	1. Test whether the new user can successfully log in 2. Test whether the existing user can successfully login 3. Test whether the front end can validate inputting data 4. Test whether corresponding data from the back end can be delivered to the front end			
Id	**Test Items**	**Procedure**	**Input Data**	**Excepted Output/Actions**
1	Interface Test	1. Check the title of the page 2. Check the link to the page 3. Check whether the tab works correctly 4. Check whether the input bar focus correctly 5. Check whether the link of the register displays	Register URL: http://localhost:9090/#/Log	1. Title: Login 2. Link page: Page of log in 3. Press the tab can submit the form 4. Click the input bar can focus on excepted bar 5. "has not registered?" shows on this page 6. Navigation is shown on this page

		6. Check whether the navigation displays at the top of the interface		
2	Telephone Validation Test	1. Input telephone number	Empty telephone number:””	“Please check the format of the telephone number”
			Less than 11 bits of telephone number:”1560478133”	“Please check the format of the telephone number”
			More than 11 bits of telephone number:”156047813321”	“Please check the format of the telephone number”
			Wrong telephone number segment:”12345678911”	“Please check the format of the telephone number”
			Illegal symbols:”1560478111.”	“Please check the format of the telephone number”

			The correct format of the telephone number:”15604781111”	Success (See Figure 63)
3	Password Validation Test	1. Input the password 2. Input confirmed password	Empty password	“Password cannot be empty”
			Less than 6 bits of password	“Please check the format of the password”
			The correct format of the password	Success (See Figure 63)
4	Request Test	Submit the form	Legal telephone number:”15604781999” Password:” 123123”	Success
5.	Data of Back End Test	1. Input an unregistered telephone number to log in 2. Submit the form	Telephone number:”” Password:” 123123”	Success (See Figure 64) Response: "state":5001,"message":"telephone number has not been register","data":null}

		1. Input error password 2. Submit the form	Telephone number:”” Password:” ”	Success (See Figure 65) Response: {"state":5002,"message":"password is wrong","data":null}
		1. Input registered telephone number 2. Input correct password 3. Submit the form		Success (See Figure 66) Response: {"state":200,"message":null,"data":{"createTime":null,"createUser":null,"updateTime":null,"updateUser":null,"user_id":22,"userName":null,"userPwd":"80554EEBFCD44F93AA76D01AA61181F2","userGender":null,"userTel":"15604781999","userSalt":null,"isDelete":null}}

Table 36 Functionality test of logging in

Interface Name	Title of Page	Link of Page	Tab Works Correctly	Input bar focus correctly	“has not registered” shows	Navigation Displays On The Top Of Page
Log in	Log in	http://localhost:9090/#/Log	Success (See Figure 62)	Success (See Figure 62)	Success (See Figure 62)	Success (See Figure 62)

Table 37 The actual result of the user interface of logging in

4.4.3 Add new delivery address test

<table>
<tr><th colspan="5">Functionality Test Case</th></tr>
<tr><td><b>Test Case Name</b></td><td colspan="4">Add new delivery address</td></tr>
<tr><td><b>Test Case Purpose</b></td><td colspan="4">1. Test whether unregistered user can successfully add new address
2. Test whether registered user can successfully add new address
3. Test whether front end can validate inputting data
4. Test whether corresponding data from front end can be delivered to the back end and stored on the database</td></tr>
<tr><td><b>Id</b></td><td><b>Test Items</b></td><td><b>Procedure</b></td><td><b>Input Data/Actions</b></td><td><b>Excepted Output/Actions</b></td></tr>
</table>

1	Interface Test	1.Check the title of page 2.Check the link of page 3. Check whether the tab works correctly 4.Check whether the input bar focus correctly 5. Check whether the required inputting bar displays 6. Check whether the select bar of choosing province, city and area displays and can be selected 7. Check whether the radio of setting default address displays and can be set 8. Check whether the navigation displays in the top of interface	Register URL: http://localhost:9090/#/address/addaddress	1. Title: add new address 2. Link page: Page of adding new address 3. Press tab can submit form 4. Click input bar can focus on excepted bar 5. Required input bar shows on this page (name, telephone, detailed address) 6. Select bar of province, city and area shows on this page 7. Radio of setting as default address shows on this page 8. Navigation shows on this page

2	Name Validation Test	1. Input receiver name	Empty name:””	“Please fill in this name”
			Correct format of name	Success (See Figure 68)
3	Telephone Validation Test	1. Input telephone number	Empty telephone number:””	“Malformed phone number”
			Less than 11 bits of telephone number:”1560478133”	“Malformed phone number”
			More than 11 bits of telephone number:”156047813321”	“Malformed phone number”
			The correct format of the telephone number:”15604781111”	Success (See Figure 68)
4	Selected bar Test	1. Click “Area” 2. Select Province 3. Select City 4. Select Area	Click “Area”	Select window pops up
			Select province: “Beijing”	The corresponding city (Beijing City) will be shown
			Select city: “Beijing City”	The corresponding area (Dongcheng District, Xicheng

		5. Click “Confirm”		District) will be shown
			Click “Confirm”	Success (See Figure 68)
		6. Click “Cancel”	Click “Cancel”	Success (See Figure 68)
5	Detailed Address Test	Input detailed address	Empty detailed address	“Address cannot be empty”
			The correct format of the detailed address	Success (See Figure 68)
6	Default address Test	Click radio to set it as the default address	None	Success (See Figure 68)
		Click the radio to cancel	None	Success (See Figure 68)
7	Request Test	Submit the form for the new address	Legal inputting data	Success

8	Data of Back End Test	1. Inputting required information of address 2. Submit the form	Legal inputting data	Success (See Figure 69) Response: {"state":200,"message":null,"data":null}
9	Data of Back End Test	1. Inputting required information of address 2. Set address as the default adddress 2. Submit the form	Legal inputting data	Success (See Figure 70) Response: {"state":200,"message":null,"data":null}

Table 38 Functionality test of adding a new address

Interface Name	Title of Page	Link of Page	Tab Works Correctly	Input bar focus correctly	Required inputting bar displays	Navigation Displays On the Top Of Page	Select bar for province, city and area displays	Radio for setting as default address displays

ADD NEW ADDRESS	ADD NEW ADDRESS	**http://localhost:9090/#/address/addaddress**	Success (See Figure 67)	Success (See Figure 67)	Success (See Figure 67)	Success (See Figure 67)	Success (See Figure 67)	Success (See Figure 67)

Table 39 Interface Test of adding a new address

The above functional tests are divided into two categories: normative validation of the form format and validation of the request sent by the front end. In the case of form format validation, the front end can provide the user with the correct data validation results. In the HTTP request test, the back-end correctly receives the request and the data carried by the front-end, as well as the data returned by the back-end, and provides corresponding feedback to the user.

4.5 Graphical user interfaces of the website

The graphic user interfaces are displayed in this section and the sequence of showing graphical user interfaces will follow the user who is the first time to browse this website and make an order.

4.5.1 Graphical User Interfaces of Homepage

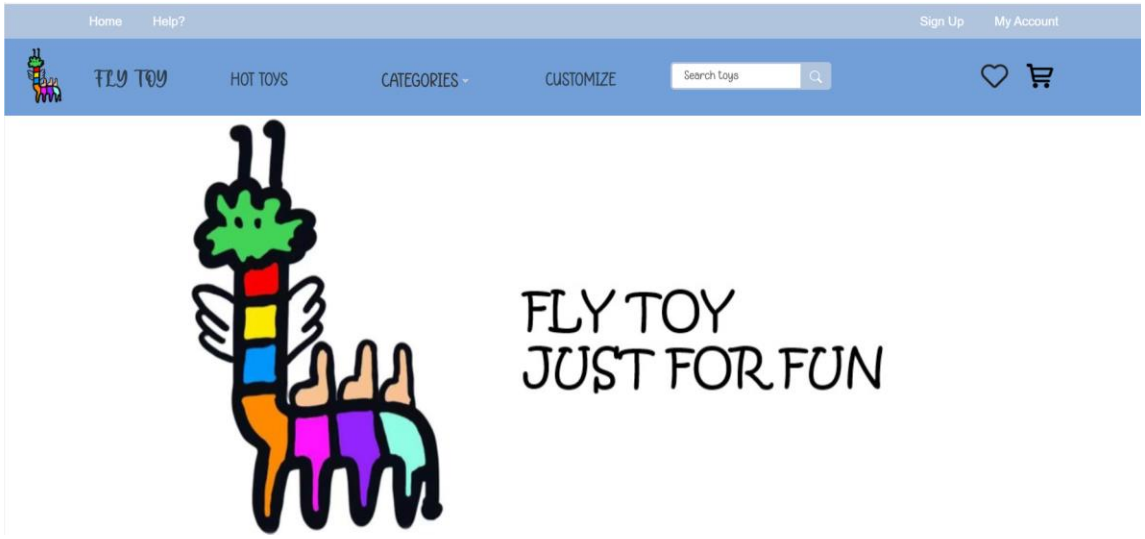
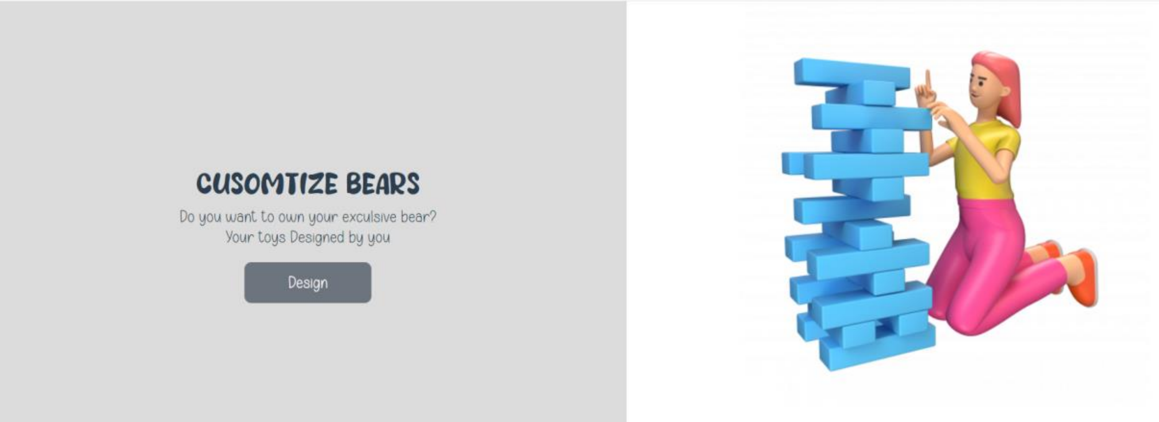


Figure 12 The top of the homepage

Two navigation bars will be available at the top of the home page to guarantee that basic demands are able to be satisfied by customers efficiently and simply. In addition, a scrolling slide show features the toy brands.



CUSOMTIZE BEARS

Do you want to own your exculsive bear?
Your toys Designed by you

Figure 13 The upper middle of the homepage

The upper middle of the homepage will be used to attract consumers to customize the bear through a tagline.

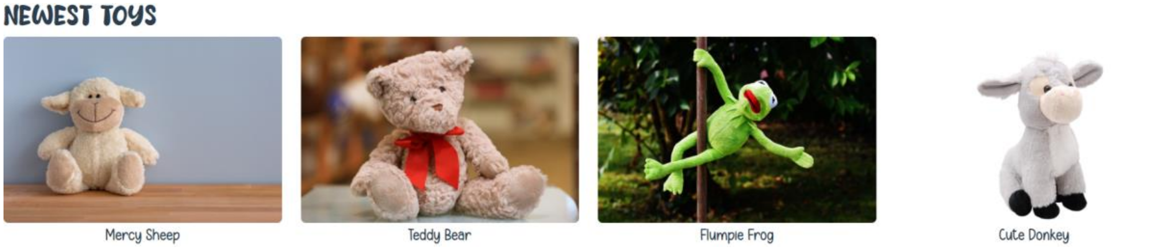
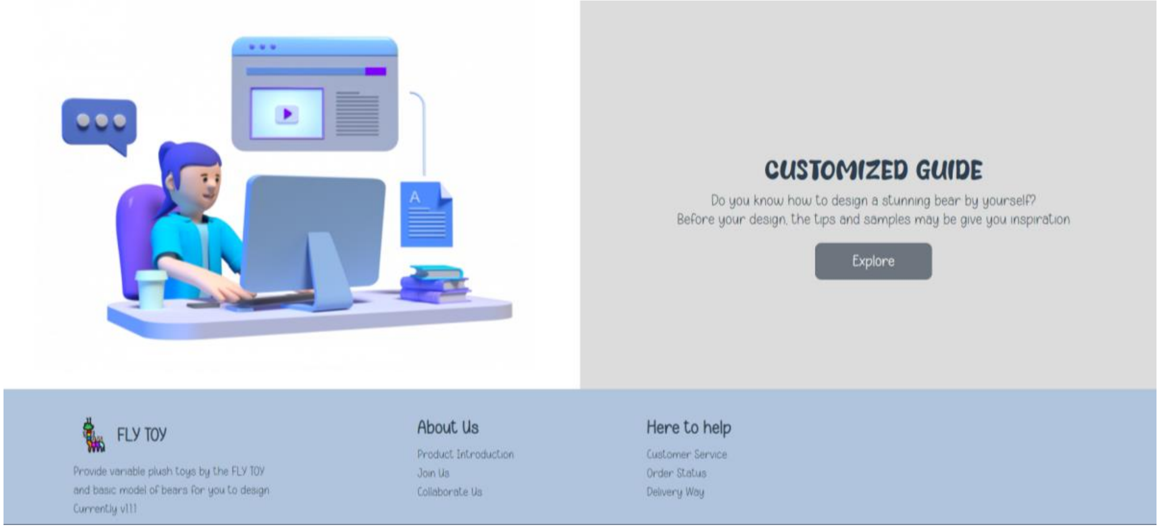


Figure 14 The lower middle of the homepage

The latest four toys will be pushed to the lower middle of the homepage to keep users up to date with the brand's new launches.


CUSTOMIZED GUIDE

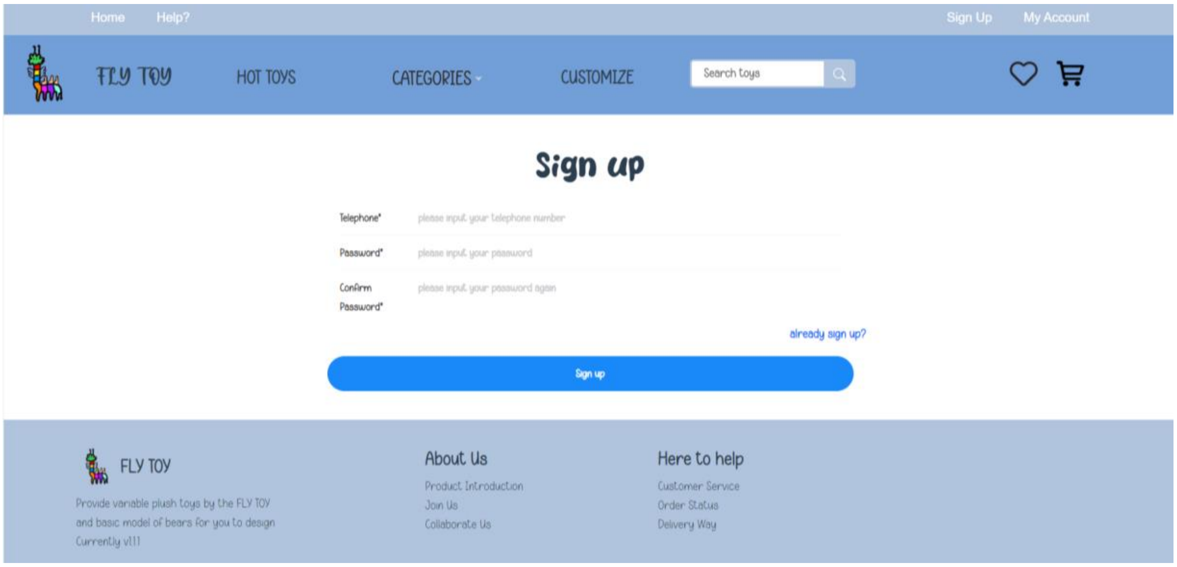
Do you know how to design a stunning bear by yourself?
Before your design, the tips and samples may be give you inspiration

Provide variable plush toys by the FLY TOY
and basic model of bears for you to design
Currently v1.1

Figure 15 The bottom of the homepage

The introductory remarks of a customized guide are provided for customers to explore how to customize a bear and receive inspiration. Users will discover services at the bottom navigation that might be of interest to them, as well as brand-related presentations, information, contact information, and culture.

4.5.2 Graphical User Interfaces of Register and Log-in System



Telephone* please input your telephone number

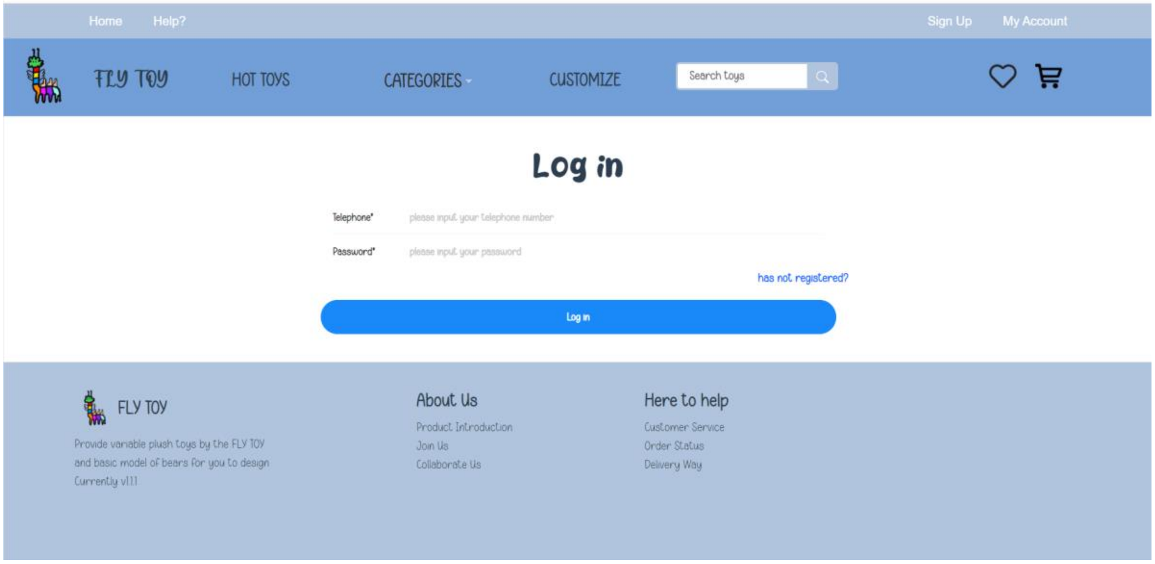
Password* please input your password

Confirm Password* please input your password again

Provide variable plush toys by the FLY TOY
and basic model of bears for you to design
Currently v1.1

Figure 16 The Graphical User Interface of Registration

The Graphical User Interface of Registration provides the necessary input bar and the placeholder for users to input precise information about registering.

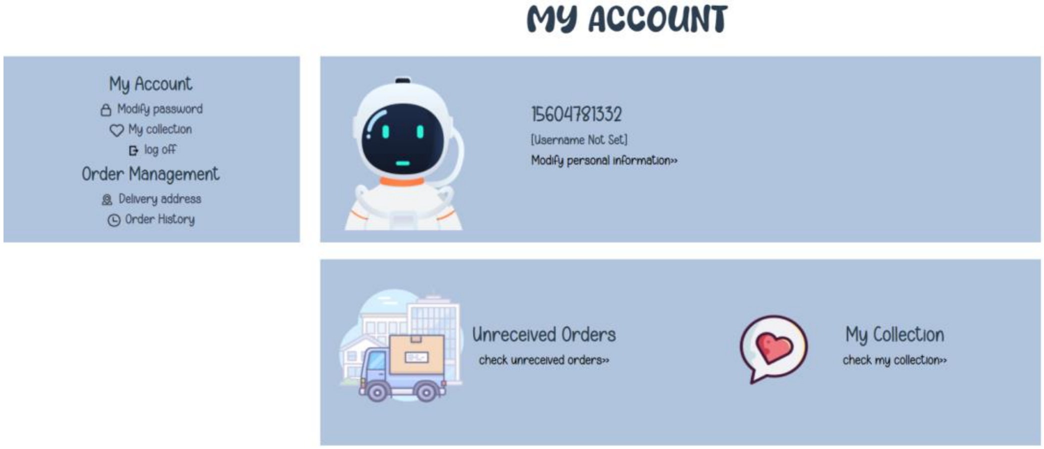


Provide variable plush toys by the FLY TOY and basic model of bears for you to design
Currently v1.1

Figure 17 The Graphical User Interface of log in

The Graphical User Interface for logging in provides the necessary input bar and the placeholder for users to input precise information about logging in.

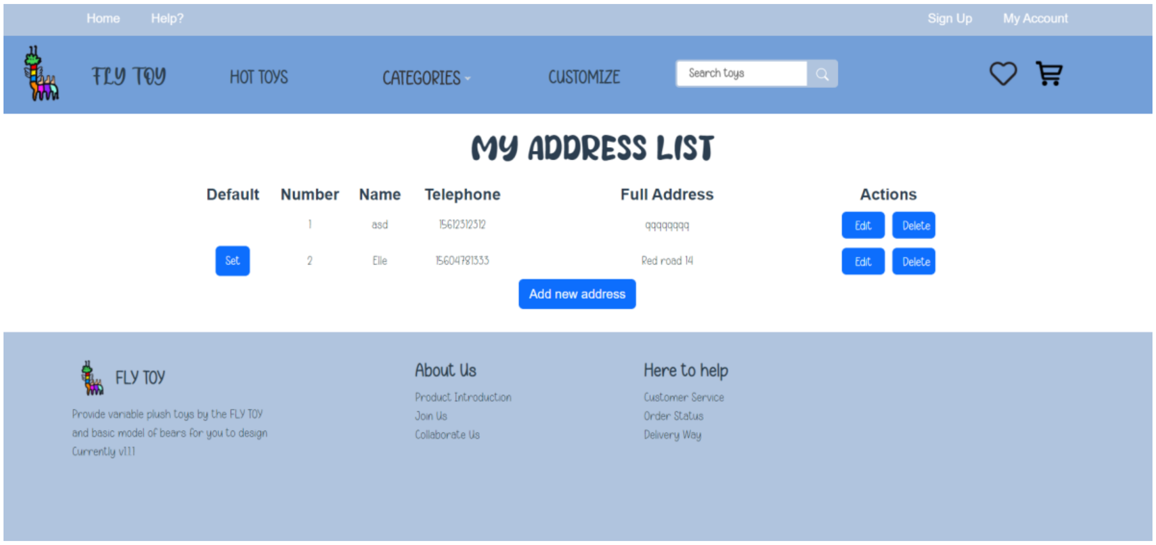
4.5.3 Graphical User Interfaces of User Account System



15604781332
[Username Not Set]
Modify personal information»

Figure 18 The Graphical User Interface of a user account after logging in

User accounts provide users with personal information, addresses, orders and other related functions.



MY ADDRESS LIST

Default	Number	Name	Telephone	Full Address	Actions
	1	asd	15612312312	qqqqqqq	Edit Delete
Set	2	Elle	15604781333	Red road 14	Edit Delete

FLY TOY

Provide variable plush toys by the FLY TOY and basic model of bears for you to design

Currently v1.1.1

Figure 19 The Graphical User Interface of the address list

The address list is provided for users to modify related information of their addresses after the user clicks the “delivery address” button.

4.5.4 Graphical User Interfaces of Hot Toys

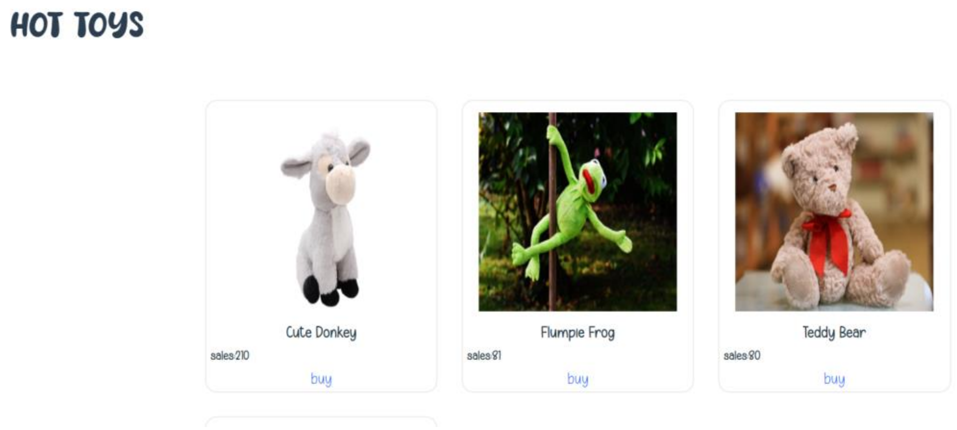
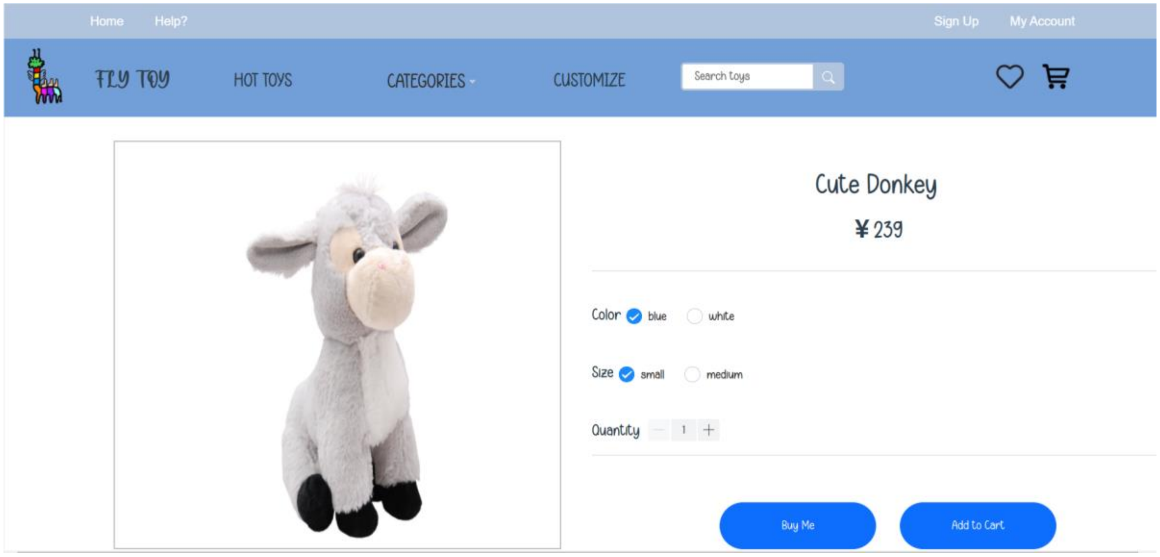


Figure 20 The Graphical User Interfaces of hot toys page

Toys are presented to customers on the Hot Toys page in descending order of sales volume. The page will be shown the name, picture, and sales volume of each popular toy.



Color ☑ blue ○ white

Size ☑ small ○ medium

Quantity − 1 +

Figure 21 The Graphical User Interface of the product details

The toy details page will provide the user with specific parameters for the selected toy, as well as the number of options available. At the same time, the user options to buy directly or add to the shopping cart. In addition, the parameters corresponding to the different sizes of the toy and the physical details will be available for the user to view.

4.5.5 Graphical User Interfaces of Customizing a Bear

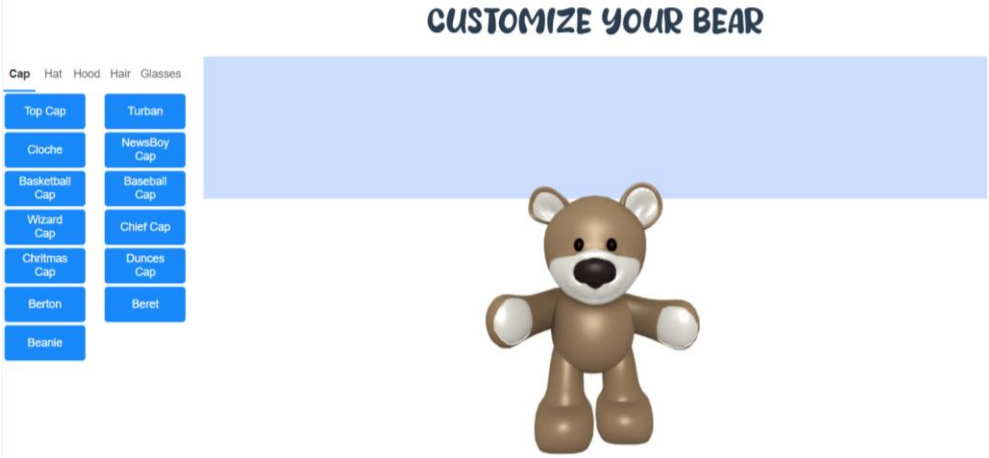


Figure 22 The initialized Graphical User Interface for customizing a bear

The base mode of the bear and various subcategories (cap, hat, hood, hair, glasses) are buttons for the bear's accessories that are displayed to the user whenever users jump to the Custom Bears page.

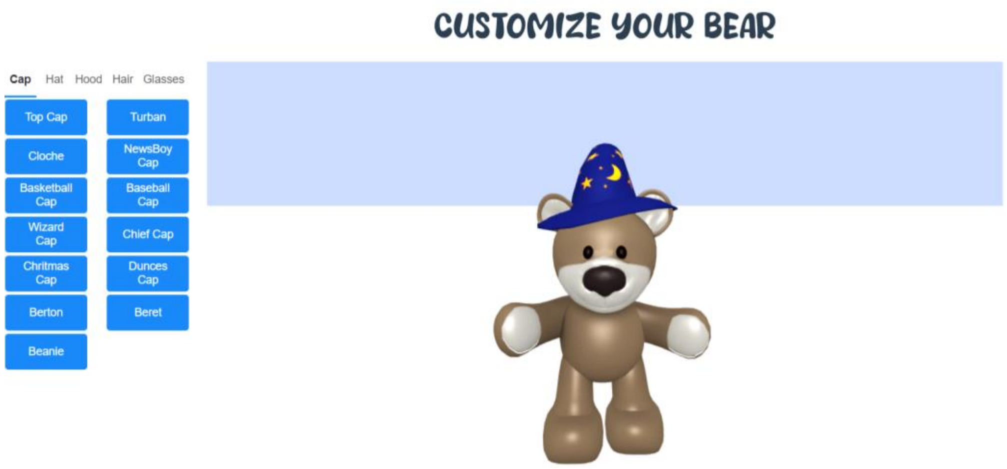


Figure 23 The Graphical User Interface after clicking the “Wizard Cap” button

The cap will be decorated on the bear's head while the user clicks on the " Wizard Cap " under the “Cap” category

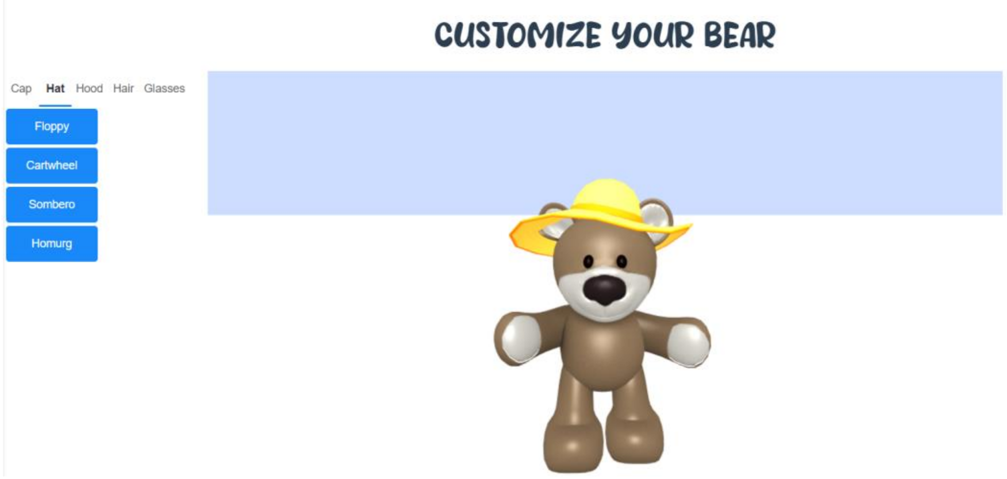


Figure 24 The Graphical User Interface after clicking the “Cartwheel” button

The Cartwheel will be decorated on the bear's head while the user clicks on the " Cartwheel " under the “Hat” category

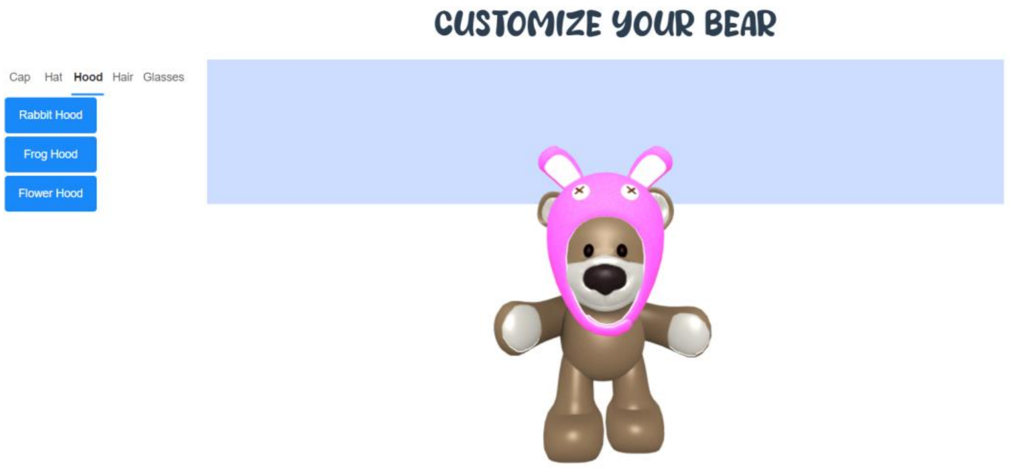


Figure 25 The Graphical User Interface after clicking the “Rabbit Hood” button

The rabbit hood will be decorated on the bear's head while the user clicks on the " Rabbit Hood" under the “Hood” category

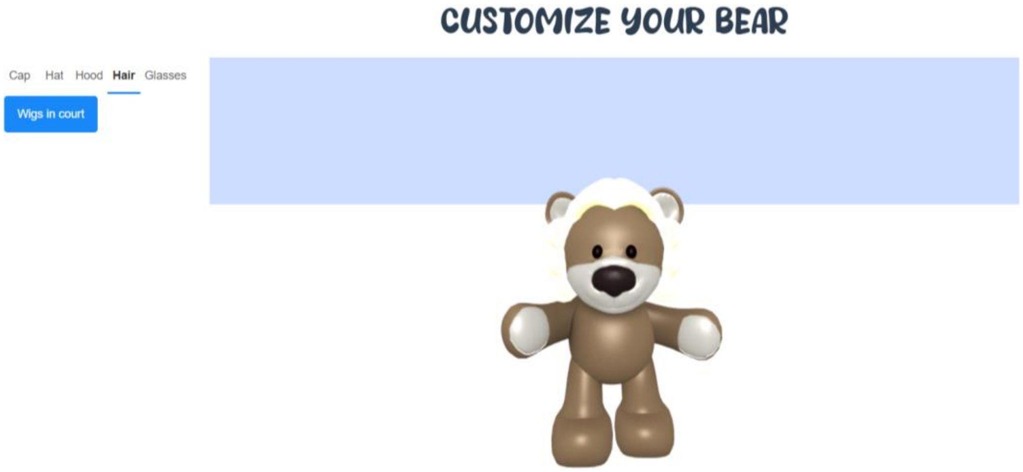


Figure 26 The Graphical User Interface after clicking the “Wigs in court” button

The hair will be decorated on the bear's head while the user clicks on the " Wigs in court " under the “Hair” category

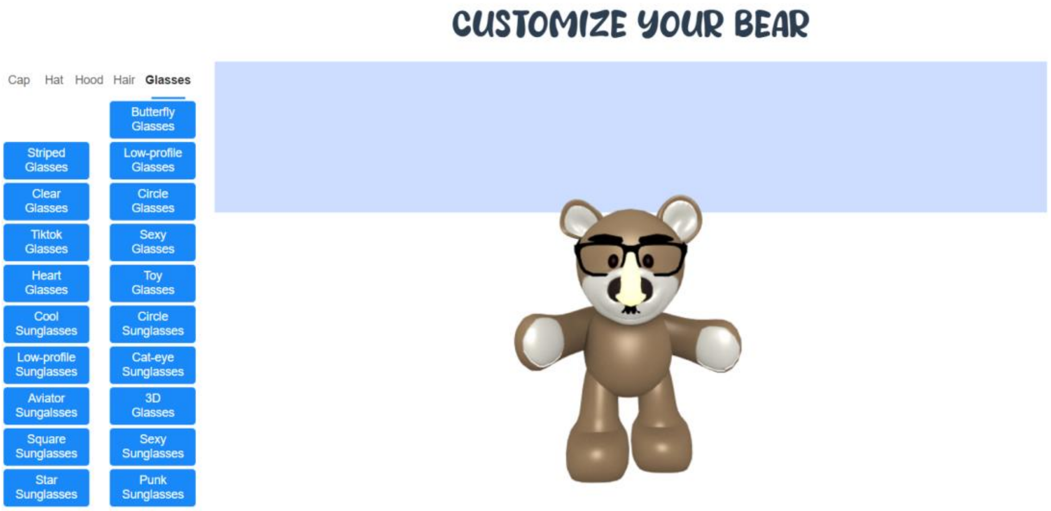


Figure 27 The Graphical User Interface after clicking the “Toy Glasses” button

The glasses will be decorated on the bear's eyes while the user clicks on the " Toy Glasses " under the “Glasses” category

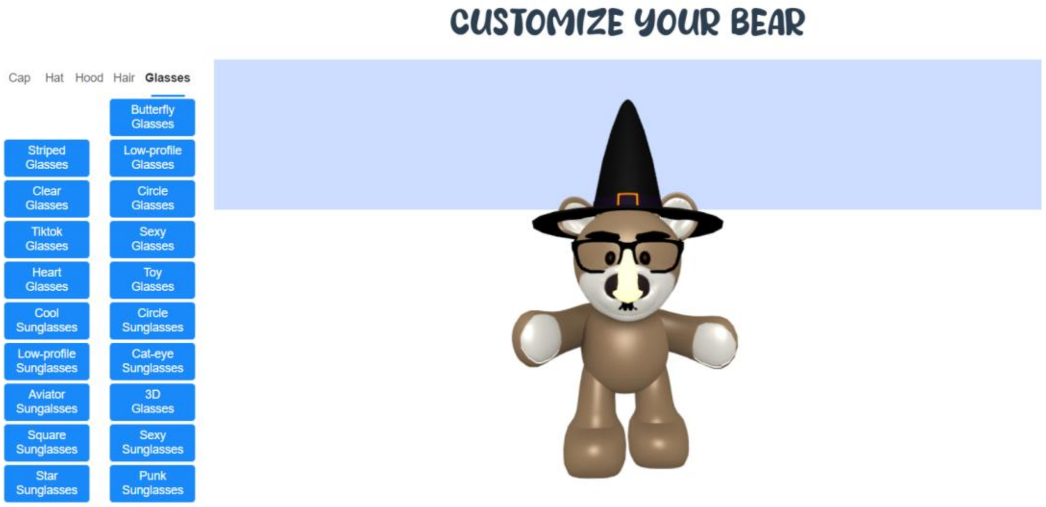


Figure 28 The Graphical User Interface after clicking the “Toy Glasses” button and the “Dunces Cap” button

The glasses and cap can be decorated together while the user clicks on the " Toy Glasses " under the “Glasses” category and the “Dunces Cap” under the “Cap” category

Chapter 5 Professional Issues

5.1 Project Management

This section will discuss the planning of project management and the degree of completion of the project.

5.1.1 Activities

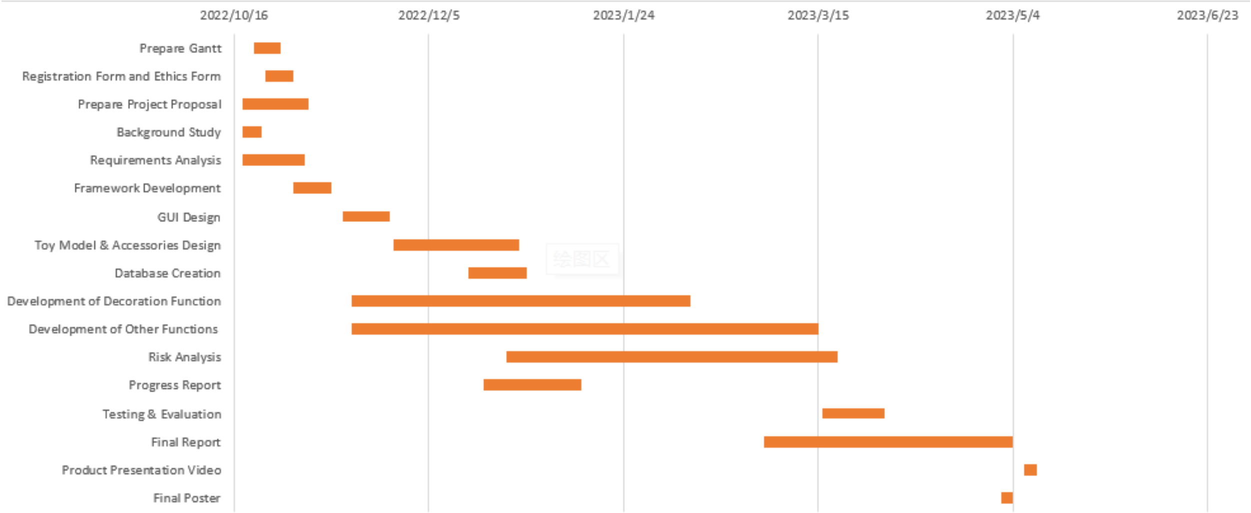
Objectives	Activities	Complete State	Start Date	End Date
Carry out a comprehensive background study and investigation of existing online toy stores	1. Search mainstream online toy stores 2. Create a feature comparison table 3. Summary limitations and disadvantages	Completed	2022/10/18	20222/11/4
Develop the framework of the web application	1. Install indeed software for constructing the web application 2. Configure the environment of the web application 3. Design UI of the basic framework of web application	Completed	2022/10/31	2022/11/10
Design the basic toy models and related toy accessories	1. Design the basic model and accessories of toys	Completed	2022/11/26	2022/12/28

	2. Modify the draft of toys 3. Build models and accessories by 3dsMax 4. Modify and perfect models			
Develop the function of decorating toys	1. Upload the models from 3dsMax to Vue 2. Program to decorate accessories on models 3. Deploy the Decorative function in the web application	Completed	2022/11/15	2023/2/10
Complete other related shopping online functions	1. Design the user interface of each function 2. Develop the front-end part 3. Develop the back-end part 4. Build the connection of frontend and backend 5. Test each function	Completed	2022/11/15	2022/3/15

Build database	1. Design ERD of this web application 2. Construct each table and its relationship 3. Connect database with back-end 4. Test whether connects successfully	Completed	2022/12/15	2/22/12/30
Testing and maintenance of the web application	1. Decide the development model 2. Follow the rules of the software development model 3. Maintenance	Completed	2023/3/16	2023/4/1

5.1.2 Schedule

The schedule is shown in Gantt Chart in Figure 29.



. Figure 29 Gantt Chart of the project

5.1.3 Project Data Management

The GitHub platform would be used for the data management of this project to record the progress of the project. Here is the URL of the GitHub account to be used: https://github.com/Wangyiting1234/project

The data management plan for this project is as follows:

Data	Management method
Ethics forms	Documentations in folder
Plan (weekly meeting logs and schedule)	Documentations in folder
Reports	Documentations in folder
Literature	Mendeley
Testing and evaluation	Documentations in folder

Table 40 The data management of the project

5.1.4 Project Deliverables

The deliverables of this project are as follows:

a. Project proposal
b. Progress report
c. Code of project

d. Poster presentation

e. Final report

5.2 Risk Analysis & Mitigation Strategies

This section discusses the project in terms of its requirements, development, technology, and operation to identify the types of risks that will be encountered during the project as well as the corresponding mitigation strategies.

Colored Cells are the Risk Categories	Low Risk	Moderate Risk	High Risk	Very High Risk

Table 41 The color cell of risk categories

Extreme	5	10	15	20	25
Severe	4	8	12	16	20
Substantial	3	6	9	12	15
Moderate	2	4	6	8	10
Slight	1	2	3	4	5
	Very Unlikely	**Not Likely**	**Likely**	**Very Likely**	**Extremely Likely**

Table 42 Risk Severity Matrix

Each type of risk was given a score of 5 for severity/consequence and likelihood of occurrence. The severity of the risk is multiplied by the likelihood of occurrence to give an overall risk rating. The overall risk rating is presented in a matrix, shown in Table 44, to demonstrate the complete level of risk severity.

Risk ID	Risk	Analysis	Severity	Likelihood	Risk
1	Requirements change risk	Requirements have become project	2	1	2

		benchmarks, but requirements continue to change			🟩
2	Progress risk	The commitment to the progress of the project has far exceeded the actual project schedule, resulting in serious time problems at the beginning of the project	5	4	20
3	Technical risks	The technical requirements of the project are not possessed or not mastered	4	3	12
4	Software bugs	Poor test plan and techniques	4	2	8
5	Loss of data	Poor plan of data management	5	2	10
6	No Information displayed on the website	The front end and back end of the project are not connecting successfully	5	4	20
7	Failure of the development environment	Update the version of the computer system	4	4	16

8	Failure to show the website while the presentation	Computer encounters error	5	1	5

Table 43 The risk analysis of the project

According to the above risk analysis, the mitigation strategies will be suggested in Table 45:

Risk ID	Risk	Mitigation Strategy
1	Requirements change risk	1. Familiar with the project process, functional requirements, system concept, and project objectives, regular work progress. 2. Regularly summarize the progress of work and existing problems in the needs analysis stage
2	Progress risk	1. In the time schedule management of project implementation, consider various potential factors and leave appropriate time to review and perfect details of the project 2. The task decomposition should be detailed to be familiar with what needs to be done at each step 3. In the whole process of project progress, compare the planned progress with

		the actual progress, find deviations in time, and take timely measures to correct or prevent
3	Technical risks	1. Be cautious in the use of new technologies and try to use mature technical solutions to complete software development work. 2. Familiarize with the developer documentation 3. Choose familiar technical to develop the website
4	Software bugs	1. Create a test plan early 2. Test after finishing each unit of the website
5	Loss of data	1. Create data version management at the start 2. Upload the new version of the website timely 3. Create cloud management of the website
6	No Information displayed on the website	1. Check the front and back ends are working properly 2. Use the developer tools to check for error messages 3. Ensure that the domain name, protocol, and port of the request sent by the

		front end matches the back-end
7	Failure of the development environment	Ensure all code, data, and documents are backed up and could be downloaded & decompress
8	Failure to show the website while the presentation	Prepare a recorded presentation video in advance

Table 44 The mitigation strategies of the project

5.3 Professional Issues

This section discusses environmental, legal, social, and ethical issues in the context of product development as well as the final product.

5.3.1 Environmental issues

One of the main environmental issues facing online toy shopping is the growing rivalry from other e-commerce platforms. Smaller online toy retailers are challenged to compete with growing businesses like Amazon, Walmart, Target, and Taobao. To engage customers in the design of toys, the web application provides a custom toy service, providing consumers with a distinctive purchasing experience. In addition, according to the BSC code of conduct [21], not discriminate based on gender, sexual orientation, marital status, nationality, color, race, ethnic origin, religion, age, or disability in the design, development, and user experience of the web applications, or any other condition or requirements. Besides, the online purchase and customization service helps more people to buy toys that are not available in their local shops or cannot be purchased offline for personal reasons, reducing the time and money spent.

5.3.2 Legal issues

With respect to this project, there are a few legal issues that need to be addressed including the Copyright, Designs and Patents Act,1988[22], Data Protection Act 2018[23]

For the Copyright, Designs, and Patents Act,1988, there are two parts of this project can be affected:

Part	**Effect**

Database	Any act by a person entitled to use the database or any part of the database necessary to access and use the contents of the database or the contents of the database in the workout of that right in relation to the database on the website is not an infringement of the copyright of the database
Images of toys	All images presented on this website are used with the permission of the registrant of the image.

Table 45 The parts of this web application will be affected by the Copyright, Designs, and Patents Act,1988[22]

In accordance with the interpretation of the Data Protection Act 2018[23] regarding the Data subject's right of access, the web application allows the data subject to request the controller to correct personal information, including information on a personal name, gender, address of receipt, personal collections, etc. The data subject also has the right to request the controller to delete information on the address of receipt, personal collections, etc.

5.3.3 Ethical issues

The process of developing and designing a web application involves the use of excellent web development processes, rigorous testing processes, and design templates, following a study in these areas. These approaches were applied in the development and design of the website, with a final presentation devoid of any fraudulent or plagiarized content. Furthermore, the website offers a customized toy service that accommodates the unique needs of consumers, as opposed to merely presenting identical products available in the market. This approach accords customers the respect of their individuality and preference.To mitigate the risk of shoppers purchasing counterfeit or misrepresented products, the website provides accurate toy dimensions, parameters, descriptions, and detailed display drawings that customers can use to make informed choices. Consequently, users do not have to rely on product selections that do not provide access to actual parameters and photos. Additionally, all data on the website is stored centrally in a database with restricted access, ensuring the confidentiality of personal information of users. This process aims to curb the menace of online toy brands collecting, storing, and processing a vast amount of personal data of users who shop on their websites, which is sold to third-party marketing agencies.

5.3.4 Social issues

The web application that allows customers to customize their toys has a variety of effects on society. Firstly, the individual needs of consumers can be satisfied through customized toys, increasing customer happiness. Secondly, toys can be customized by consumers, which reduces waste and prevents environmental damage. Thirdly, consumers can expand their sense of creativity and imagination by designing their own toys. Finally, it can assist additional individuals who find it difficult to leave the house to purchase the toys they require. Therefore, the website provides various positive influences for society.

Chapter 6 Conclusion

6.1 Conclusion

This document presents an overview of the online toy brand, including its introduction, objectives, implementation approach, outcomes, and testing results. The primary aim of this project is to provide consumers with an online platform where they can purchase standard or customized toys. The Vue + Springboot framework was utilized to develop both the front-end and back-end of the online toy store. The web application is equipped with various features such as user accounts, delivery addresses, order details, product displays, and personalized teddy bears that enhance the user experience. The data and information generated are stored in databases to facilitate ease of access and modification. Prior to processing any data, rigorous testing was conducted to ensure the effective functioning of all components of the web application.

6.2 Reflections

My skills and expertise in the field of BSc software engineering have grown as a result of accomplishing this project. The project encompassed an initial requirements analysis, comprehensive technical study, risk assessment, and project strategy, enabling me to gain valuable insights into the IT industry. Although the project met the essential requirements and achieved the basic functionality required for an online toy brand, several areas for enhancement were identified in the user experience of the web application. For instance, customers who frequently forget their passwords face inconvenience as the user account system mandates the use of phone number and password for logging in. Moreover, the inability of users to save customized teddy bear decorations to the shopping cart and to instantly ascertain the cost of decorations while designing creates suboptimal user experience and budgeting challenges. The website also lacks features such as the ability to share the personalized teddy bear on social media and the user interface and functionality for payments made through WeChat, Alipay, etc. Addressing these areas could drive innovation and markedly improve the website's functionality and usability.

6.3 Future Work

In the future, customized toy modules will be developed firstly in more depth to bring a better customization experience to the users. The base toy models' detailing will be increased as part of the model development process, for example, by giving the toy more hair in order to provide it with a more realistic appearance. Second, by including more fundamental models to improve user selectivity and engagement. Third, customers will first be able to check the costs of accessories in real-time when consumers click on the accessories button for functional development. Finally, users complete their decorating, and consumers are able to share the outcomes of their designs on social networking sites like WeChat, Weibo, and Instagram. A back-office administration system will also be developed. By logging in to the administrator account and making modifications to the products including creating new products and related parameters, modifying related information of existing products, deleting existing products, and so forth. The administrator also could create and modify information on the user’s order. Meanwhile, the administrator will be able to amend the users' personal information.

References

[1] LEGO, “https://www.lego.com/zh-cn,” 2022.

[2] Jellycat, “https://www.jellycat.com/us/,” 2022.

[3] Gökçe Esenduran, Paolo Letizia, and Anton Ovchinnikov, “Why You Should Allow Returns on Customized Products,” *Havard Business Review*, Mar. 09, 2022. Accessed: Apr. 12, 2023. [Online]. Available: https://hbr.org/2022/03/why-you-should-allow-returns-on-customized-products

[4] NIKE, “https://www.nike.com/nike-by-you,” 2022.

[5] Apple, “https://www.apple.com.cn/.”

[6] POPMART, “https://global.popmart.com/,” 2022.

[7] Mattel, “https://about.mattel.com/,” 2022.

[8] Vans, “https://vans.com.cn/customs.html,” 2022.

[9] S. Al-Saqqa, S. Sawalha, and H. Abdelnabi, “Agile Software Development: Methodologies and Trends”, doi: 10.3991/ijim.v14i11.13269.

[10] IDEA, “https://www.jetbrains.com/zh-cn/idea/.”

[11] Visual Studio Code, “https://code.visualstudio.com/.”

[12] “Navicat Premium | Connects to multiple databases on a single GUI.” https://navicat.com/en/products/navicat-premium (accessed Mar. 23, 2023).

[13] Vue, “https://vuejs.org/.”

[14] 3ds Max, “https://www.autodesk.com/products/3ds-max/overview.”

[15] Bootstrap, “https://getbootstrap.com/”.

[16] Balsmiq Wrieframes, “https://balsamiq.com/wireframes/.”

[17] Photoshop, “https://www.photoshop.com/”.

[18] Vant, “https://vant-contrib.gitee.io/vant/v3/#/zh-CN.”

[19] Iconfont, “https://www.icofont.com.”

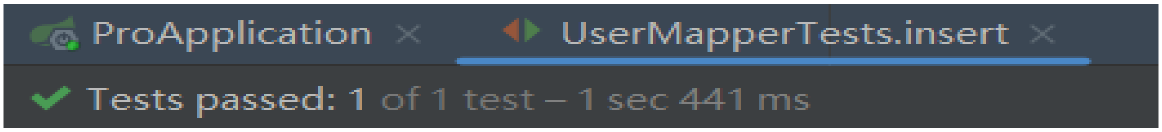
[20] J. Qiu, “Behavior-Driven Development,” in *Test-Driven Development with React*, Berkeley, CA: Apress, 2021, pp. 165–177. doi: 10.1007/978-1-4842-6972-5_10.

[21] BSC, “https://www.bsc.org,” 2020.

[22] UK Public General Acts, “Copyright, Designs and Patents Act 1988,” 1988, Accessed: Apr. 27, 2023. [Online]. Available: https://www.legislation.gov.uk/ukpga/1988/48/contents

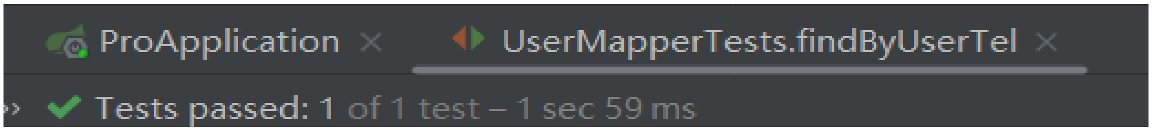
[23] UK Public General Acts, “Data Protection Act 2018,” 2018, Accessed: Apr. 27, 2023. [Online]. Available: https://www.legislation.gov.uk/ukpga/2018/12/contents/enacted

Appendices



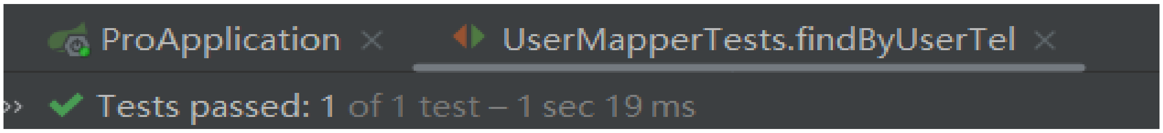
Tests passed: 1 of 1 test – 1 sec 441 ms

Figure 30 Actual output of unit test case 1 in user mapper layer



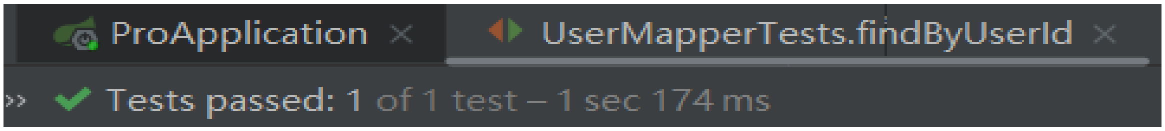
Tests passed: 1 of 1 test – 1 sec 59 ms

Figure 31 Actual output of unit test case 2 in user mapper layer



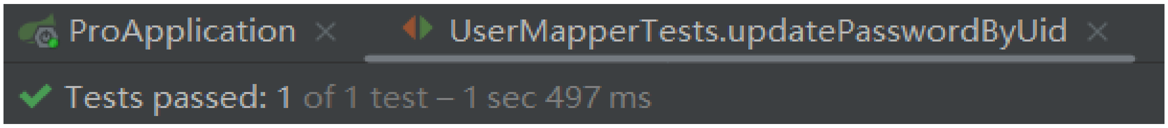
Tests passed: 1 of 1 test – 1 sec 19 ms

Figure 32 Actual output of unit test case 3 in user mapper layer



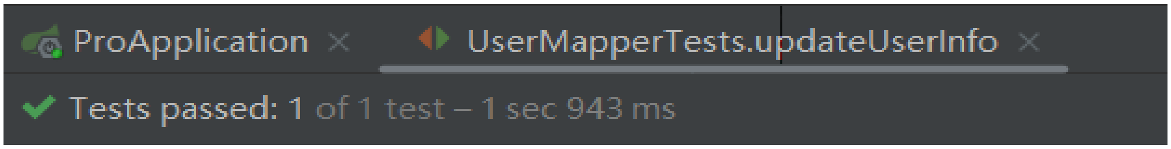
Tests passed: 1 of 1 test – 1 sec 174 ms

Figure 33 Actual output of unit test case 4 in user mapper layer



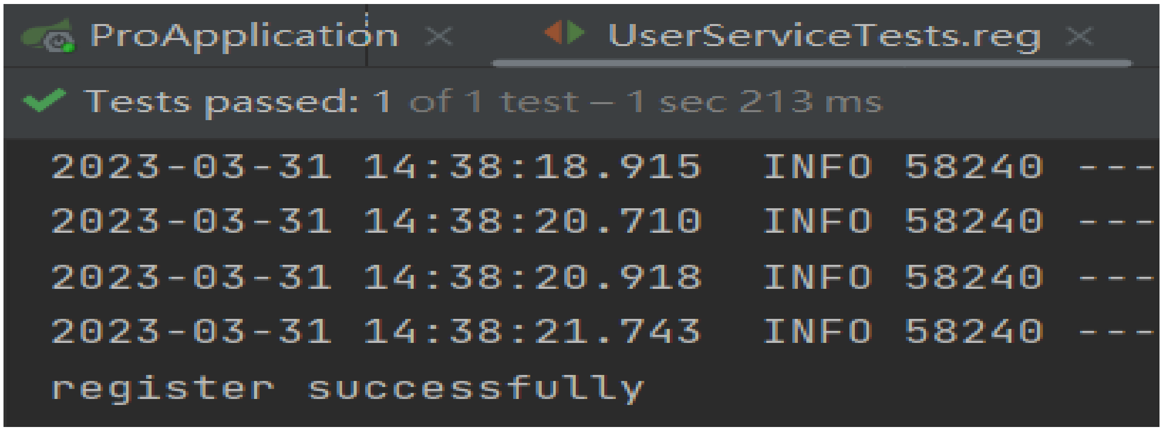
Tests passed: 1 of 1 test – 1 sec 497 ms

Figure 34 Actual output of unit test case 5 in user mapper layer



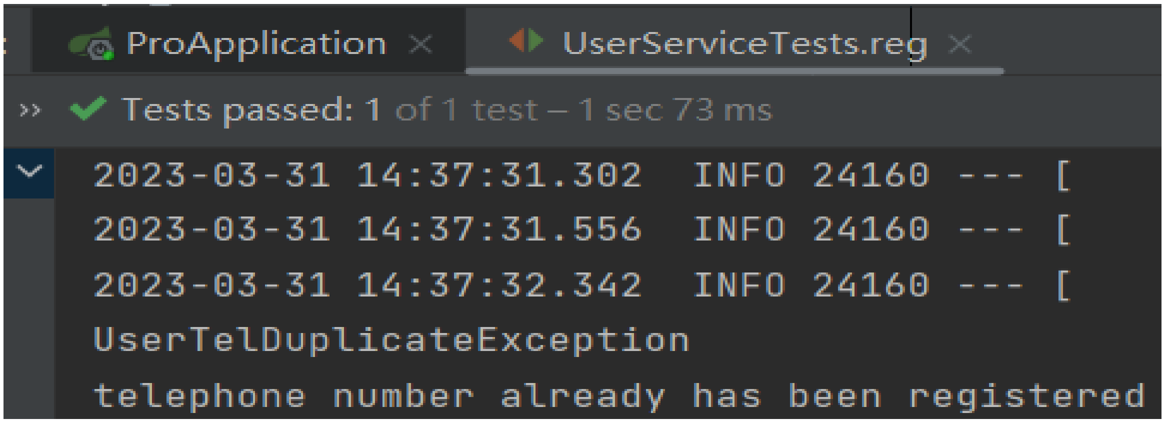
Tests passed: 1 of 1 test – 1 sec 943 ms

Figure 35 Actual output of unit test case 6 in user mapper layer



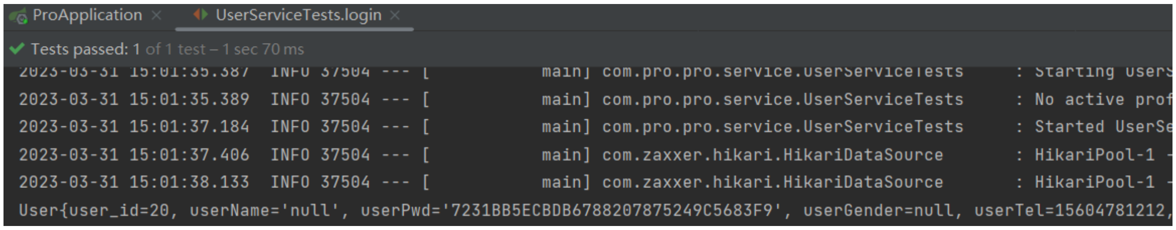
```
Tests passed: 1 of 1 test – 1 sec 213 ms
2023-03-31 14:38:18.915  INFO 58240 ---
2023-03-31 14:38:20.710  INFO 58240 ---
2023-03-31 14:38:20.918  INFO 58240 ---
2023-03-31 14:38:21.743  INFO 58240 ---
register successfully
```

Figure 36 Actual output of unit test case 1 in user servicer layer while user telephone not exists



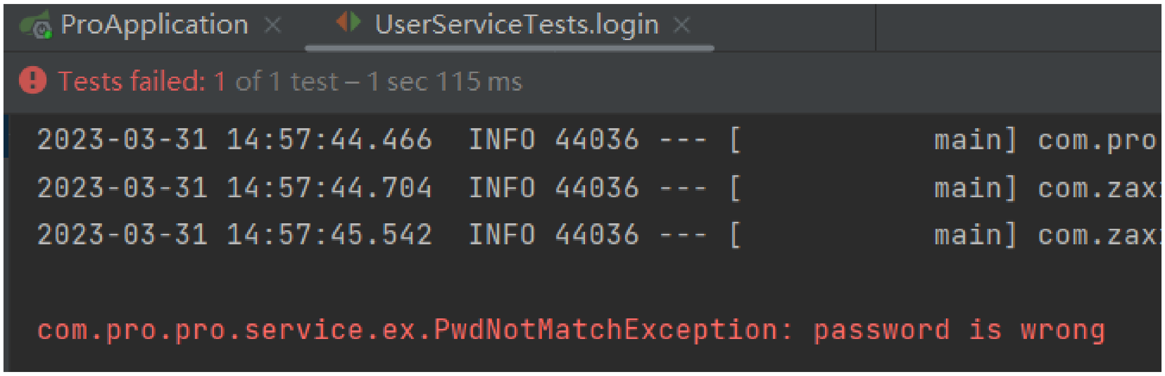
```
Tests passed: 1 of 1 test – 1 sec 73 ms
2023-03-31 14:37:31.302  INFO 24160 --- [
2023-03-31 14:37:31.556  INFO 24160 --- [
2023-03-31 14:37:32.342  INFO 24160 --- [
UserTelDuplicateException
telephone number already has been registered
```

Figure 37 Actual output of unit test case 1 in user servicer layer while user telephone exits



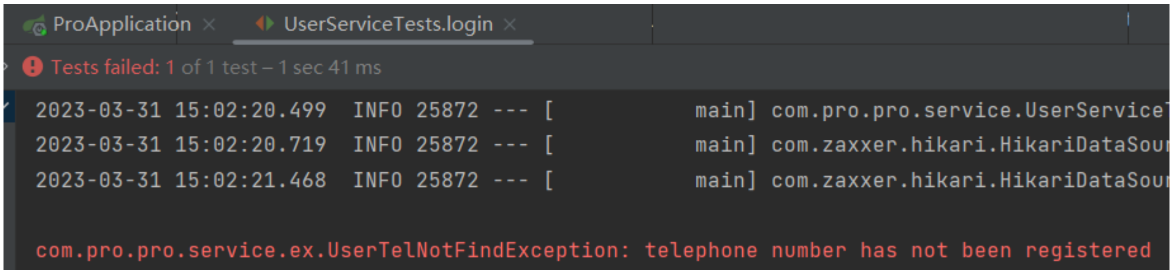
```
Tests passed: 1 of 1 test – 1 sec 70 ms
2023-03-31 15:01:35.387  INFO 37504 --- [           main] com.pro.pro.service.UserServiceTests     : Starting UserS
2023-03-31 15:01:35.389  INFO 37504 --- [           main] com.pro.pro.service.UserServiceTests     : No active prof
2023-03-31 15:01:37.184  INFO 37504 --- [           main] com.pro.pro.service.UserServiceTests     : Started UserSe
2023-03-31 15:01:37.406  INFO 37504 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 -
2023-03-31 15:01:38.133  INFO 37504 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 -
User{user_id=20, userName='null', userPwd='7231BB5ECBDB6788207875249C5683F9', userGender=null, userTel=15604781212,
```

Figure 38 Actual output of unit test case 2 in user servicer layer while user telephone and password match



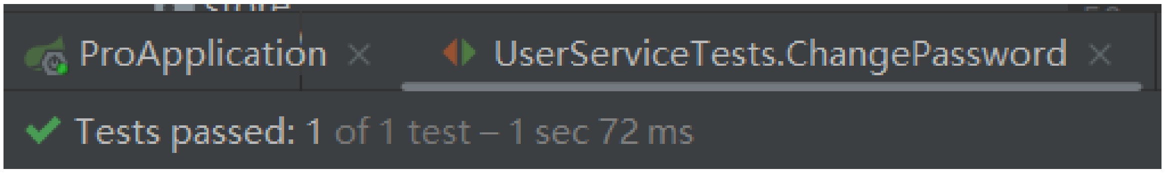
```
Tests failed: 1 of 1 test – 1 sec 115 ms
2023-03-31 14:57:44.466  INFO 44036 --- [           main] com.pro
2023-03-31 14:57:44.704  INFO 44036 --- [           main] com.zax
2023-03-31 14:57:45.542  INFO 44036 --- [           main] com.zax

com.pro.pro.service.ex.PwdNotMatchException: password is wrong
```

Figure 39 Actual output of unit test case 2 in user servicer layer while user telephone and password do not match



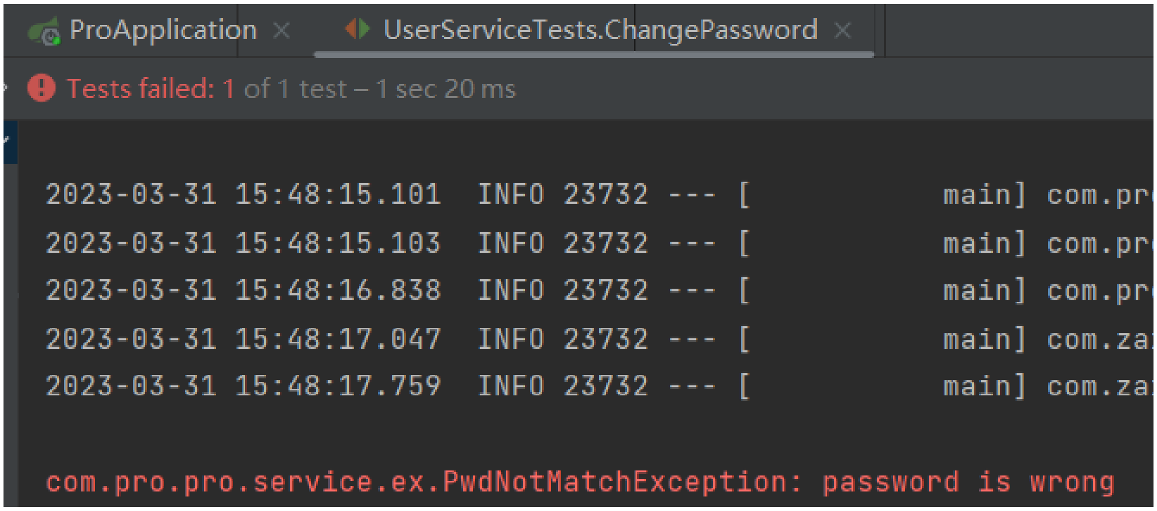
```
Tests failed: 1 of 1 test – 1 sec 41 ms
2023-03-31 15:02:20.499  INFO 25872 --- [           main] com.pro.pro.service.UserService
2023-03-31 15:02:20.719  INFO 25872 --- [           main] com.zaxxer.hikari.HikariDataSou
2023-03-31 15:02:21.468  INFO 25872 --- [           main] com.zaxxer.hikari.HikariDataSou

com.pro.pro.service.ex.UserTelNotFindException: telephone number has not been registered
```

Figure 40 Actual output of unit test case 2 in the user servicer layer while the user’s telephone is not stored on the database



```
Tests passed: 1 of 1 test – 1 sec 72 ms
```

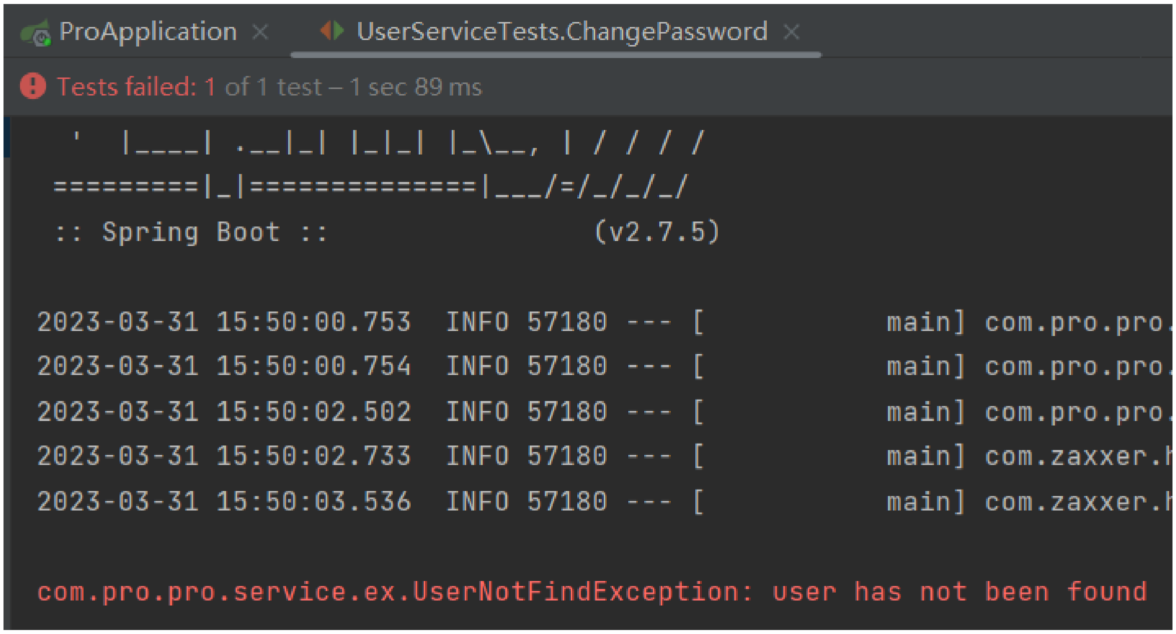
Figure 41 Actual output of unit test case 3 in user servicer layer while the user's current password matches with the password of database and user can be found



Tests failed: 1 of 1 test – 1 sec 20 ms

```
2023-03-31 15:48:15.101  INFO 23732 --- [           main] com.pr
2023-03-31 15:48:15.103  INFO 23732 --- [           main] com.pr
2023-03-31 15:48:16.838  INFO 23732 --- [           main] com.pr
2023-03-31 15:48:17.047  INFO 23732 --- [           main] com.za
2023-03-31 15:48:17.759  INFO 23732 --- [           main] com.za

com.pro.pro.service.ex.PwdNotMatchException: password is wrong
```

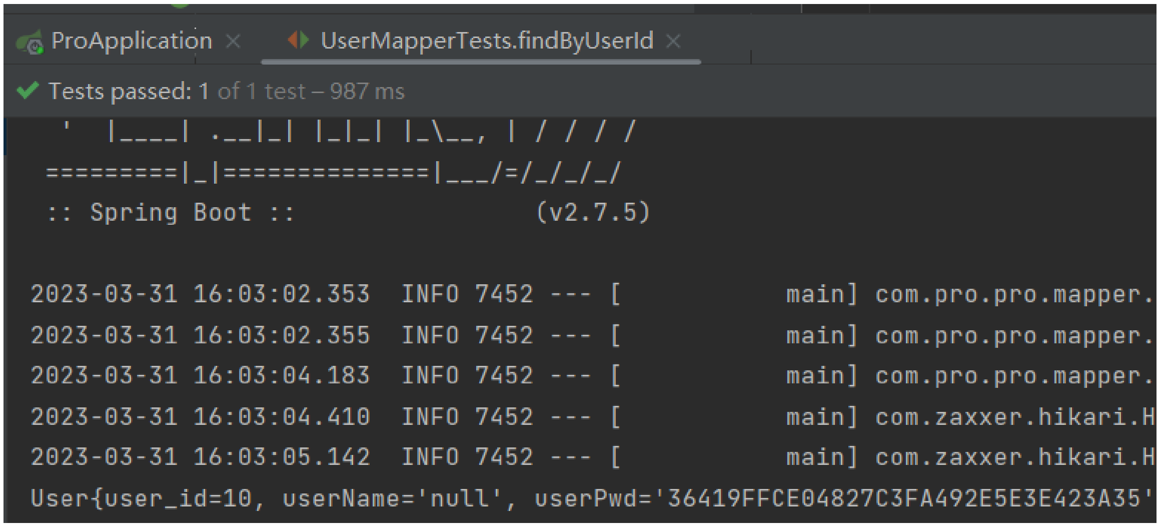
Figure 42 Actual output of unit test case 3 in user servicer layer, while the user's current password is not matched with the password of the database and user, can be found



Tests failed: 1 of 1 test – 1 sec 89 ms

```
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.7.5)

2023-03-31 15:50:00.753  INFO 57180 --- [           main] com.pro.pro.
2023-03-31 15:50:00.754  INFO 57180 --- [           main] com.pro.pro.
2023-03-31 15:50:02.502  INFO 57180 --- [           main] com.pro.pro.
2023-03-31 15:50:02.733  INFO 57180 --- [           main] com.zaxxer.h
2023-03-31 15:50:03.536  INFO 57180 --- [           main] com.zaxxer.h

com.pro.pro.service.ex.UserNotFindException: user has not been found
```

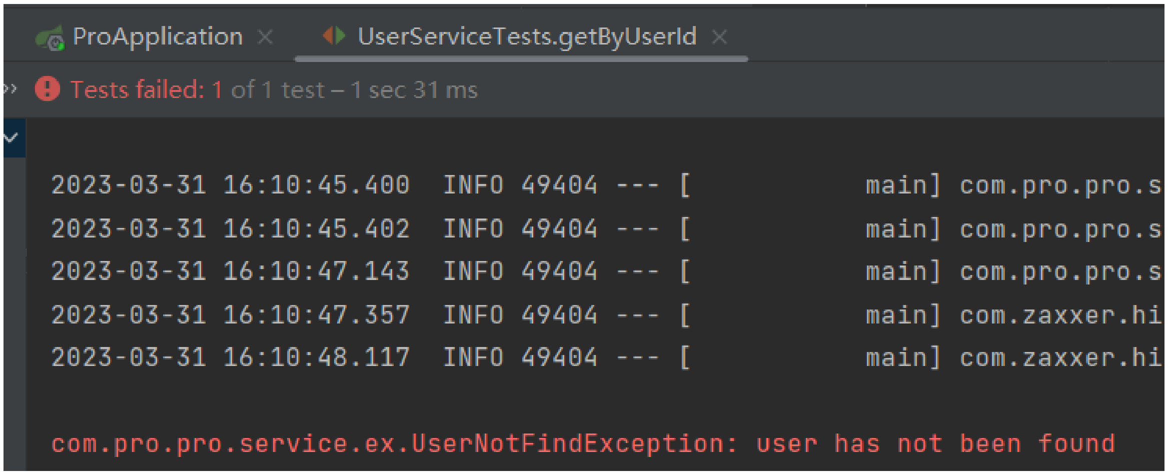
Figure 43 Actual output of unit test case 3 in user servicer layer while user’s id can be not found in the database



Tests passed: 1 of 1 test – 987 ms

```
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.7.5)

2023-03-31 16:03:02.353  INFO 7452 --- [           main] com.pro.pro.mapper.
2023-03-31 16:03:02.355  INFO 7452 --- [           main] com.pro.pro.mapper.
2023-03-31 16:03:04.183  INFO 7452 --- [           main] com.pro.pro.mapper.
2023-03-31 16:03:04.410  INFO 7452 --- [           main] com.zaxxer.hikari.H
2023-03-31 16:03:05.142  INFO 7452 --- [           main] com.zaxxer.hikari.H
User{user_id=10, userName='null', userPwd='36419FFCE04827C3FA492E5E3E423A35'
```

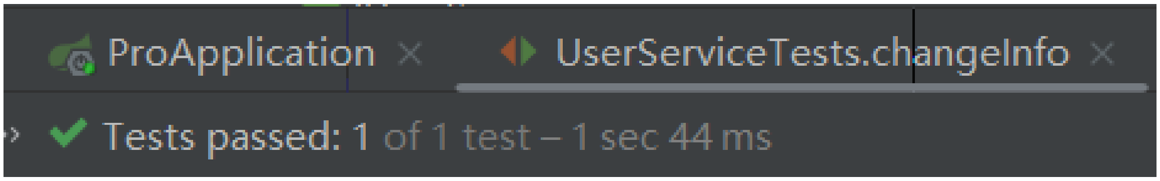
Figure 44 Actual output of unit test case 4 in user servicer layer while user id can be found in the database and return user's gender and name



Tests failed: 1 of 1 test – 1 sec 31 ms

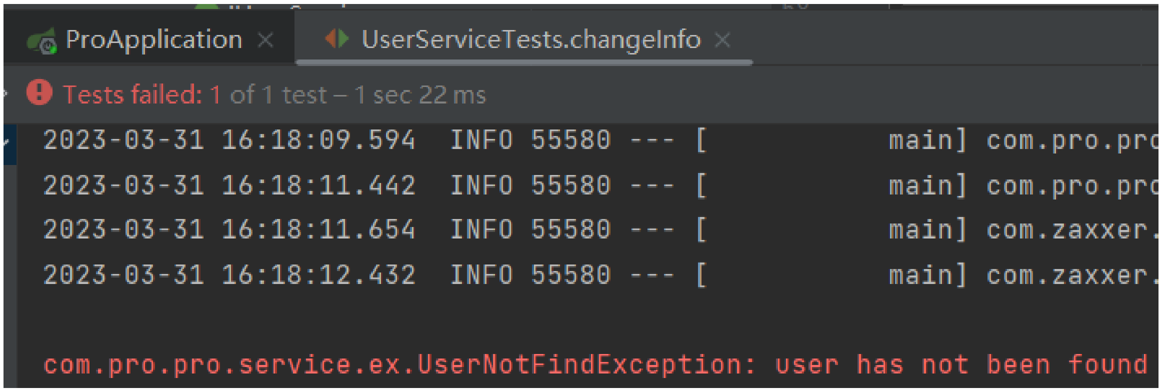
```
2023-03-31 16:10:45.400  INFO 49404 --- [           main] com.pro.pro.s
2023-03-31 16:10:45.402  INFO 49404 --- [           main] com.pro.pro.s
2023-03-31 16:10:47.143  INFO 49404 --- [           main] com.pro.pro.s
2023-03-31 16:10:47.357  INFO 49404 --- [           main] com.zaxxer.hi
2023-03-31 16:10:48.117  INFO 49404 --- [           main] com.zaxxer.hi

com.pro.pro.service.ex.UserNotFindException: user has not been found
```

Figure 45 Actual output of unit test case 4 in user servicer layer while user's id can be not found in the database



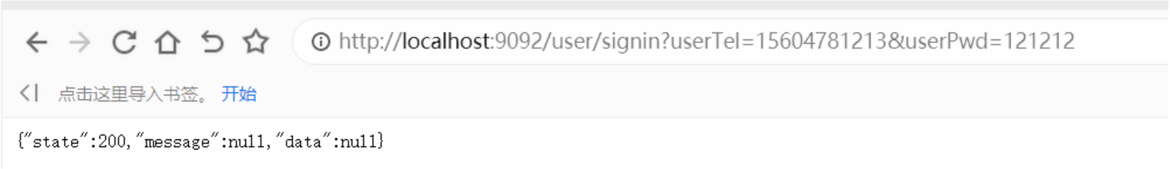
Tests passed: 1 of 1 test – 1 sec 44 ms

Figure 46 Actual output of unit test case 5 in user servicer layer while user's id can be found in the database and update user gender and name



```
Tests failed: 1 of 1 test – 1 sec 22 ms
2023-03-31 16:18:09.594  INFO 55580 --- [           main] com.pro.pro
2023-03-31 16:18:11.442  INFO 55580 --- [           main] com.pro.pro
2023-03-31 16:18:11.654  INFO 55580 --- [           main] com.zaxxer.
2023-03-31 16:18:12.432  INFO 55580 --- [           main] com.zaxxer.

com.pro.pro.service.ex.UserNotFindException: user has not been found
```

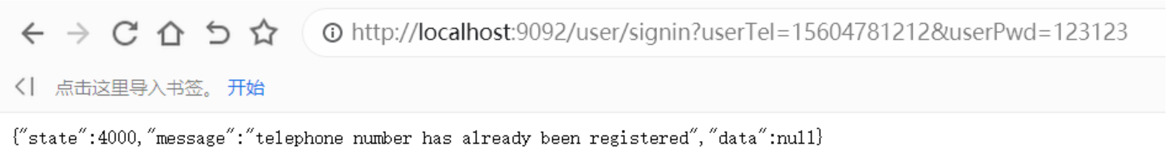
Figure 47 Actual output of unit test case 5 in user servicer layer while user’s id can be not found in the database and throw exceptions



http://localhost:9092/user/signin?userTel=15604781213&userPwd=121212

```
{"state":200,"message":null,"data":null}
```

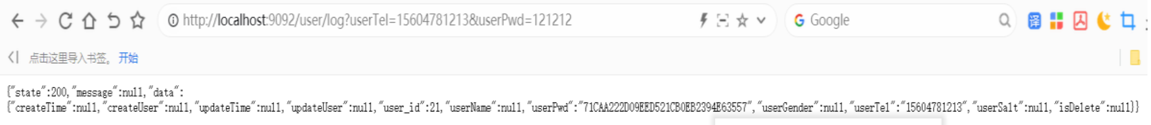
Figure 48 Actual output of integration test case 1 in User while user’s information can be inserted into the database successfully by HTTP request



http://localhost:9092/user/signin?userTel=15604781212&userPwd=123123

```
{"state":4000,"message":"telephone number has already been registered","data":null}
```

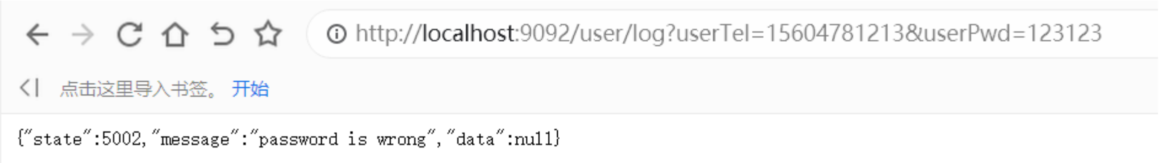
Figure 49 Actual output of integration test case 1 in User while the existing user’s information can be inserted into the database failingly by HTTP request



http://localhost:9092/user/log?userTel=15604781213&userPwd=121212

```
{"state":200,"message":null,"data":
{"createTime":null,"createUser":null,"updateTime":null,"updateUser":null,"user_id":21,"userName":null,"userPwd":"71CAA222D09EED521CB0EB2394E63557","userGender":null,"userTel":"15604781213","userSalt":null,"isDelete":null}}
```

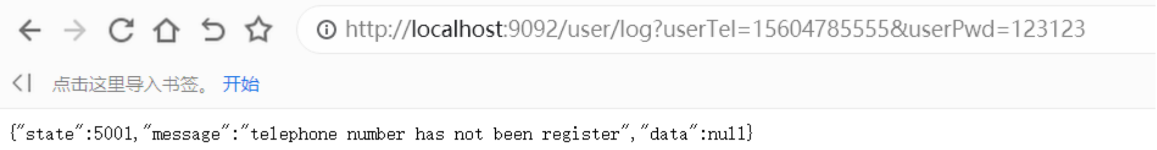
Figure 50 Actual output of integration test case 2 in User while the existing user’s information can be matched with data on the database by HTTP request



http://localhost:9092/user/log?userTel=15604781213&userPwd=123123

```
{"state":5002,"message":"password is wrong","data":null}
```

Figure 51 Actual output of integration test case 2 in User while the requested password can be not matched with the password on the database by HTTP request

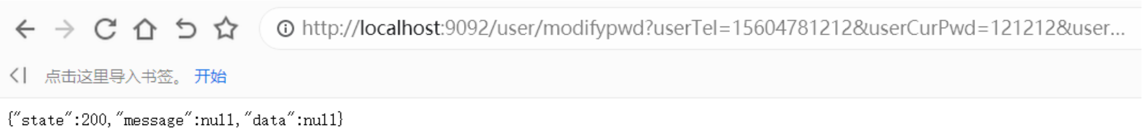


http://localhost:9092/user/log?userTel=15604785555&userPwd=123123

<| 点击这里导入书签。 开始

```
{"state":5001,"message":"telephone number has not been register","data":null}
```

Figure 52 Actual output of integration test case 2 in User while the requested telephone can be not found from the database by HTTP request

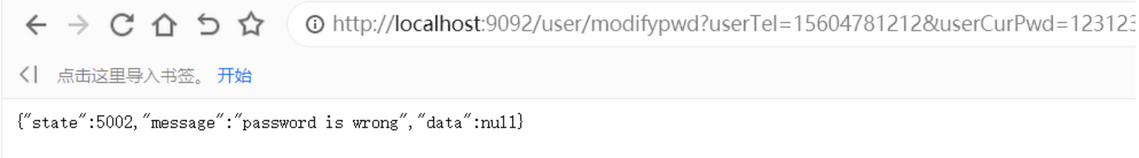


http://localhost:9092/user/modifypwd?userTel=15604781212&userCurPwd=121212&user...

<| 点击这里导入书签。 开始

```
{"state":200,"message":null,"data":null}
```

Figure 53 Actual output of integration test case 3 in User while the user exists and the current password can be matched with data on the database by HTTP request

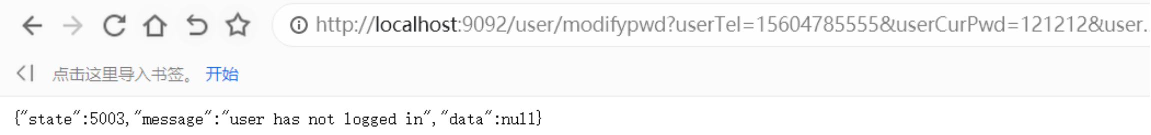


http://localhost:9092/user/modifypwd?userTel=15604781212&userCurPwd=12312

<| 点击这里导入书签。 开始

```
{"state":5002,"message":"password is wrong","data":null}
```

Figure 54 Actual output of integration test case 3 in User while the requested password can be not matched with password on the database by HTTP request

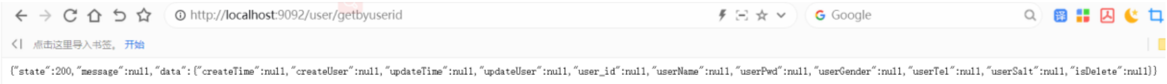


http://localhost:9092/user/modifypwd?userTel=15604785555&userCurPwd=121212&user.

<| 点击这里导入书签。 开始

```
{"state":5003,"message":"user has not logged in","data":null}
```

Figure 55 Actual output of integration test case 2 in User while user information has not stored in session before updating password

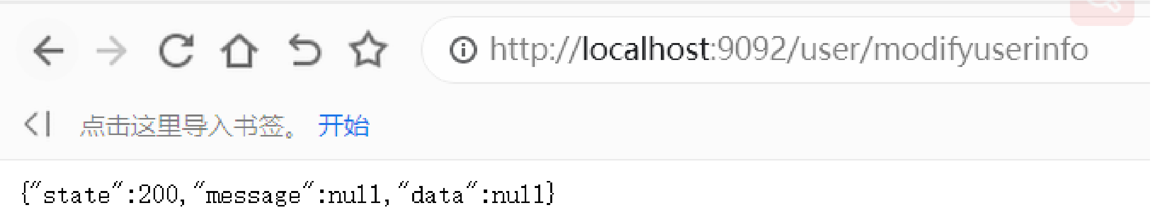


http://localhost:9092/user/getbyuserid

<| 点击这里导入书签。 开始

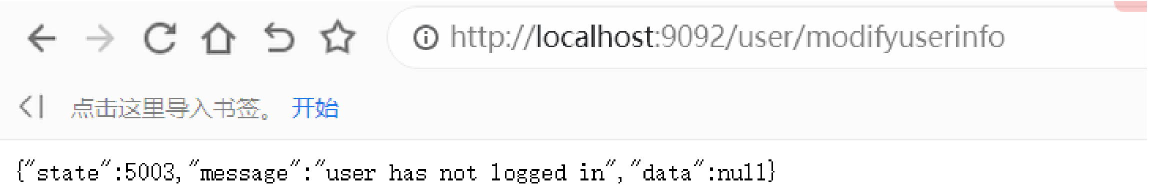
```
{"state":200,"message":null,"data":{"createTime":null,"createUser":null,"updateTime":null,"updateUser":null,"user_id":null,"userName":null,"userPwd":null,"userGender":null,"userTel":null,"userSalt":null,"isDelete":null}}
```

Figure 56 Actual output of the integration test case 4 in User



```
{″state″:200,″message″:null,″data″:null}
```

Figure 57 Actual output of the integration test case 5 in User while the user has logged in



```
{″state″:5003,″message″:″user has not logged in″,″data″:null}
```

Figure 58 Actual output of integration test case 5 in User while the user has not logged in

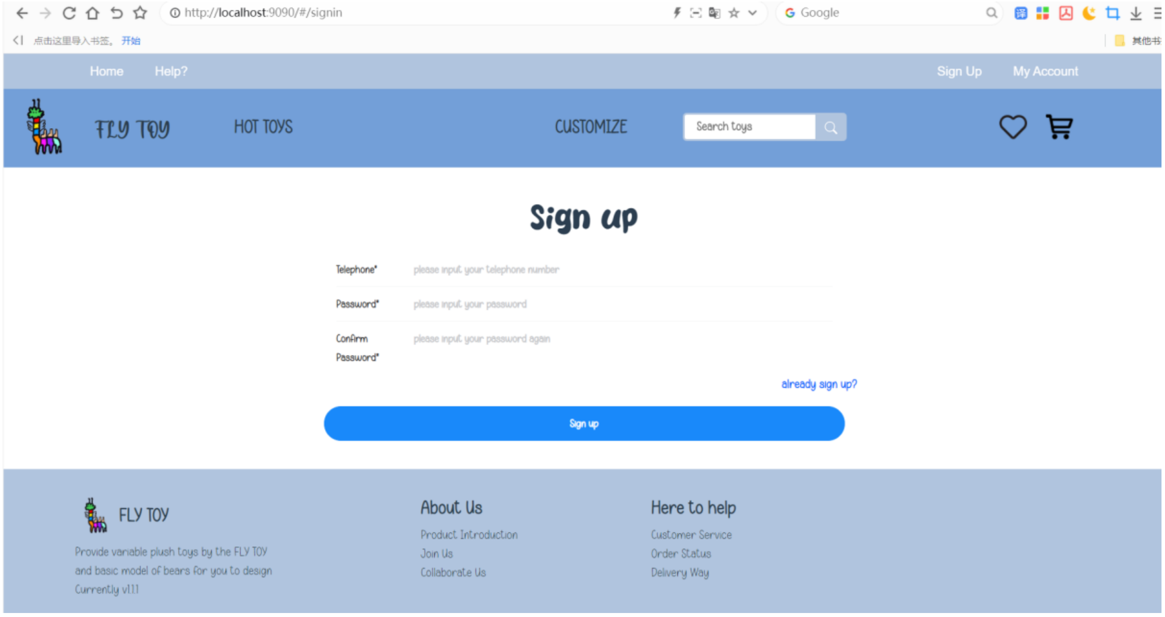
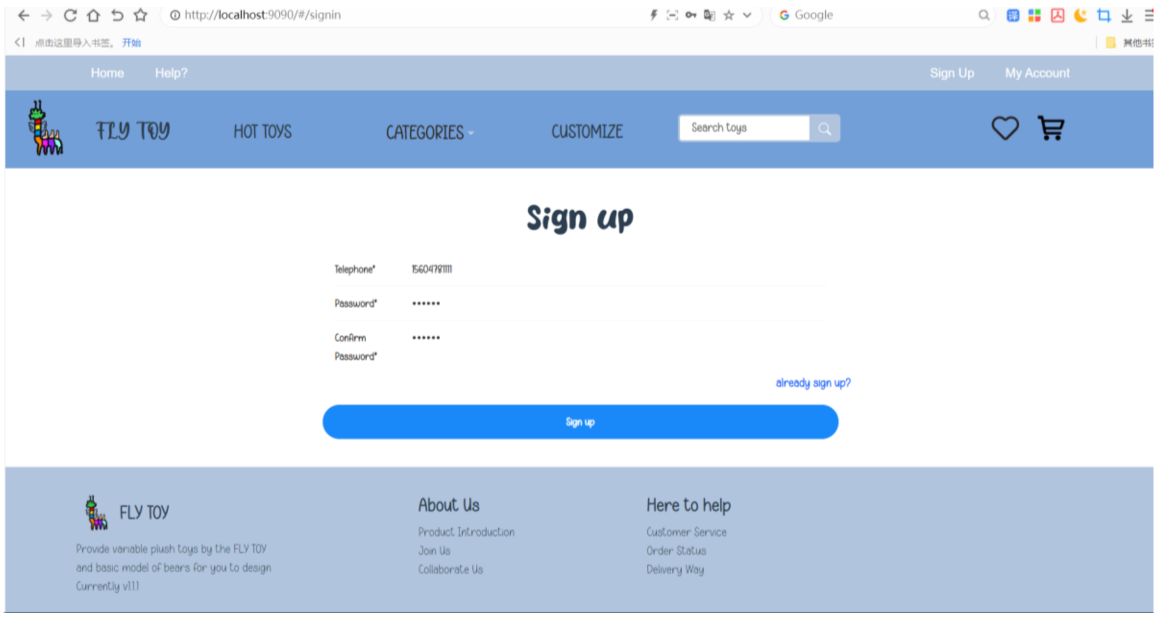


Figure 59 Interface Test Result of Register



Sign up

Telephone* 15604781111

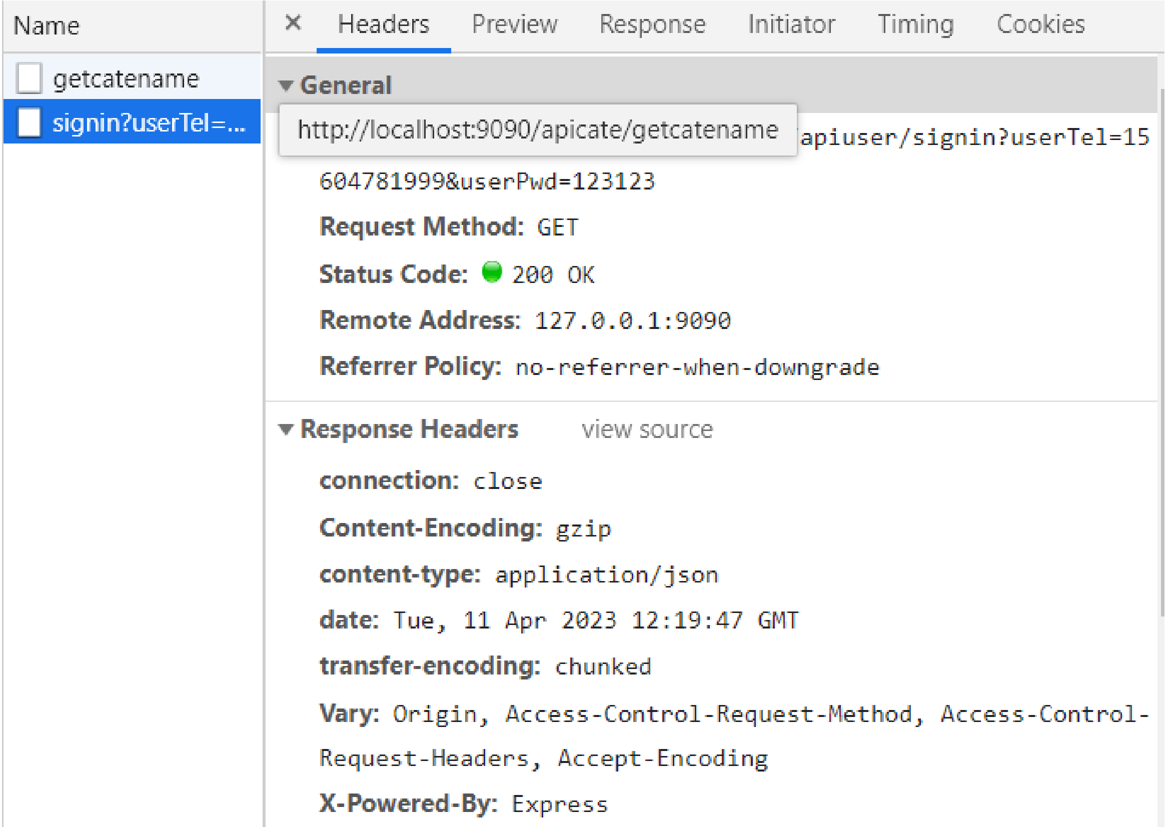
Password* ••••••

Confirm Password* ••••••

already sign up?

Sign up

Figure 60 The results of inputting the correct format of the password



General

http://localhost:9090/apicate/getcatename apiuser/signin?userTel=15604781999&userPwd=123123

Request Method: GET

Status Code: ● 200 OK

Remote Address: 127.0.0.1:9090

Referrer Policy: no-referrer-when-downgrade

Response Headers

connection: close

Content-Encoding: gzip

content-type: application/json

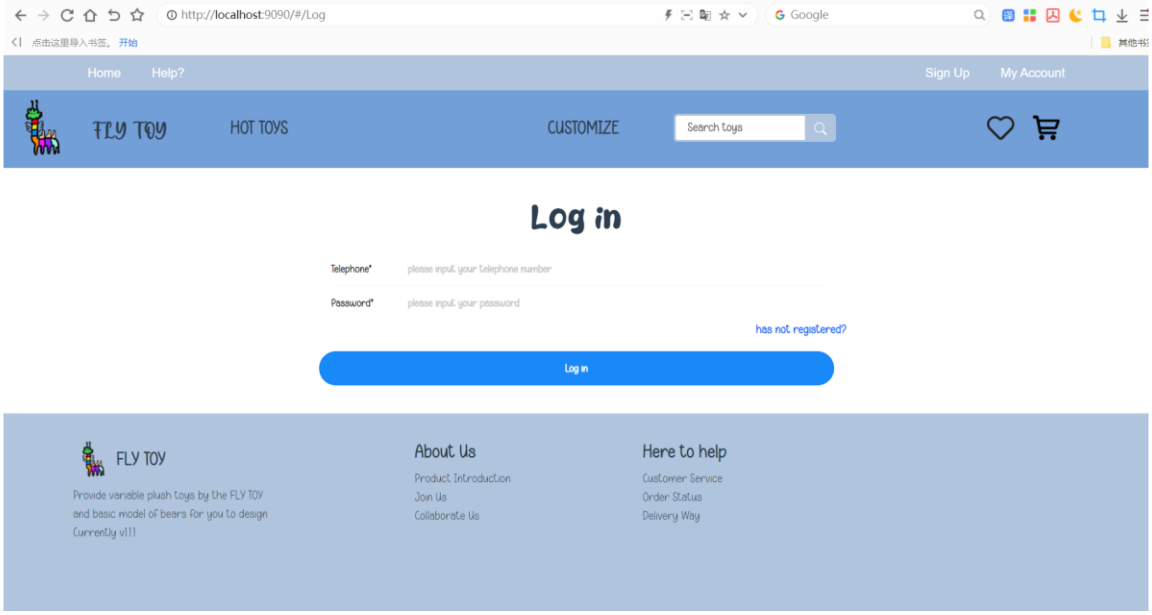
date: Tue, 11 Apr 2023 12:19:47 GMT

transfer-encoding: chunked

Vary: Origin, Access-Control-Request-Method, Access-Control-Request-Headers, Accept-Encoding

X-Powered-By: Express

Figure 61 The results of registering successfully



Log in

Telephone* please input your telephone number

Password* please input your password

has not registered?

Log in

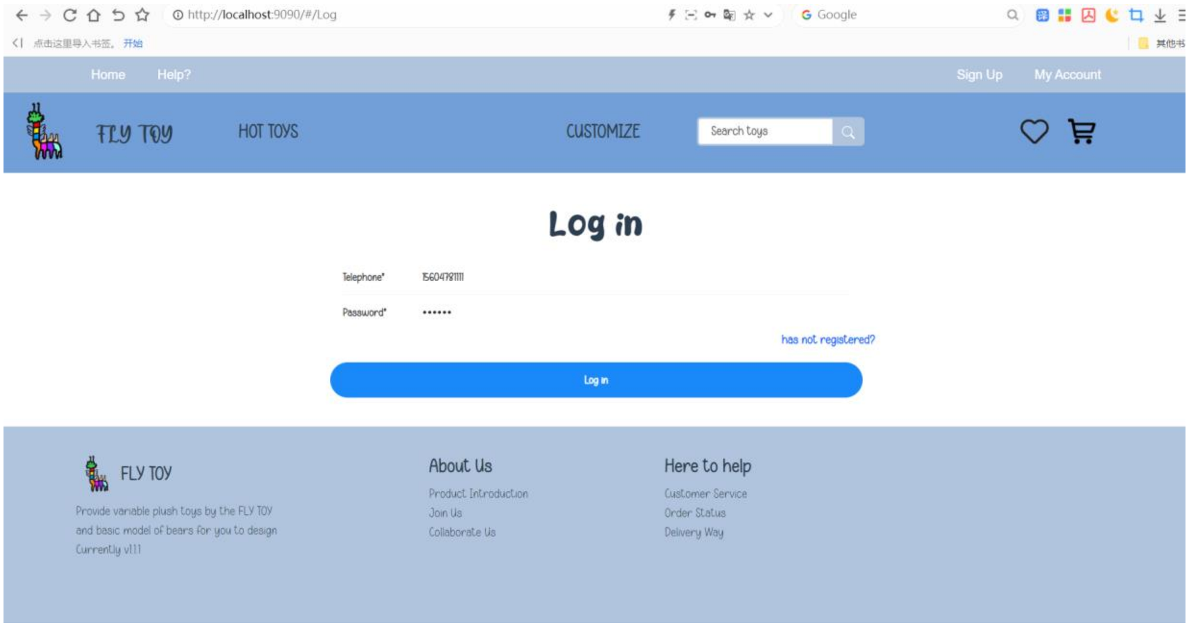
FLY TOY

Provide variable plush toys by the FLY TOY and basic model of bears for you to design
Currently v1.1.1

About Us
Product Introduction
Join Us
Collaborate Us

Here to help
Customer Service
Order Status
Delivery Way

Figure 62 The User Interface of Logging in



Log in

Telephone* 15604781111

Password* ••••••

has not registered?

Log in

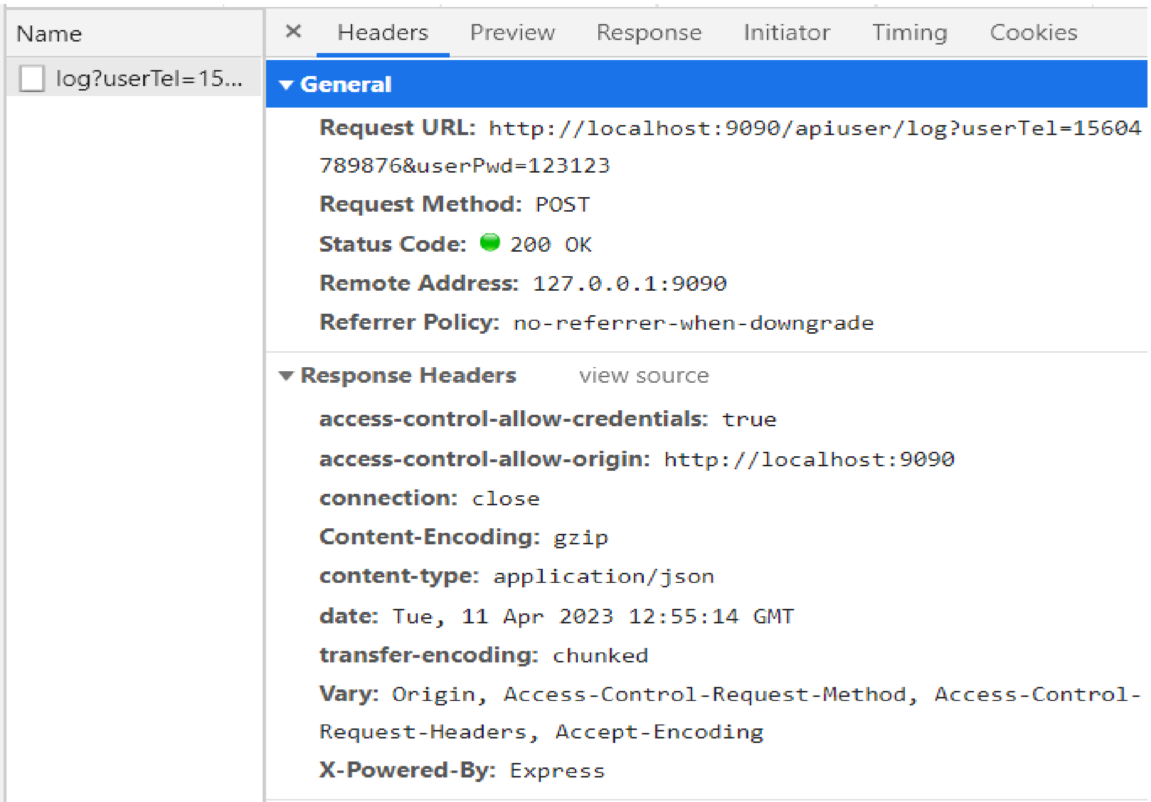
FLY TOY

Provide variable plush toys by the FLY TOY and basic model of bears for you to design
Currently v1.1.1

About Us
Product Introduction
Join Us
Collaborate Us

Here to help
Customer Service
Order Status
Delivery Way

Figure 63 The result of the correct format of the password



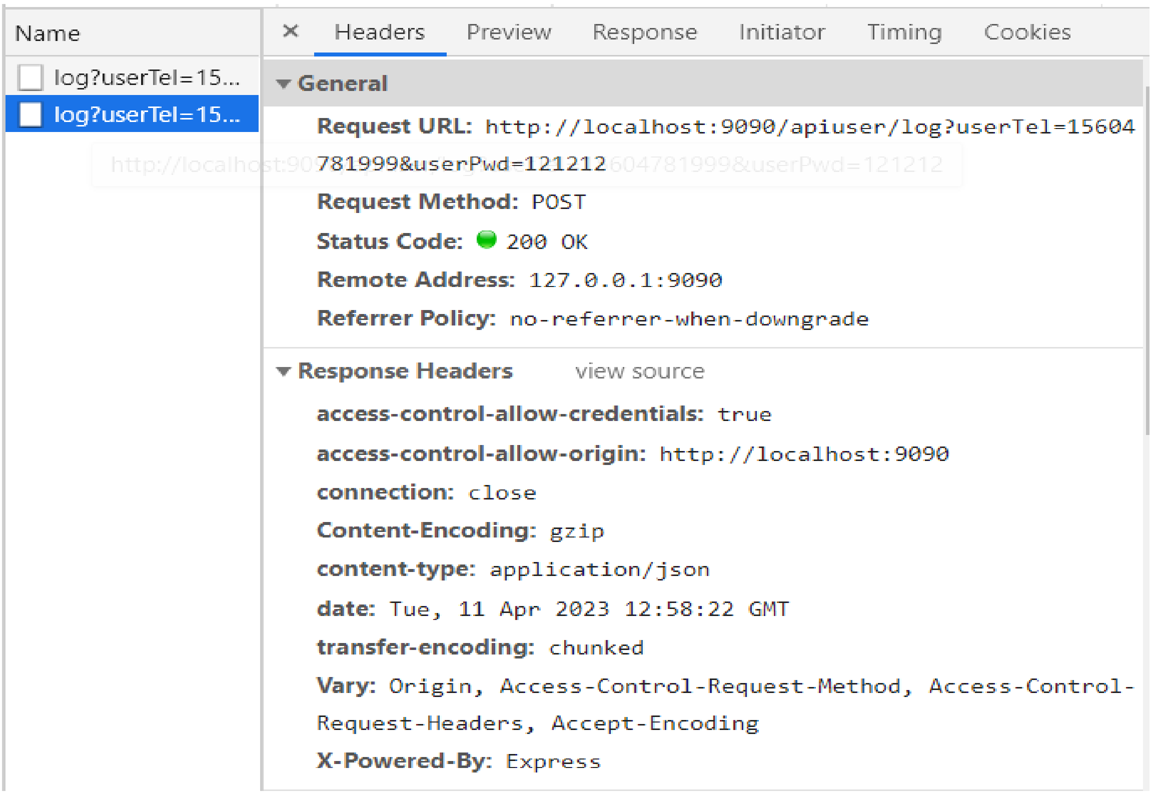
General

Request URL: http://localhost:9090/apiuser/log?userTel=15604789876&userPwd=123123
Request Method: POST
Status Code: ● 200 OK
Remote Address: 127.0.0.1:9090
Referrer Policy: no-referrer-when-downgrade

Response Headers

access-control-allow-credentials: true
access-control-allow-origin: http://localhost:9090
connection: close
Content-Encoding: gzip
content-type: application/json
date: Tue, 11 Apr 2023 12:55:14 GMT
transfer-encoding: chunked
Vary: Origin, Access-Control-Request-Method, Access-Control-Request-Headers, Accept-Encoding
X-Powered-By: Express

Figure 64 The result of using an unregistered telephone number to log in



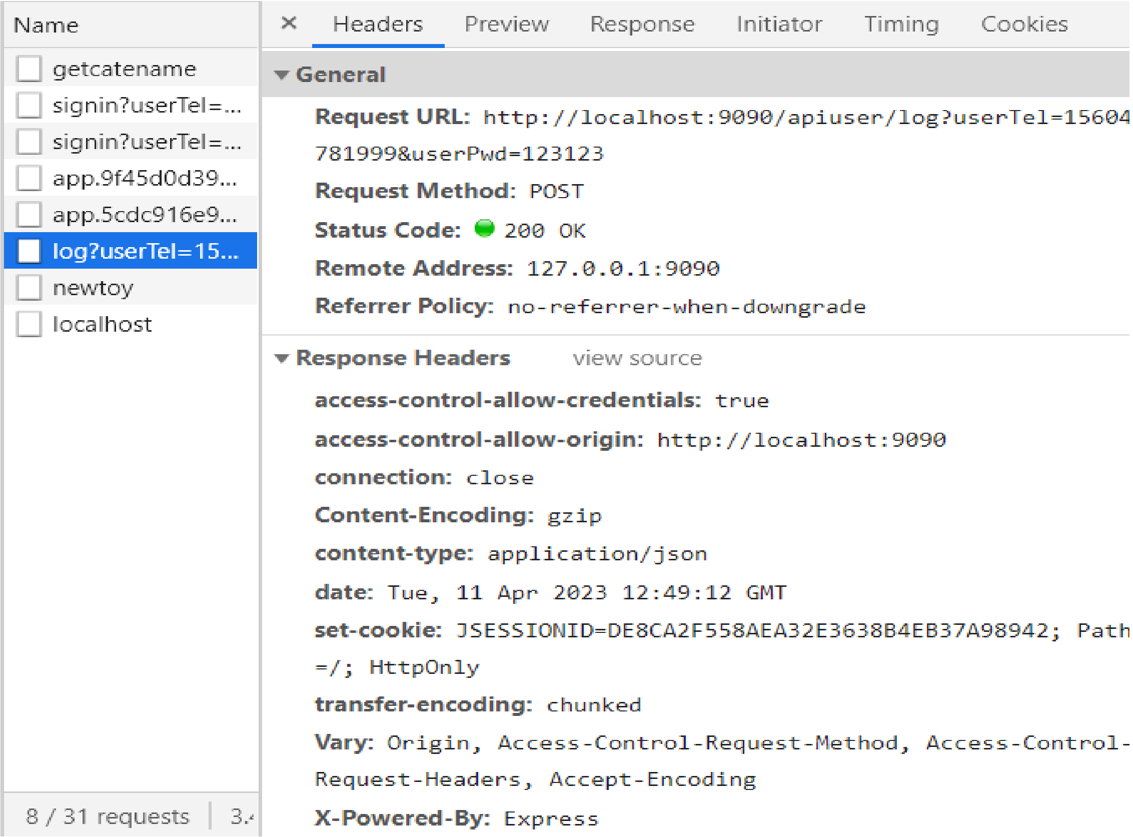
General

Request URL: http://localhost:9090/apiuser/log?userTel=15604781999&userPwd=121212
Request Method: POST
Status Code: ● 200 OK
Remote Address: 127.0.0.1:9090
Referrer Policy: no-referrer-when-downgrade

Response Headers

access-control-allow-credentials: true
access-control-allow-origin: http://localhost:9090
connection: close
Content-Encoding: gzip
content-type: application/json
date: Tue, 11 Apr 2023 12:58:22 GMT
transfer-encoding: chunked
Vary: Origin, Access-Control-Request-Method, Access-Control-Request-Headers, Accept-Encoding
X-Powered-By: Express

Figure 65 The result of inputting an error password to log in



```
▼ General
    Request URL: http://localhost:9090/apiuser/log?userTel=15604
    781999&userPwd=123123
    Request Method: POST
    Status Code: ● 200 OK
    Remote Address: 127.0.0.1:9090
    Referrer Policy: no-referrer-when-downgrade
▼ Response Headers      view source
    access-control-allow-credentials: true
    access-control-allow-origin: http://localhost:9090
    connection: close
    Content-Encoding: gzip
    content-type: application/json
    date: Tue, 11 Apr 2023 12:49:12 GMT
    set-cookie: JSESSIONID=DE8CA2F558AEA32E3638B4EB37A98942; Path
    =/; HttpOnly
    transfer-encoding: chunked
    Vary: Origin, Access-Control-Request-Method, Access-Control-
    Request-Headers, Accept-Encoding
    X-Powered-By: Express
```

Figure 66 The result of logging in successfully

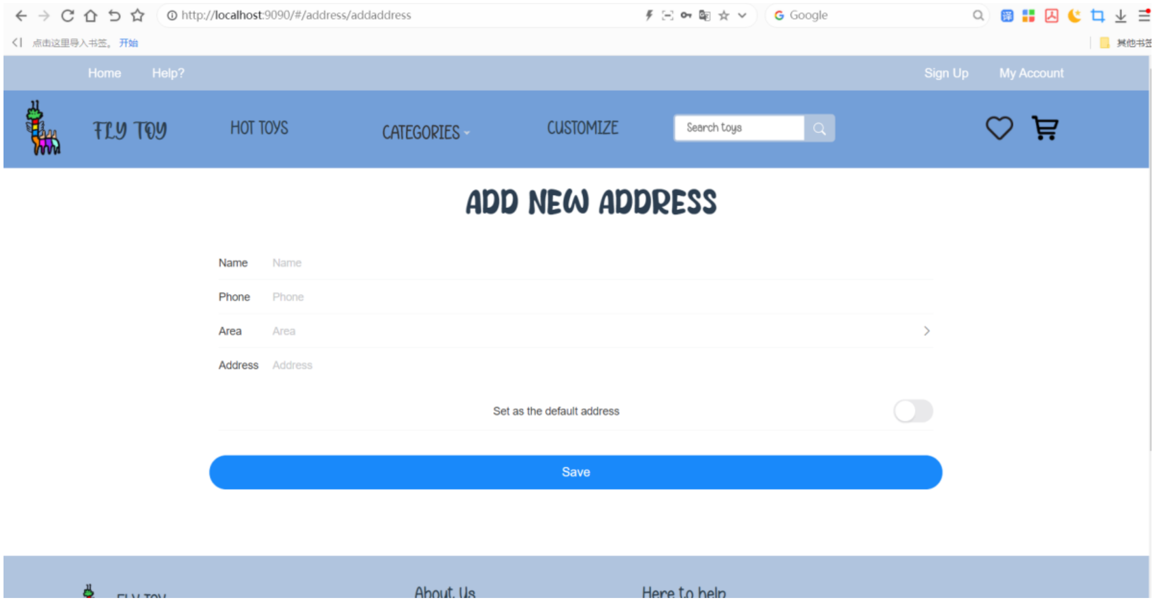
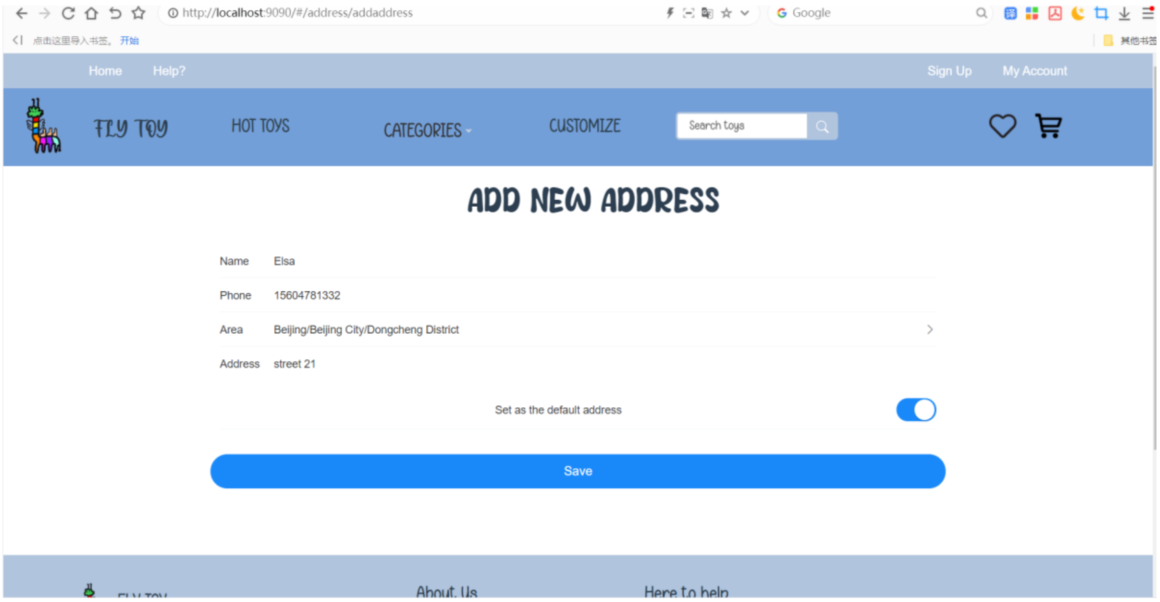


Figure 67 The User Interface of adding the new address



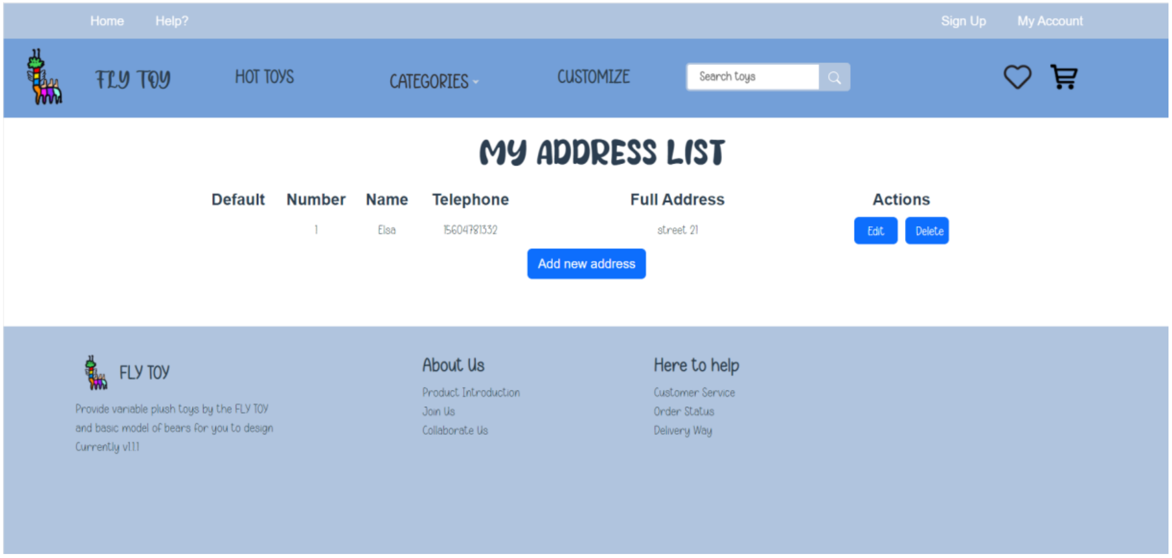
ADD NEW ADDRESS

Name	Elsa
Phone	15604781332
Area	Beijing/Beijing City/Dongcheng District
Address	street 21

Set as the default address ●

Save

Figure 68 The result of setting this new address as the default address



MY ADDRESS LIST

Default	Number	Name	Telephone	Full Address	Actions
	1	Elsa	15604781332	street 21	Edit Delete

Add new address

FLY TOY

Provide variable plush toys by the FLY TOY
and basic model of bears for you to design
Currently v1.1.1

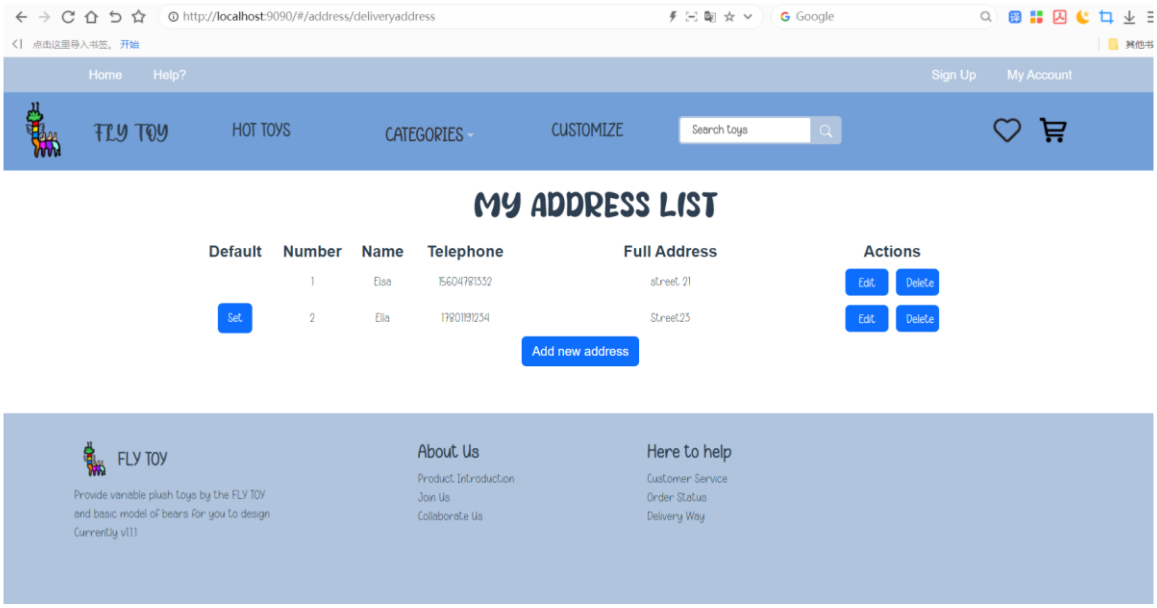
About Us

Product Introduction
Join Us
Collaborate Us

Here to help

Customer Service
Order Status
Delivery Way

Figure 69 The result of adding a new address successfully



MY ADDRESS LIST

Default	Number	Name	Telephone	Full Address	Actions
	1	Elsa	15604781332	street 21	Edit Delete
Set	2	Ella	17801191234	Street23	Edit Delete

FLY TOY

Provide variable plush toys by the FLY TOY and basic model of bears for you to design

Currently v1.1.1

Figure 70 The result of adding a new address and set as the default address successfully