**CS408 Test Plan**

**Team Member: Nanxin Jin, Weijie Meng, Yang Shi, Jintao Zeng, Yaoxi Liang**

**Functional Requirements:**

|  |  |
| --- | --- |
| Backlog ID | Functional Requirements |
| 1 | As a user, I would like to have a page to sign up my account with email.   * 1.1 Signup user, Severity = 1   + Expected input:     - Enter email and password in sign up page   + Expected output:     - Server receive email and password * 1.2 Create valid user, Severity = 1   + Expected input:     - Create user with email and password in server   + Expected output:     - User created in database and the signup page goes to main page in frontend * 1.3 Create invalid user, Severity = 1   + Expected input:     - Create user with invalid email and password in server   + Expected output:     - No user was created in database and error message show in front end. |
| 2 | As a user, I would like to have a secure login page with my email and password.   * 2.1 Login valid user, Severity = 1   + Expected input:     - Enter email and password in sign up page   + Expected output:     - Login page goes to main page * 2.2 Login invalid user, Severity = 1   + Expected input:     - Enter invalid email and password in sign up page   + Expected output:     - Error message shows in front end |
| 3 | As a user, I would like to be able to modify my account information.   * 3.1 Render edit account page, Severity = 1   + Expected input:     - Click edit account in main page   + Expected output:     - An edit account page rendered * 3.2 Save valid modification, Severity = 1   + Expected input:     - User modify the account and save   + Expected output:     - Server update user information in database and front end receive success message * 3.3 Save invalid modification, Severity = 1   + Expected input:     - User modify the account with invalid information (invalid email etc.) and save   + Expected output:     - User’s information in database stays unchanged and front end receive error message |
| 4 | As a user, I would like to stay at log in stage for a certain of time, even after I close the website.   * 4.1 Login user’s account, Severity = 1   + Expected input:     - Login with valid email and password   + Expected output:     - Token will be received in frontend * 4.2 Reopen website, Severity = 1   + Expected input:     - Login user account     - Close webpage and reopen webpage   + Expected output:     - User stays login in webpage |
| 5 | As a user, I would like to have a button to allow me to log out my account.   * 5.1 Logout user account, Severity = 1   + Expected input:     - User clicks logout button in main page   + Expected output:     - Web goes to login page * 5.2 Reopen website, Severity = 1   + Expected input:     - User clicks logout button in main page     - User close and reopen website   + Expected output:     - Login page will show |
| 6 | As a user, I would like to find the information of the stuff that people are willing to sell.   * 6.1 Render page, Severity = 2   + Expected input:     - User login and goes to main page   + Expected output:     - A page with listed used stuff showing in main page |
| 7 | As a user, I would like to be able to post used stuff.   * 7.1 Purchase books from cart, Severity = 1   + Expected input:     - From the cart screen as a logged in user, user clicks purchase button     - User enters payment information     - User clicks purchase     - User checks their email   + Expected output:     - User has a confirmation email with purchase number and amount * 7.2 Purchase books from empty cart, Severity = 1   + Expected input:     - From the cart screen as a logged in user, user clicks purchase button   + Expected output:     - Display error message and render cart screen |
| 8 | As a user, I would like to be able to modify the used stuff I posted.   * 8.1 Re-render page, Severity = 1   + Expected input:     - a user makes modifications to the post he/she posted   + Expected output:     - A re-rendered, edited post |
| 9 | As a user, I would like to be able to view all the stuff posted.   * 9.1 Stuff view, Severity = 1   + Expected input:     - User clicks view stuff button   + Expected output:     - Web goes to stuff category page * 9.2 Reopen stuff page after adding stuff, Severity = 1   + Expected input:     - User clicks view stuff button     - User clicks back and add a stuff then reopen the stuff category page   + Expected output:     - The new stuff should be in the stuff list |
| 10 | As a user, I would like to be able to delete the post I made.   * 10.1 Render page, Severity = 2   + Expected input:     - User clicks the sorting by price option   + Expected output:     - A new list of goods sorted by price from low to high or high to low |
| 11 | As a user, I would like to be able to sort the used stuff with price.   * 11.1 Re-Render page, Severity = 2   + Expected input:     - User clicks the sorting by price option   + Expected output:     - A new list of goods sorted by price from low to high or high to low |
| 12 | As a user, I would like to check the stock of my posted stuff in my account.   * 12.1 Romve one stuff from stock and check the number of that stuff, Severity = 2   + Expected input:     - User choose one of his or her stuff and click on revove and reload the page   + Expected output:     - The number of that particular stuff will decrease by one |
| 13 | As a user, I would like to chat with my buyers to advertise my merchandise, if I am a seller.   * 13.1 Start online chatting with my buyer if he or she is interested in my stuff and wants to know more about it, Severity = 1   + Expected input:     - User click on chat button to start the chatting room   + Expected output:     - The chatting room window appears |
| 14 | As a user, I would like to chat with my sellers to bid the price, if I am a buyer.   * 14.1 Start online chatting with my seller to ask for information and to bid the price, Severity = 1   + Expected input:     - User click on chat button to start the chatting room   + Expected output:     - The chatting room window appears |
| 15 | As a user, I would like to search items by price range.   * 15.1 Search desire items by price range, Severity = 1   + Expected input:     - User click on search and choose items by price range   + Expected output:     - The items will be found and displayed in the price range given by the user |
| 16 | As a user, I would like to search items by categories.   * 16.1 Search desire items by categories, Severity = 1   + Expected input:     - User click on search and choose items by categories   + Expected output:     - The items will be found and displayed in the category given by the user |
| 17 | As a user, I would like to save items to wishlist, so that I can check them later.   * 17.1 Add items in the wishlist and check, Severity = 1   + Expected input:     - User click on add to wishlist button and then go back to reopen the wishlist   + Expected output:     - The items added by the user will be found and displayed in the to the user * 17.2. Update the wishlist after successful transction, Severity = 1   + Expected input:     - User buy a item in the wishlist and then go back to reopen the widhlist   + Expected output:     - The item that was just bought by the user should be extracted from the wishlist and put into the history |
| 18 | As a user, I would like to search items by conditions.   * 18.1 Search for items that exist, Severity = 1   + Expected input:     - User types in keywords and click search   + Expected output:     - System returns a list of items that match the key words * 18.2. Search for items that do not exist , Severity = 1   + Expected input:     - User types types in keywords that do not match anything and click search   + Expected output:     - System returns an error message saying the no items found matched. |
| 19 | As a user, I would like to have my order list.   * 19.1 Users place some orders, Severity = 1   + Expected input:     - User clicks on ‘order list’   + Expected output:     - System returns a list of orders that the users placed * 19.2. Users have not placed any orders, Severity = 1   + Expected input:     - User clicks on ‘order list’ of an empty list   + Expected output:     - System returns a message saying order list is empty |
| 20 | As a user, I want to see the rating and comment of buyers.   * 20.1 Display rating and comment about a buyer, Severity = 2   + Expected input:     - User click into a buyer   + Expected output:     - Informations like ratings and comments of that buyer are displayed. |
| 21 | As a user, I would like to rate the buyer after transaction.   * 21.1 Rating option after buying, Severity = 2   + Expected input:     - Users buys and recieves an item   + Expected output:     - A rating option pops out for buyer to rate the seller |
| 22 | As a user, I want to see the rating and comment of sellers.   * 22.1 Display rating and comment about a buyer, Severity = 2   + Expected input:     - User click into another user   + Expected output:     - Informations like ratings and comments of that user are displayed. |
| 23 | As a user, I would like to rate the seller after transaction.   * 23.1 Rating option after buying, Severity = 2   + Expected input:     - Users sells and the buyer recieves an item   + Expected output:     - A rating option pops out for seller to rate the buyer |
| 24 | As a user, I want buyers know when the item is available for sell.   * 24.1 Render available time, Severity = 2   + Expected input:     - User open main page   + Expected output:     - Available time showing on each of the market item. |
| 25 | As a developer, I would like to receive user feedback.   * 25.1 Click send feedback button, Severity = 1   + Expected input:     - Add some feedback and click send feedback button.   + Expected output:     - Server receive the feedback. * 25.2 Create valid feedback, Severity = 2   + Expected input:     - Write few sentances and click send feedback.   + Expected output:     - Feedback goes into database and a “send successful” window popup. * 25.3 Create invalid feedback, Severity = 2   + Expected input:     - Creat a blank feedback and click send feedback button.   + Expected output:     - An “Invalid feedback” window should popup and the blank feedback should not goes into database. |
| 26 | As a user, I would like to be able to upload my own avatar.   * 26.1 Upload a valid picture as avatar, Severity = 1   + Expected input:     - Choose a valid picture(fitted size) and click upload.   + Expected output:     - User’s avatar should be updated and database should be updated. * 26.2 Upload an invalid picture as avatar, Severity = 1   + Expected input:     - Choose an invalid picture(wrong format, too large) and click upload.   + Expected output:     - User should get a popup window that contains “Invalid Picture” message. |
| 27 | As a user, I would like to change my password.   * 27.1 Send a valid user name and request for changing password, Severity = 1   + Expected input:     - A valid username and reuqest for changing password.   + Expected output:     - A link to chang password page should be sent to the email account associated with the account * 27.2 Send a invalid user name and request for changing password, Severity = 1   + Expected input:     - An invalid username and reuqest for changing password.   + Expected output:     - Error message should be displayed. No change in the databse |
| 28 | As a user, I would like to block user sending spams.   * 28.1 Block a user , Severity = 1   + Expected input:     - Enter the valid userName .   + Expected output:     - A confirmation window should be propmted and the related info is saved in the database * 28.2 Block an non-existing user , Severity = 1   + Expected input:     - Enter an userName that hasn’t been created yet.   + Expected output:     - Error message should be displayed.No change in the databse |
| 29 | As a user, I would like to report bad seller / buyer.   * 29.1 Send the bad seller/buyer’s info and related proof, Severity = 1   + Expected input:     - A valid user’s info and screenshot of the proof   + Expected output:     - A message should be displayed to tell the user that the team will start investigation on this. The report info is saved in the database * 29.2 Send an invalid user info , Severity = 1   + Expected input:     - An invalid usernameExpected output:     - Error message should be displayed. No change in the databse |
| 30 | As a user, I would like to give some suggestions for future modifies.   * 30.1 Send a feedback and contact info, Severity = 1   + Expected input:     - Feedback content and contact info   + Expected output:     - A thanks message is displayed and feedback info is saved in the database |