Interface Specification

**Revision History:**

|  |  |  |
| --- | --- | --- |
| Date | Author | Description |
| 2020-09-09 | 王历昊 | Algorithm part |
| 2020-09-12 | 李风鹏 | Crawl part |
| 2020-09-12 | 冯昊天 | Web-Front part(index.html) |
| 2020-09-12 | 冯昊天 | Back-end part(index.html) |
| 2020-09-12 | 曹雪娇 | Web-Front part(rating.html) |
| 2020-09-12 | 曹雪娇 | Back-end part(rating.html) |
| 2020-09-12 | 朱晨 | Web-Front part(recom.html) |
| 2020-09-12 | 朱晨 | Web-end part(recom.html) |

目录

[Interface Specification 1](#_Toc50843059)

[Interface Specification 3](#_Toc50843060)

[<algorithm > 3](#_Toc50843061)

[1 Introduction 3](#_Toc50843062)

[2 Services 3](#_Toc50843063)

[2.1 Services Provided 3](#_Toc50843064)

[2.2 Access Method 3](#_Toc50843065)

[2.3 Access Method Effects 4](#_Toc50843066)

[3 Test Cases 4](#_Toc50843067)

[3.1.1 T1 4](#_Toc50843068)

[3.1.2 T2 5](#_Toc50843069)

[Interface Specification 6](#_Toc50843070)

[<front > 6](#_Toc50843071)

[1. Introduction 6](#_Toc50843072)

[2. Services 6](#_Toc50843073)

[2.1Services Provided 6](#_Toc50843074)

[2.2Access Method 6](#_Toc50843075)

[3. Test Cases 7](#_Toc50843076)

[3.1.1 T1(fht)：User Login 7](#_Toc50843077)

[3.1.2 T2(fht)：User Login fail 7](#_Toc50843078)

[3.1.3 T3(fht): User get uid 8](#_Toc50843079)

[3.1.4 T4(fht): Page redirect to rating.html 8](#_Toc50843080)

[3.1.5 T5(cxj):Search books 8](#_Toc50843081)

[3.1.6 T6(cxj): Rate Books 8](#_Toc50843082)

[3.1.7 T7(cxj): Get Recommendation 9](#_Toc50843083)

[3.1.8 T8(cxj):Logout 9](#_Toc50843084)

[3.1.9 T9 : Show result 9](#_Toc50843085)

[3.1.10 T10: Show other recommended books 9](#_Toc50843086)

[Interface Specification 10](#_Toc50843087)

[< back-end > 10](#_Toc50843088)

[1. Introduction 10](#_Toc50843089)

[2 Services 10](#_Toc50843090)

[2.1 Services Provided 10](#_Toc50843091)

[2.2 Access Method 10](#_Toc50843092)

[2.3 Access Method Effects 11](#_Toc50843093)

[3 Test Cases 11](#_Toc50843094)

[3.1.1 T1：User view index page 11](#_Toc50843095)

[3.1.2 T2：User login 11](#_Toc50843096)

[3.1.3 T3：User login failed 12](#_Toc50843097)

[3.1.4 T4：User get uid 12](#_Toc50843098)

[3.1.5 T5：User rates books 12](#_Toc50843099)

[3.1.6 T6：User logs out 12](#_Toc50843100)

[3.1.7 T7：User get the information of the book 13](#_Toc50843101)

[3.1.8 T8：User get the information of other books 13](#_Toc50843102)

[Interface Specification 14](#_Toc50843103)

[<crawl> 14](#_Toc50843104)

[1 Introduction 14](#_Toc50843105)

[2 Services 14](#_Toc50843106)

[2.1 Services Provided 14](#_Toc50843107)

[2.2 Access Method 14](#_Toc50843108)

[3 Test Cases 14](#_Toc50843109)

[3.1.1 T1：Get data 14](#_Toc50843110)

Interface Specification

<algorithm >

# Introduction

This module is to implement the algorithm of recommender system. The details would presented in the following.

# Services

## Services Provided

|  |  |  |  |
| --- | --- | --- | --- |
|  | Service | Provided By | Tested By |
| 1 | The user gets a recommendation list of books. | recommend | T1 |
| 2 | The user updates/re-trains the existing model. | model\_fit | T2 |

## Access Method

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Access Method** | **Parameter name** | **Parameter type** | **Description** | **Exceptions** | **Map to services** |
| recommend | user\_id, items\_info, users\_info, ratings\_info, model | int, table, table, table, pickle | user\_id is the id of the user who is using the system;  items, users, and ratings are three tables in the database; model is a pickle file which represents the existing model. | There is no existing model. | 1 |
| model\_fit | items\_info, users\_info, ratings\_info | table, table, table | items, users, and ratings are three tables in the database; |  | 2 |

## Access Method Effects

|  |  |
| --- | --- |
| **Access Method** | **Description** |
| recommend | The user can get a list of recommendation list of books via this method. The default size of the result list is 4. The information in this list includes the ISBN, title, publisher, publish year, author and the url of the book cover. |
| model\_fit | The user can update/re-train the existing model when necessary.(e.g. The database was updated.) The new model would take place of the origin model. |

# Test Cases

<Characterize the expected value of the outputs over sets calls to the module.>

### T1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Description** | **Input Type/Value** | **Expected Results** | **Service** | **Preamble** |
| 1 | recommend | user\_id, items\_info, users\_info, ratings\_info | A list of top 4 recommendation books. |  |  |

### T2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Description** | **Input Type/Value** | **Expected Results** | **Service** | **Preamble** |
| 1 | model\_fit | items\_info, users\_info, ratings\_info | A new model to be saved in place of origin model. |  |  |

Interface Specification

<front >

# Introduction

Front-end contains several pages that will be shown to the user. Besides the basic html on templates, other techniques like ajax and java-scripts are also added into the web page. With making request and getting response from the rear-end, the front-end are enabled to provide various information to the frontier.

# Services

## 2.1Services Provided

|  |  |  |
| --- | --- | --- |
| **Service** | **Provided By** | **Tested By** |
| 1. User login | Index.html | T1 (fht) |
| 1. User login fail | Index.html | T2 (fht) |
| 1. Get user Id | Index.html | T3 (fht) |
| 1. Page redirect to rating.html | Index.html | T4(fht) |
| 1. Search books | rating.html | T5(cxj) |
| 1. Rate books | rating.html | T6(cxj) |
| 1. Get recommendation | rating.html | T7(cxj) |
| 1. Logout | rating.html | T8(cxj) |
| 1. Show result | recom-1.html | T9(zc) |
| 1. Show other recommended books | recom-2.html | T11(zc) |

## 2.2Access Method

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Access Method** | **Parameter name** | **Parameter type** | **Description** | **Exceptions** | **Map to services** |
| Ajax | None | None |  |  | Service 1 |
| Ajax.success | rs | dict | Add content to html |  | Service 1 |
| Ajax | Uid  ajid | Str  int |  | No uid matched in the Database | Service 2 |
| Ajax.success | rs | dict | Add content to html |  | Service 2 |
| Ajax.error | None | None | Add content to html |  | Service 3 |
| getBook() | none | none | Send user’s input(the isbn) to backend and search results from database. Then show results on page. |  | Service 5,6 |
| postRating() | none | none | Store user’s id, isbn of the book and its rating score. |  | Service 7 |
| logout() | none | none | User logs out and page will turn to the first page before login |  | Service 8 |

# Test Cases

### 3.1.1 T1(fht)：User Login

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Description** | **Input Type/Value** | **Expected Results** | **Service** | **Preamble** |
| 1 | User click ‘yes’ button | none | Upper right block light up | none | None |
| 2 | User text uid and click ‘submit’ button | uid | Page redirect to “rating.html”  &  The block shows “login fail” | *Service 2*  *Service 3*  *Service 4* | Step 1 |

### 3.1.2 T2(fht)：User Login fail

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Description** | **Input Type/Value** | **Expected Results** | **Service** | **Preamble** |
| 1 | User click ‘yes’ button | none | Upper right block light up | none | None |
| 2 | User text uid and click ‘submit’ button | uid | The block shows “login fail” | *Service 2*  *Service 3* | Step 1 |

### T3(fht): User get uid

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Description** | **Input Type/Value** | **Expected Results** | **Service** | **Preamble** |
| 1 | User click ‘no’ button | none | Bottom left block light up | none | None |
| 2 | User click ‘get uid’ button | none | The block shows “Your ID is:” with 7 digit of number | Service 1 | Step 1 |

### T4(fht): Page redirect to rating.html

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Description** | **Input Type/Value** | **Expected Results** | **Service** | **Preamble** |
| 1 | User click ‘submit’ button | none | Page Jump to rating,html | none |  |

### T5(cxj):Search books

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Description** | **Input Type/Value** | **Expected Results** | **Service** | **Preamble** |
| 1 | User click ‘Rate Books” button | none | The part of “SEARCH BY KEY” becomes opaque | none |  |
| 2 | User input the isbn of one book | string/isbn | The information of book shows on the “SEARCH REASULTS” part | Service 5 |  |

### T6(cxj): Rate Books

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Description** | **Input Type/Value** | **Expected Results** | **Service** | **Preamble** |
| 1 | User submit his rating score | int/rating | Submit successfully | Service 6 |  |
| 2 | The backend store the scores | int/rating | Store successfully | backend |  |

### T7(cxj): Get Recommendation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Description** | **Input Type/Value** | **Expected Results** | **Service** | **Preamble** |
| 1 | User click the “Get Recommendation” button | none | Page jumps to Recom-1.html | Service 7 |  |

### T8(cxj):Logout

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Description** | **Input Type/Value** | **Expected Results** | **Service** | **Preamble** |
| 1 | User click the “Logout” button | none | Page jumps to index.heml | Service 8 |  |

### T9 : Show result

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Description** | **Input Type/Value** | **Expected Results** | **Service** | **Preamble** |
| 1 | The web page obtains and displays the book information from the back-end algorithm | book information | Webpage display picture author press | Service 9 |  |

### T10: Show other recommended books

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Description** | **Input Type/Value** | **Expected Results** | **Service** | **Preamble** |
| 2 | Users click the "see also" button to jump to other recommended lists | None | The website displays the information of three recommended books | Service 10 |  |

Interface Specification

< back-end >

# 1. Introduction

Rear-end is in charge of dealing with the information sent from the front-end, load data and save data to the database, and making response to the front-end. Rear-end binds with the front-end through the config file called “urls.py” in django web-frame.

# Services

## 2.1 Services Provided

|  |  |  |
| --- | --- | --- |
| Service | Provided By | Tested By |
| 1. Render ‘index.html’ | Indexpage.py | T1(fht) |
| 1. User login | Indexpage.py | T2(fht) |
| 1. User get uid | Indexpage.py | T3(fht) |
| 1. User login fail | Indexpage.py | T4(fht) |
| 1. User rates books | rating.py | T5(cxj) |
| 1. User gets recommendation | rating.py | T5(cxj) |
| 1. User logs out | rating.py | T6(cxj) |
| 1. User get the information of the book | recommand.py | T7(zc) |
| 1. User get the information of other books | recommand.py | T8(zc) |

## Access Method

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Access Method** | **Parameter name** | **Parameter type** | **Description** | **Exceptions** | **Map to services** |
| index | request | HttpRequest |  |  | Service 1 |
| Login | request | HttpRequest |  |  | Service 2 |
| Match | Id | Str |  |  | Service 2 |
| Getid | request | HttpRequest |  |  | Service 3 |
| Match | Id | Str |  |  | Service 3 |
| Add\_to\_uinfo | uid | str |  |  | Service 3 |
| getRatingPage | request | HttpRequest |  |  | Service 5 |
| getRatingBooks | request | HttpRequest |  |  | Service 5 |
| getBookInfo | isbn | str |  |  | Service 5 |
| saveRating | request | HttpRequest |  |  | Service 5 |
| logout | request | HttpRequest |  |  | Service 7 |
| recomBook | request | HttpRequest |  |  | Service 8 |
|  |  |  |  |  |  |

## Access Method Effects

|  |  |
| --- | --- |
| **Access Method** | **Description** |
| Index | Render “index.html” |
| Login | Check if the user’s uid is valid |
| Match | Check if the input uid is in the database and return check result |
| Getid | Rand a 7digit number as a new uid and return |
| Add\_to\_uinfo | Add the input uid to the database |
| getRatingPage | Render “rating.html” |
| getRatingBooks | Return JsonREsponse(book) |
| getBookInfo | Get information of book from database |
| saveRating | Store the use\_id ,rating, isbn to one csv file |
| logout | Render “index.html” |
| recomBook | Render “remo-1.html” or“remo-2.html” |

# Test Cases

### T1：User view index page

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Description** | **Input Type/Value** | **Expected Results** | **Service** | **Preamble** |
| 1 | Index() got request | request | Show index.html | Service 1 | None |

### T2：User login

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Description** | **Input Type/Value** | **Expected Results** | **Service** | **Preamble** |
| 1 | Login() got request | request | Return JsonResponse | Service 2 | T1 |
| 2 | Match() activated | uid | Return true | Service 2 | T1 |

### T3：User login failed

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Description** | **Input Type/Value** | **Expected Results** | **Service** | **Preamble** |
| 1 | Login() got request | request | Return JsonResponse | Service 4 | T1 |
| 2 | Match() activated | uid | Return false | Service 4 | T1 |

### T4：User get uid

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Description** | **Input Type/Value** | **Expected Results** | **Service** | **Preamble** |
| 1 | getid() got request | request | Return JsonResponse | Service 3 | T1 |
| 2 | Match() activated | uid | Return false | Service 3 | T1 |
| 3 | Add\_to\_info() activated | Uid | None | Service 3 | T1 |

### T5：User rates books

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Description** | **Input Type/Value** | **Expected Results** | **Service** | **Preamble** |
| 1 | getBookInfo() gets information | isbn | Return rs | Service 5 |  |
| 2 | getRatingBooks() return the book information | request | Return JsonResponse | Service 5 |  |
| 3 | saveRating() saves the rating,isbn,user\_id to a file | request | Return JsonResponse | Service 5 |  |

### T6：User logs out

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Description** | **Input Type/Value** | **Expected Results** | **Service** | **Preamble** |
| 1 | logout() makes page turn to index.html | request | Return JsonResponse | Service 7 |  |

### T7：User get the information of the book

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Description** | **Input Type/Value** | **Expected Results** | **Service** | **Preamble** |
| 1 | request.session | none | uid | Service 8 |  |
| 2 | recommend | uid | books | Service 8 | Step1 |
| 3 | render | request | Render Recom1 | Service 8 | Step2 |

### T8：User get the information of other books

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Description** | **Input Type/Value** | **Expected Results** | **Service** | **Preamble** |
| 1 | request.session | none | uid | Service 9 |  |
| 2 | recommend | uid | books | Service 9 | Step1 |
| 3 | render | request | Render Recom1 | Service 9 | Step2 |

Interface Specification

<crawl>

# 1 Introduction

Crawl the required information and clean the data

# Services

## 2.1 Services Provided

|  |  |  |
| --- | --- | --- |
| **Service** | **Provided By** | **Tested By** |
| Get data | items\_wo\_duplicates.csv | Lfp |

## Access Method

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Access Method** | **Parameter name** | **Parameter type** | **Description** | **Exceptions** | **Map to services** |
| Store data | Data | Csv | Data | The data was saved successfully | Service 1 |

# Test Cases

### T1：Get data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Description** | **Input Type/Value** | **Expected Results** | **Service** | **Preamble** |
| 1 | View data | none | Normal specification data | none | None |