1.1.2. Units

Login Authorization

Login is one of the most important functionalities that the product has to work properly. The user will need to provide the username that he or she registered and the password matching to the username. This functionality will guarantee that the account information will be secured properly and not accessible for other users.

Test Cases:

1. Test the username exists. If the username does not exist, the warning will be popped up.
2. Check if the password is matching to the username in the database. A warning will appear if the password is invalid for the user.
3. Users can request forgotten password with their username. An email will be automatically sent to the user’s email to help them make a new password.

Logout

Logging out is to help the user to block their accounts from other users who use the same device. If a user cannot logout, the next person, who uses the same computer, can access to his or her account. This problem can lead to some serious problems such as losing the private information or losing the account.

Test Cases:

1. Make sure the user completely logout of the system. When the new site is opened, the account should not be accessed.
2. The password will not be saved in the browser without noticing from the user.

Sign up

Signup is a functionality for new users to create their new accounts.

Test Cases:

1. The new username should not be the same as the existed usernames.
2. The minimum length of the username should be long as the defaults (from 4 to 16).
3. The spaces should not be used in the username and the password.
4. The password and the confirm password should be the same.

Manage Preferences

A user can provide many different options such as how many courses they want, when they usually want to have classes during a day: mornings, afternoons or evenings to the application to generate preferences for their coming semester. This functionality is the main ability of the application to attract many users. Therefore, it is very important to make sure the application work correctly.

Test Cases:

1. Testing if the number of the classes which is generated in the preferences is the same as the number of classes that the user inputs.
2. Ensure the time of the preference schedules matched to the time the user provide.
3. Testing if the user wants a specific weekday not to have classes, the application shall provide recommended schedules without any classes on that day.

Unit Testing

We have specifically created a more static and detailed database which is is only used for testing in order to test and find the errors and bugs in the application.

Login

The login procedure is one of the actions must be tested carefully because this is the first place users will evaluate the application. Therefore, it is very important to ensure that the login functionality work properly. The file test.blade.php has the method to test if a user log into the application successfully. The username will be looked up in the database. The test will be failed if the username is invalid. Moreover, the password will be checked if it is matched to the username’s password. If this password does not go with the username, the test will display a failed message. Importantly, we also check if the user is an admin or just a normal user like student. If a student try to access into the application by his or her account as an admin, the system will deny and display a failed message.

Manage Preferences

Managing preferences has to be tested to guarantee that the generated preference schedules satisfy demands of the user for their next semester. In the test.blade.php file, we have created some specific cases to test the manage preferences functionality. The generated preferences will be compared to the inputs that the user provided. If the preferences do not satisfy the user’s demands, the test will be failed. Otherwise, the test will pass successfully.

test.blade.php

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>TEST</title>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap. min.css"integrity="sha384-1q8mTJOASx8j1Au+a5WDVnPi2lkFfwwEAa8hDDdjZlpLegxhjVME1fgjWPGm kzs7" crossorigin="anonymous">

<link rel="stylesheet" type="text/css" href="CSS/Custom CSS.css">

<script src="https://cdnjs.cloudflare.com/ajax/libs/react/0.14.6/react-with-addons.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/react/0.14.6/react-dom.js"></script>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<!--<script src="https://cdnjs.cloudflare.com/ajax/libs/react/0.14.6/react-with-addons.js"></script>-->

<script src="https://cdnjs.cloudflare.com/ajax/libs/react/0.13.3/JSXTransformer.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/react-bootstrap/0.28.2/react-bootstrap.js"></script>

<script src="Components/typeahead.bundle.js" type="text/jsx"></script>

<script src="Components/ServerBridge.js" type="text/jsx"></script>

<script src="Components/CookieManager.js" type="text/jsx"></script>

<script src="Components/UIManager.js" type="text/jsx"></script>

<script src="Components/Inputs.js" type="text/jsx"></script>

<script src="Components/Dialogs.js" type="text/jsx"></script>

<script src="Components/Login.js" type="text/jsx"></script>

<script src="Components/Preferences.js" type="text/jsx"></script>

<script src="Components/AccountPage.js" type="text/jsx"></script>

<script src="Components/SchedulePage.js" type="text/jsx"></script>

<script src="Components/AdminPage.js" type="text/jsx"></script>

<script src="Components/CourseSequence.js" type="text/jsx"></script>

</head>

<body>

<div id="pageContent"></div>

<script type="text/jsx">

var serverBridge=realServerBridge;

var response = serverBridge.login('User', 'password', function(data) {

if(data.success=='true'&&data.username=='User'&&data.isAdmin=='false') {

console.log('Test valid user successful');

}

else {

console.log('Test valid user failed');

}

});

response = serverBridge.login('Admin', 'password', function(data) {

if(data.success=='true'&&data.username=='Admin'&&data.isAdmin=='true') {

console.log('Test valid admin successful');

}

else {

console.log('Test valid admin failed');

}

});

response = serverBridge.login('notauser', 'password', function(data) {

if(data.success=='false') {

console.log('Test invalid user successful');

}

else {

console.log('Test invalid user failed');

}

});

var testPref = {"courseLoad":"5", "day":"Monday", "time":"Mornings"};

var response = serverBridge.editPreferences(testPref);

var cookie = cookieManager.addCookie("username", "Jason", 1);

serverBridge.getUserPrefs(function(data){

if (data.courseload == testPref.courseLoad && data.dayoff == testPref.day && data.preferredTime == testPref.time){

console.log('Test valid set preferences successful');

}

else{

console.log('Test valid set preferences failed');

}

});

serverBridge.editPreferences({"courseLoad": "", "day" : "", "time" : ""});

cookieManager.removeCookie("username");

response = serverBridge.register('Jason', '123', 'pass', function(data){

if(data.success == "false"){

console.log("Registration for existing user test succeeded");

}

else{

console.log("Registration for existing user test failed!");

}

});

response = serverBridge.register('BatmanVsSuperman', 'justiceleague', 'password', function(data){

if(data.success == "true"){

console.log("Registration for new user test succeeded");

}

else{

console.log("Registration for new user test failed");

}

});

</script>

</body>

</html>