

HACETTEPE UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

BBM 487 SOFTWARE ENGINEERING LAB

TEST CASES

Özdeş Öztürk
21228635

Ekin Kandemir
21228434

Kadir Bulut
21228137



1. Login

1.1 <testLoginWithValidInputsForValidUser>

Description: The user enters his/her username and password that is already registered. Main page for user should be opened

Pre-conditions: The user must click “Log In” button after s/he fills the form.

Post-conditions: Main page for user should be opened.

Data required: Username: valid, Password: valid

1.2 <testLoginWithValidInputsForInvalidUser>

Description: The user enters his/her username and password that is not registered. An error message should be shown.

Pre-conditions: The user must click “Log In” button after s/he fills the form.

Post-conditions: Login page should be opened after showing error message.

Data required: Username: valid, Password: valid

1.3 <testLoginWithUnvalidInputs>

Description: The user enters unvalid characters or NULL as input.

Pre-conditions: The user must click “Log In” button after s/he fills the form.

Post-conditions: Login page for user should be opened.

Data required: Username or Password: invalid

2. Logout

2.1 <testLogout>

Description: When a user wants to log out, login page should be shown after closing connection.

Pre-conditions: The user must be logged in and click “Log out” button.

Post-conditions: Login page should be opened after showing closing connection.

Data required: -

3. Add Book

3.1 <testAddExistingBookWithValidInputs>

Description: The library staff enters properties of the book by filling the form. A success message should be shown.

Pre-conditions: The staff must be logged in and click “Confirm” button.

Post-conditions: The stock number of the book must be increased by 1.

Data required: Title: valid, Author: valid

3.2 <testAddNotExistingBookWithValidInputs>

Description: The library staff enters properties of the book by filling the form. A success message should be shown.

Pre-conditions: The staff must be logged in and click “Confirm” button.

Post-conditions: A new book should be recorded to the database.

Data required: Title: valid, Author: valid

3.3 <testAddBookWithUnvalidInputs>

Description: The library staff enters unvalid characters or NULL as input. An error message shown and the adding book page should be shown.

Pre-conditions: The staff must be logged in and click “Confirm” button.

Post-conditions: Adding book page should be opened after showing an error message

Data required: Title or Author: unvalid

4. Update Book

4.1 <testUpdateBookWithValidInputs>

Description: The staff enters valid inputs by filling the update form. A success message should be shown after saving changes.

Pre-conditions: The staff must be logged in and book needed to be searched & found. Then staff must click on confirm button after filling form.

Post-conditions: Changes should be saved and a success message should be shown.

Data required: Title: Valid, Author: Valid

4.2 <testUpdateBookWithUnValidInputs>

Description: The staff enters unvalid or NULL inputs by filling the update form. An error message should be shown.

Pre-conditions: The staff must be logged in and book needed to be searched & found. Then staff must click on confirm button after filling form.

Post-conditions: An error message should be shown and update form will be shown again.

Data required: Title or Author: Invalid

5. Delete Book

5.1 <testDeleteBook>

Description: The staff clicks delete button for any book s/he wants to delete. A success message should be shown after saving changes.

Pre-conditions: The staff must be logged in and book needed to be searched & found. Then staff must click on delete button.

Post-conditions: An error message should be shown and update form will be shown again.

Data required: -

6. Search Book

6.1 <testSearchExistingBook>

Description: The user enters a book name to search field then click search button. The book should be shown.

Pre-conditions: The user must be logged in and click search button after s/he typed a name for book.

Post-conditions: The book that user wants should be shown with its properties in page.

Data required: title: valid

6.2 <testSearchNonExistingBook>

Description: The user enters a book name to search field then click search button. A message should be shown.

Pre-conditions: The user must be logged in and click search button after s/he typed a name for book.

Post-conditions: A “not found” message should be shown.

Data required: title: valid

6.3 <testSearchBookWithUnvalidInput>

Description: User enters invalid or NULL as input. An error message should be shown.

Pre-conditions: The user must be logged in and click search button after s/he typed an input for book name.

Post-conditions: An error message should be shown.

Data required: title: unvalid

7. Borrow Book

7.1 <testBorrowAvailableBook>

Description: The staff clicks borrow button for an existing book and confirms borrowing. A success message should be shown.

Pre-conditions: The staff must be logged in and click borrow button.

Post-conditions: A success message should be shown.

Data required: -

7.2 <testBorrowUnavailableBook>

Description: The staff clicks borrow button for an existing book and confirms borrowing. An error message should be shown.

Pre-conditions: The staff must be logged in and click borrow button.

Post-conditions: An error message should be shown.

Data required: -

8. Return Book

8.1 <testReturnBorrowedBook>

Description: The staff clicks return button for a borrowed book. A success message should be shown.

Pre-conditions: The staff must be logged in and click return button.

Post-conditions: A success message should be shown and stock size of the book should be incremented.

Data required: -

8.2 <testReturnUnborrowedBook>

Description: The staff clicks return button for an unborrowed book. An error message should be shown.

Pre-conditions: The staff must be logged in and click borrow button.

Post-conditions: An error message should be shown.

Data required: -

9. Add Member

9.1 <testAddNewMember>

Description: The staff enters the informations of the new member and clicks add button. A success message should be shown.

Pre-conditions: The staff must be logged in.

Post-conditions: A new member(user) should be created and saved in the database. Then a success message should be shown.

Data required: username: valid, password: valid, firstname: valid, lastname: valid, email: valid, phone: valid

9.2 <testAddExistingMember>

Description: The staff clicks borrow button for an existing member and confirms borrowing. An error message should be shown.

Pre-conditions: The staff must be logged in.

Post-conditions: An error message should be shown.

Data required: username: valid, password: valid, firstname: valid, lastname: valid, email: valid, phone: valid

10. Delete Member

10.1 <testDeleteExistingMember>

Description: The staff enters a username and clicks delete button for an existing member. A success message should be shown after user is deleted.

Pre-conditions: The staff must be logged in.

Post-conditions: User should be deleted from database and a success message should be shown.

Data required: username: valid

10.2 <testDeleteNonexistingMember>

Description: The staff enters a username and clicks delete button for a nonexisting member. A success message should be shown after user is deleted.

Pre-conditions: The staff must be logged in.

Post-conditions: An error message should be shown.

Data required: username: valid

11. Update Member

11.1 <testUpdateExistingMember>

Description: The staff enters a username and other properties for existing user. The changes should be saved and a success message should be shown.

Pre-conditions: The staff must be logged in.

Post-conditions: The system should update user in database with given properties and a success message should be shown.

Data required: username: valid, password: valid, firstname: valid, lastname: valid, email: valid, phone: valid

11.2 <testUpdateNonexistingMember>

Description: The staff enters a username and other properties for nonexisting user. An success message should be shown.

Pre-conditions: The staff must be logged in.

Post-conditions: An error message should be shown.

Data required: username: valid, password: valid, firstname: valid, lastname: valid, email: valid, phone: valid

12. View Books

12.1 <testViewBooksExisting>

Description: The user clicks "view my books" button and his/her books that s/he borrowed should be listed.

Pre-conditions: The user must be logged in and has at least 1 borrowed books.

Post-conditions: Books borrowed by user should be listed with all of their properties.

Data required: -

12.2 <testViewBooksNonexisting>

Description: The user clicks "view my books" button and a not found message should be listed.

Pre-conditions: The user must be logged in and must not have any borrowed books.

Post-conditions: A not found message should be shown.

Data required: -