

**Plantae Pickerus**

Your home plants assistant software



Testing

# Test Plan

## User Interface Functionality Testing

|  |  |
| --- | --- |
|  | addressed Functional Requirement |
| 1. Splash Screen |  |
| 1. Login Screen |  |
| 1. Create New User | 1.1.1 Create new account |
| 1. Validate Username/Password |  |
| 1. Main Menu | 1.3 Manage My House |
| 1. User’s Rooms | 1.3 Manage My House |
| * 1. Show User’s Rooms |  |
| * 1. Add Room | 1.3.1 Add Room |
| * 1. Edit Room |  |
| * 1. Delete Room | 1.3.2 Delete Room |
| 1. User’s Spots |  |
| * 1. Show User’s Spots |  |
| * 1. Add Spot | 1.3.3 Add Spot |
| * 1. Edit Spot |  |
| * 1. Delete Spot | 1.3.4 Delete Spot |
| 1. User’s Plants |  |
| * 1. Show User’s Plants |  |
| * 1. Add Plant |  |
| * 1. Delete Plant |  |
| 1. Logout (from many windows) |  |
| 1. Quit/Exit Application |  |

## Functional Requirements (from Project Specification)

**Note:** Only functionality from Group 1: User Accounts has been implemented in this prototype.  
For more information: Please read the *Note from Project Manager.doc* file.

|  |  |  |
| --- | --- | --- |
| **No** | **Requirements** | **Description** |
| **1** | **Group:** USER ACCOUNTS | |
| **1.1** | Create an Account |  |
| 1.1.1 | *Create new account* | Add a new user account to the system/database (username, password, e-mail address and other personalised data) |
|  |  |  |
| **1.2** | Update an Account | User can update account/personal/contact details |
| 1.2.1 | *Update details* | Allows the user to update details, like e-mail address, change password, add location, etc. |
|  |  |  |
| **1.3** | Manage My House | Allows the user to manage “the house”: the quantity of the rooms and the spots around those rooms |
| 1.3.1 | *Add Room* | Adds new room |
| 1.3.2 | *Delete Room* | Deletes existing room |
| 1.3.3 | *Add Spot* | Adds new spot around the room |
| 1.3.4 | *Delete Spot* | Deletes existing spot from the room |
|  |  |  |
| 2 | **Group:** MAIN (PLANT PICKER) | |
| 2.1 | Find a plant for desired place | User can find a plant for desired place around own house (it requires input from the user: *expectations about the plant*) |
|  |  |  |
| 2.2 | Find a spot for desired plant | User can find a perfect (or suggested) spot for desired plant (requires input from the user:  *name of the plant*) |
|  |  |  |
| 3 | **Group:** PLANTSPEDIA (\*functionality implemented in the next iteration) | |
| 3.1 | Browse Plantspedia | Allows user to browse different plants, its description and search for particular plants.  (This part of the system will be implemented/developed in the next iteration  of the project) |
|  |  |  |

# Test Strategy

## Splash Screen

Splash Screen should appear for about 3 seconds, every time the application starts.  
Splash Screen should have transparent background.

## Login Screen

Login Screen should be displayed after Splash Screen.  
Ensure that the application can access local database file,   
otherwise it should create new database file.

## Create New User

Ensure that the new user has been added to the system/database.

Ensure that the chosen username (of a new user) doesn’t exist in the users’ database.  
Ensure that the username and password can contain only letters and digits - no special characters allowed.

## Validate Username/Password

Ensure that the password matches correct username.  
Ensure that the password’s characters are displayed as dots: “•”

## Main Menu

Main Menu should be displayed with proper buttons that work correctly.

Ensure that the user has access to the Manage My House functionality   
and the information about his: rooms, spots, plants.

## User’s Rooms

Ensure that the view shows *rooms* of logged-in user.  
Ensure that the *room* has been added successfully.  
Ensure that the *room* has been updated (edited and saved) successfully.  
Ensure that the *room* has been deleted successfully.

## User’s Spots

Ensure that the view shows *spots* of logged-in user.  
Ensure that the *spot* has been added successfully.  
Ensure that the *spot* has been updated (edited and saved) successfully.  
Ensure that the *spot* has been deleted successfully.

## User’s Plants

Ensure that the view shows *plants* of logged-in user.  
Ensure that the *plant* has been added successfully.  
Ensure that the *plant* has been deleted successfully.

## Logout (from many windows)

Ensure that Logout button logs out current user and displays Login Screen after.

## Quit/Exit Application (from many windows)

Ensure that the quit option (X button) exits the application.

# Test Cases

## Test Case 1

### Splash Screen

Splash Screen should appear for about 3 seconds, every time the application starts.  
Splash Screen should have transparent background.

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | Expected Results | Actual Results | PASS? |
| Run the application | Splash Screen appears  for about 3 seconds | Splash Screen appears  for at least 2-3 seconds | **Yes** |
| Run the application | Splash Screen have transparent background | Splash Screen have transparent background (only green logo appears) | **Yes** |

### Testing Screenshots:



## Test Case 2

### Login Screen

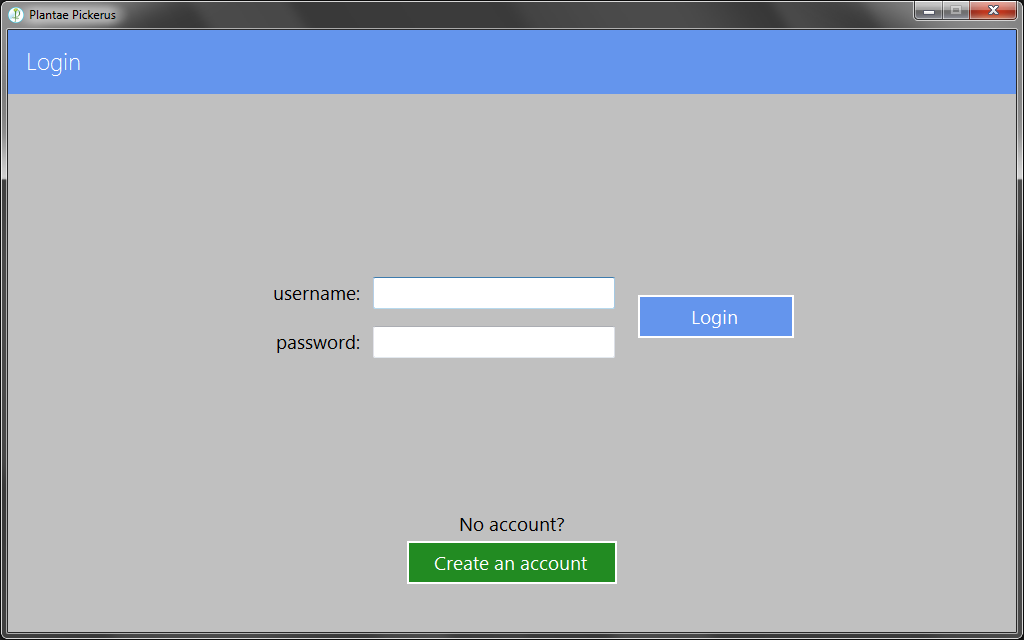
Login Screen should be displayed after Splash Screen.

Ensure that the application can access local database file,   
otherwise it should create new database file.

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | Expected Results | Actual Results | PASS? |
| Run the application | Login Screen displayed,  after the Splash Screen | Login Screen has been displayed, just after the Splash Screen disappeared | **Yes** |
| 1. Delete database file 2. Run the application 3. Check if the file has been created 4. Close the application 5. Run the application 6. Check if a new database file hasn’t been created | The file should be created on the first run. The file shouldn’t be created on the second run. | The file *PPdatabase.db*  has been created correctly on the local hard drive.  After closing and rerunning of the application, the only one file exists with this name and the date of creation shows that it has been created before this run. | **Yes** |

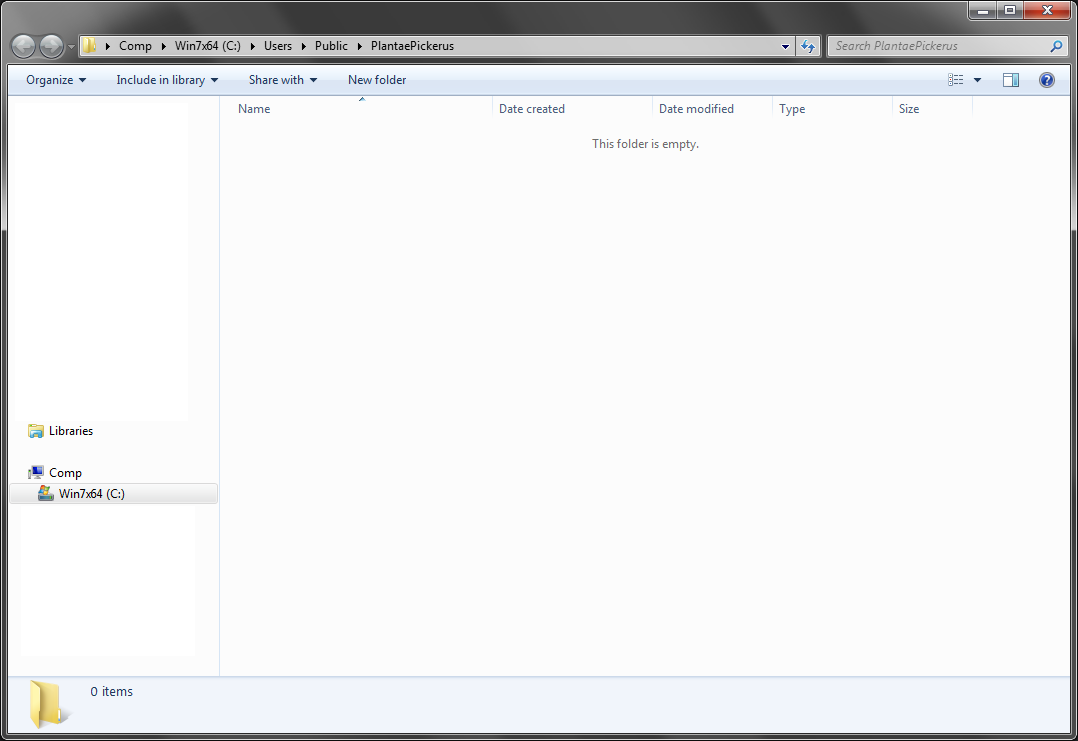
### Testing Screenshots:

Testing: Run the application to check if the Login Screen appears correctly

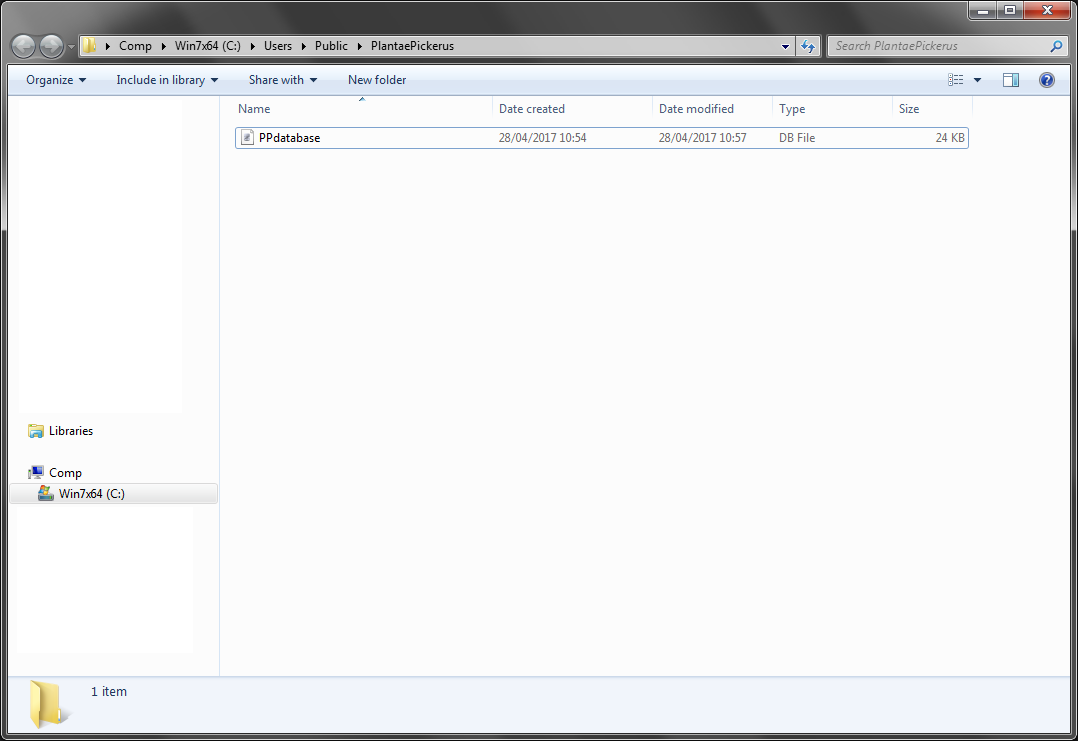


Testing: Database file check/creation

Before first run:



After second run:



## Test Case 3

### Create New User

Ensure that the new user has been added to the system/database.

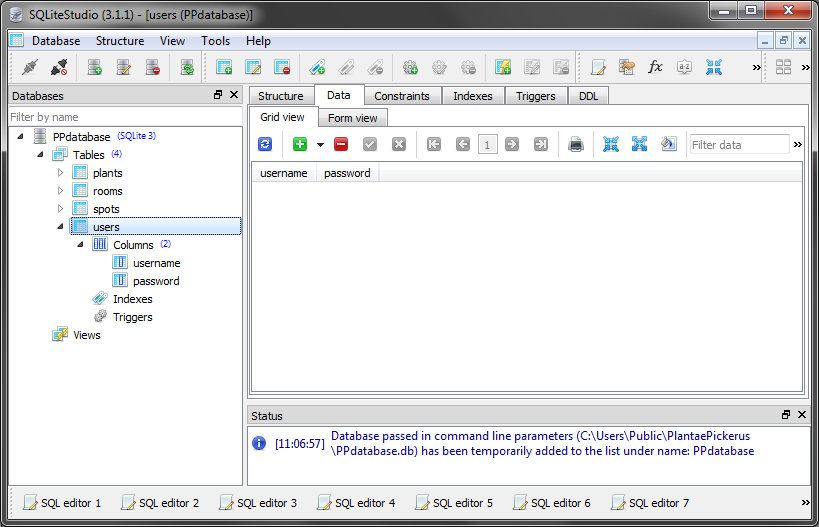
Ensure that the chosen username (of a new user) doesn’t exist in the users’ database.

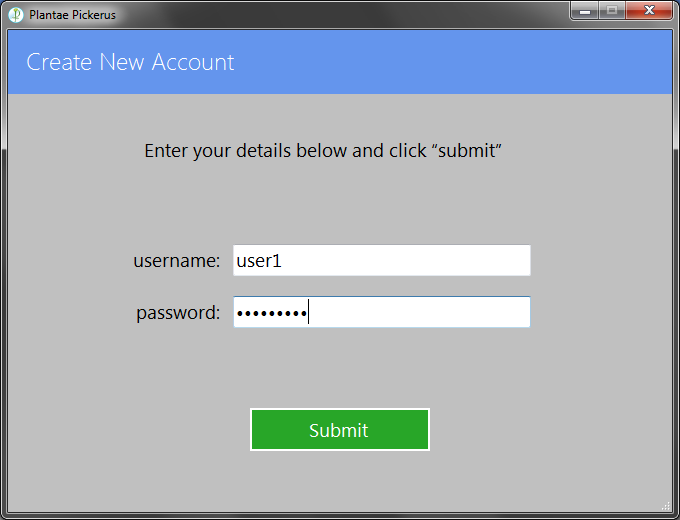
Ensure that the username and password can contain only letters and digits - no special characters allowed.

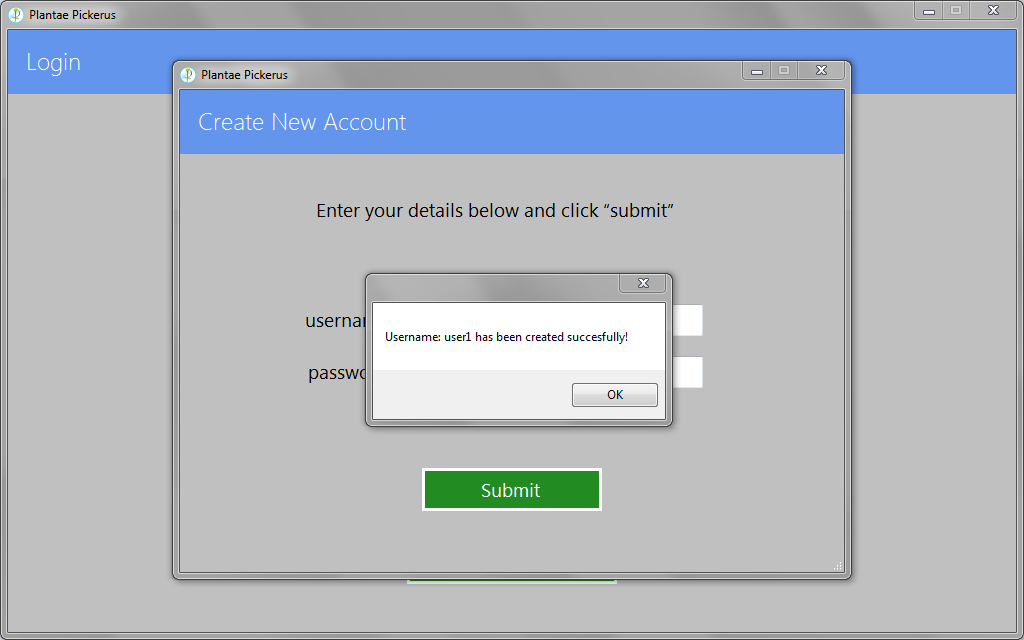
|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | Expected Results | Actual Results | PASS? |
| Create new username: *user1* with password: *password1* | User: *user1* has been created in the database. | User: user1 has been created correctly in the database. Also the user1 can log in using this username. | **Yes** |
| Create new username: *user1*  with password: *password2* | Error message: “User with this name already exist”. | Correct error message appears. | **Yes** |
| Create new username: user5=/!  with password: password3 | Error message: “Username can contain only letters and numbers”. | Correct error message appears. | **Yes** |
| Create new username: user3 with password: password5#-$ | Error message: “Password  can contain only letters and numbers”. | Correct error message appears. |  |

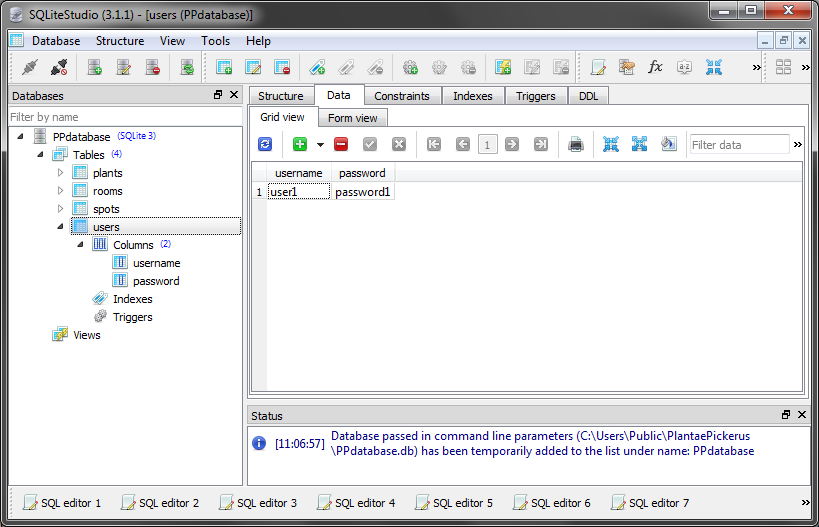
### Testing Screenshots:

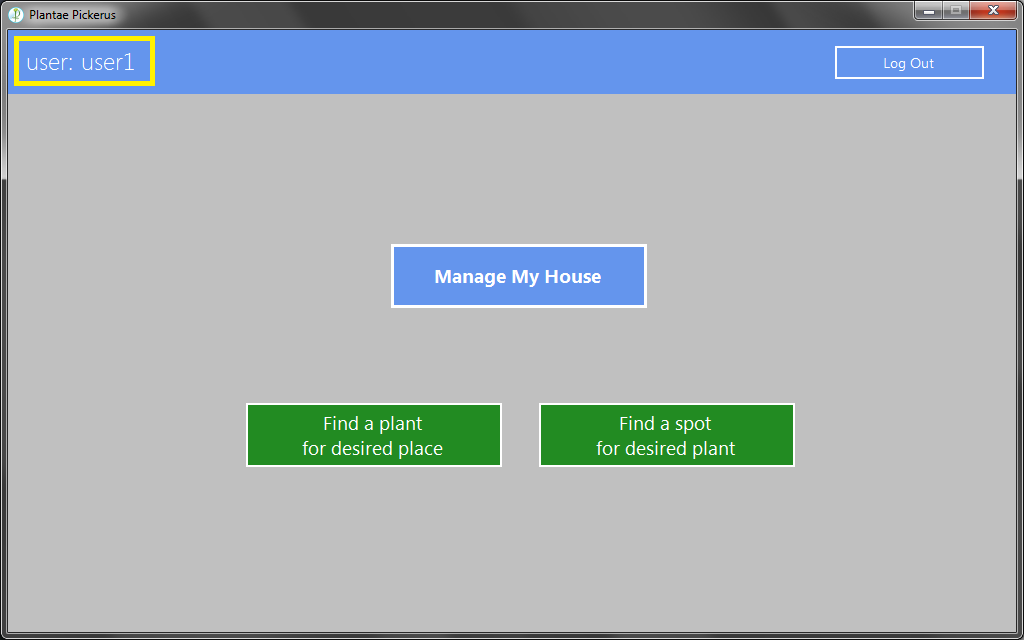
Testing: Create new user *user1*



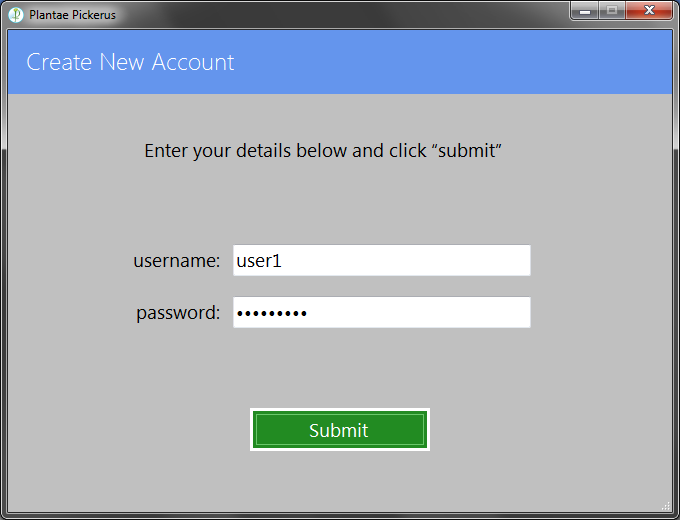


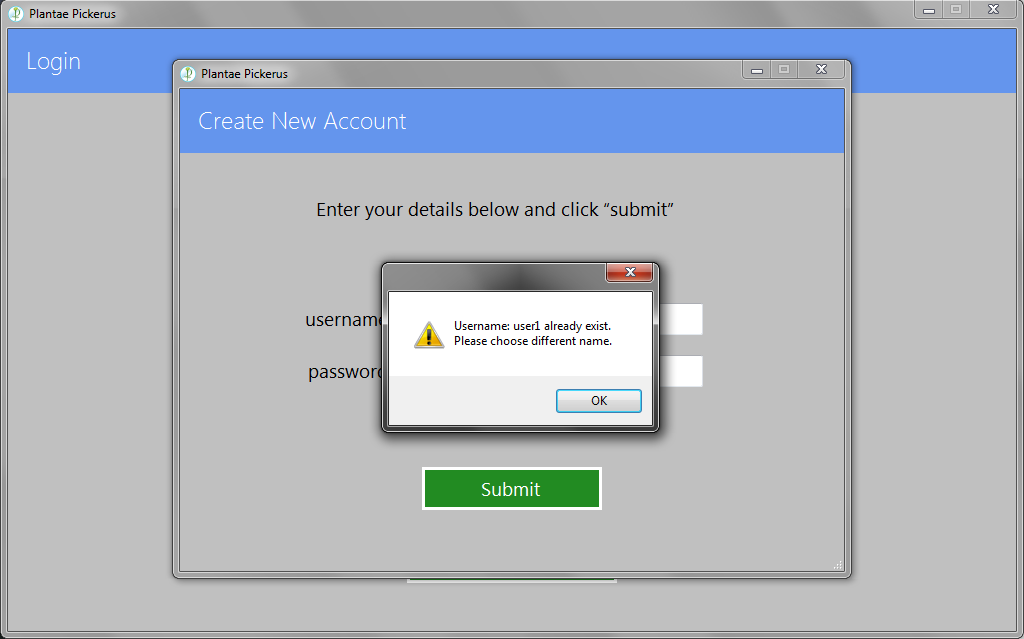




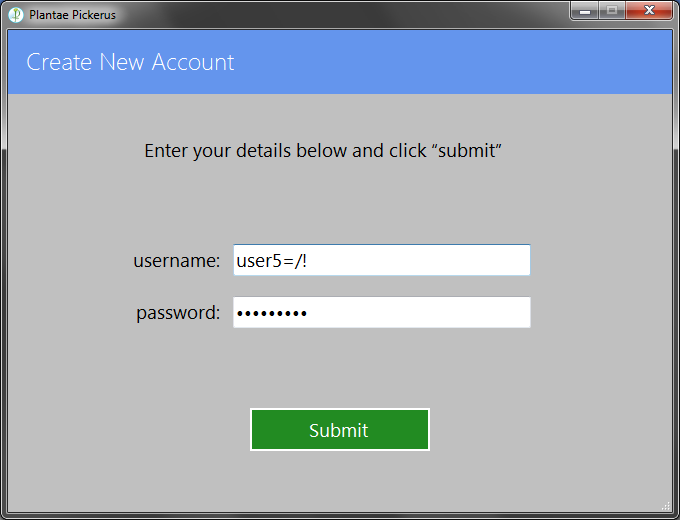


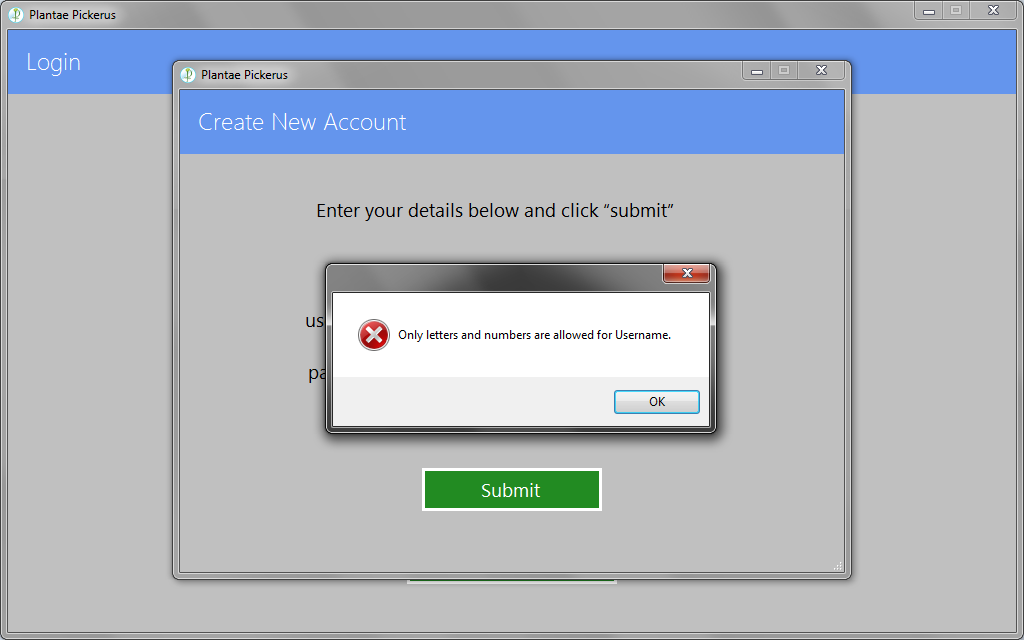
Testing: Create user with existing name

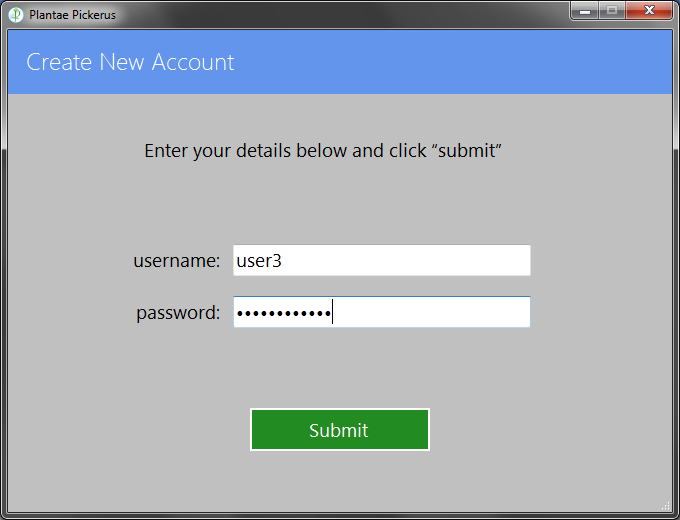


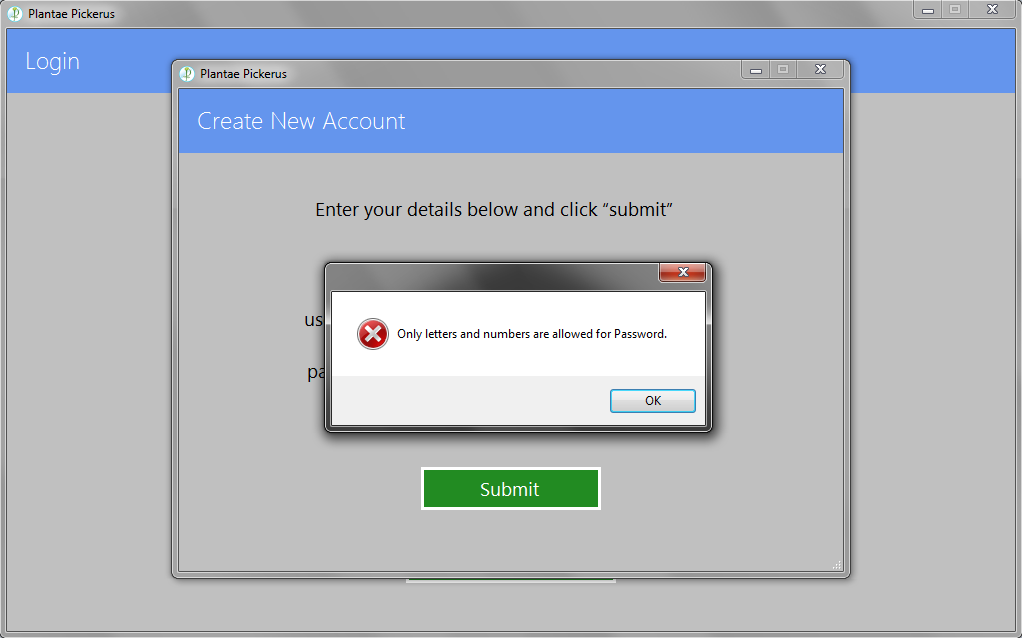


Testing: Allowed characters for username/password: only letters and digits









## Test Case 4

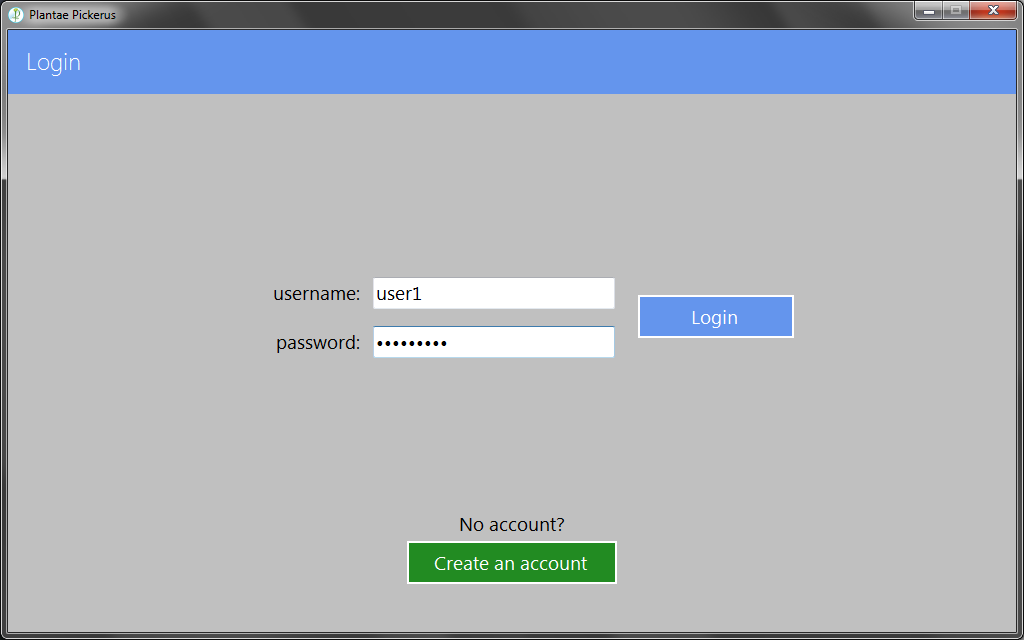
### Validate Username/Password

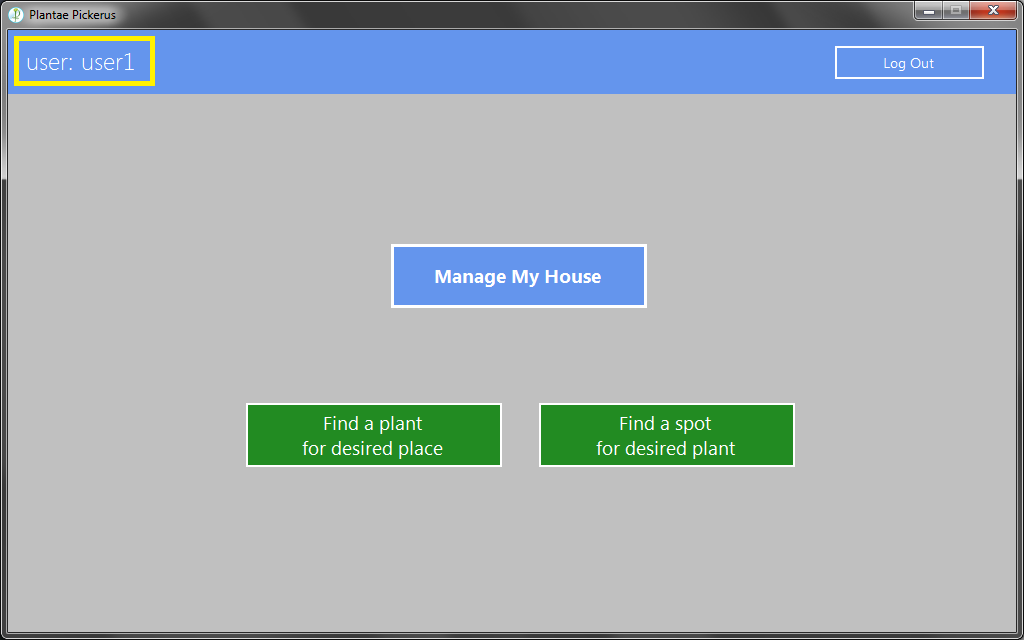
Ensure that the password matches correct username.  
Ensure that the password’s characters are displayed as dots: “•”

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | Expected Results | Actual Results | PASS? |
| Login to the system using username:  *user1* and password: *password1* | The user should login to the system. The corresponding username should be displayed on the Main windows top bar/header. | The user is logged in correctly. The username: user1 is displayed on the top of the Main screen. | **Yes** |
| Login to the system using username: *user555*  and password: *password1* | The user shouldn’t be logged int. Proper error message should be displayed. | The user cannot login using non-matching username/password combination. | **Yes** |
| Login to the system using username: *user1*  and password: *password555* | The user shouldn’t be logged int. Proper error message should be displayed. | The user cannot login using non-matching username/password combination. | **Yes** |
| Login to the system using username:  *user1* and password: *password1*  Pay attention to the password characters! | The characters for typing in password shouldn’t show actual characters, but they should be masked with  dots “•” | The password is masked with dots. | **Yes** |

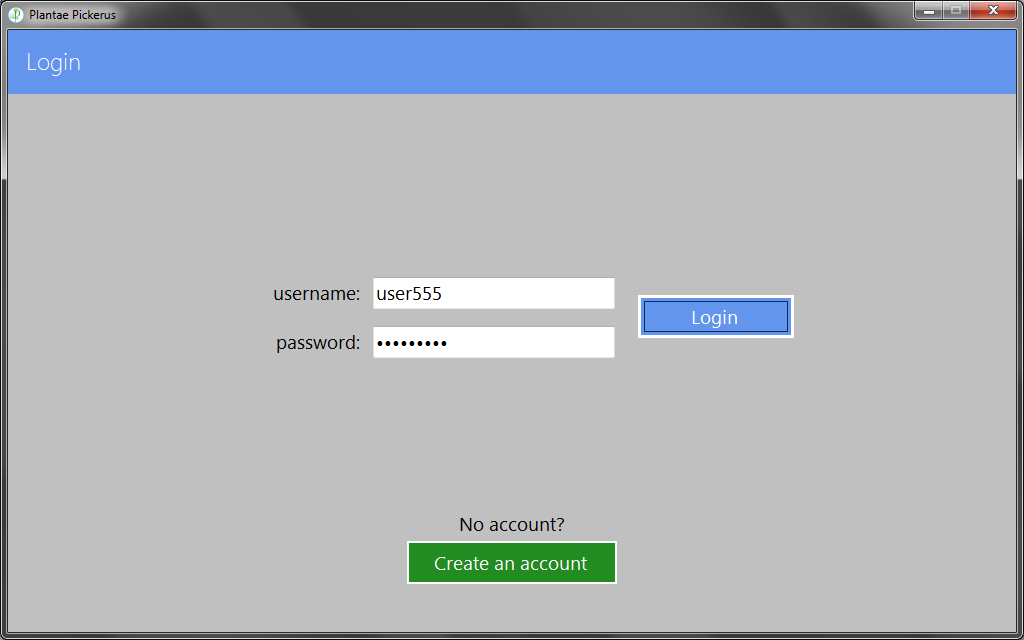
### Testing Screenshots:

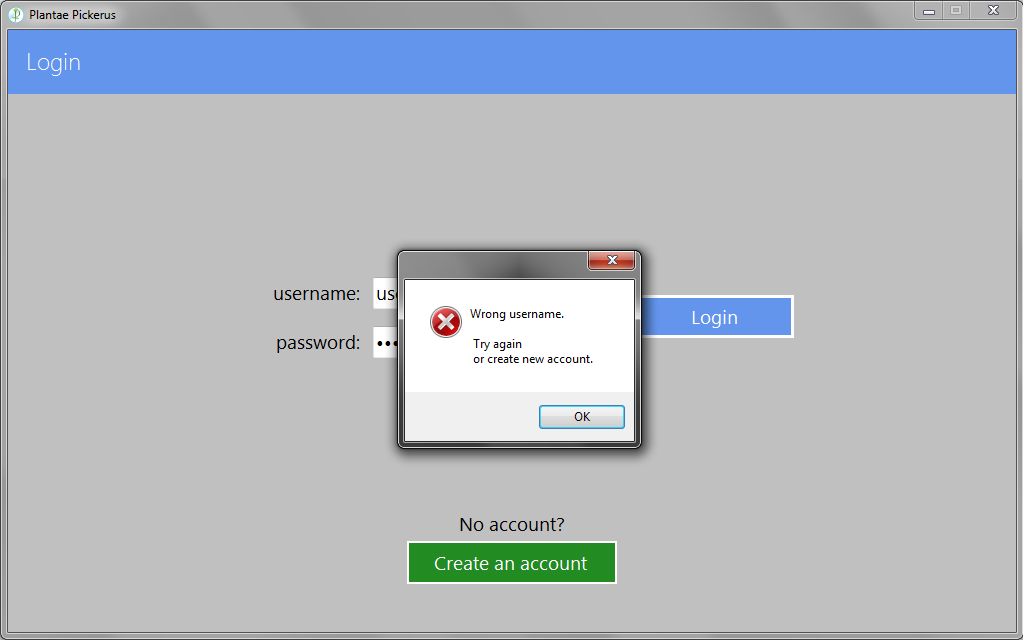
Testing: Login using matching username/password

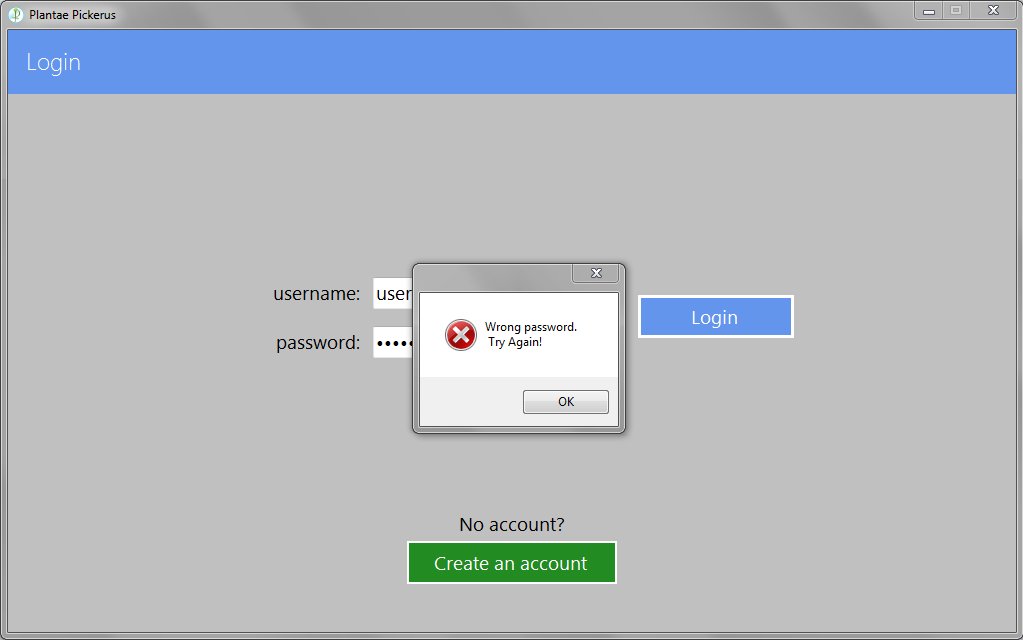
**

**

Testing: Login using non-matching username/password

**

**

**

## Test Case 5

### Main Menu

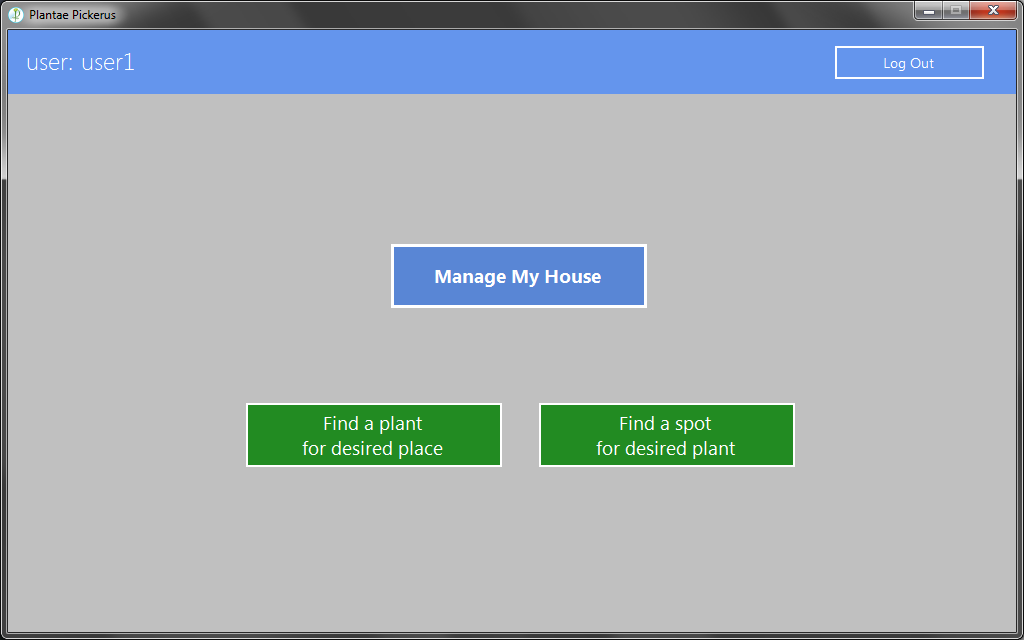
Main Menu should be displayed with proper buttons that work correctly.

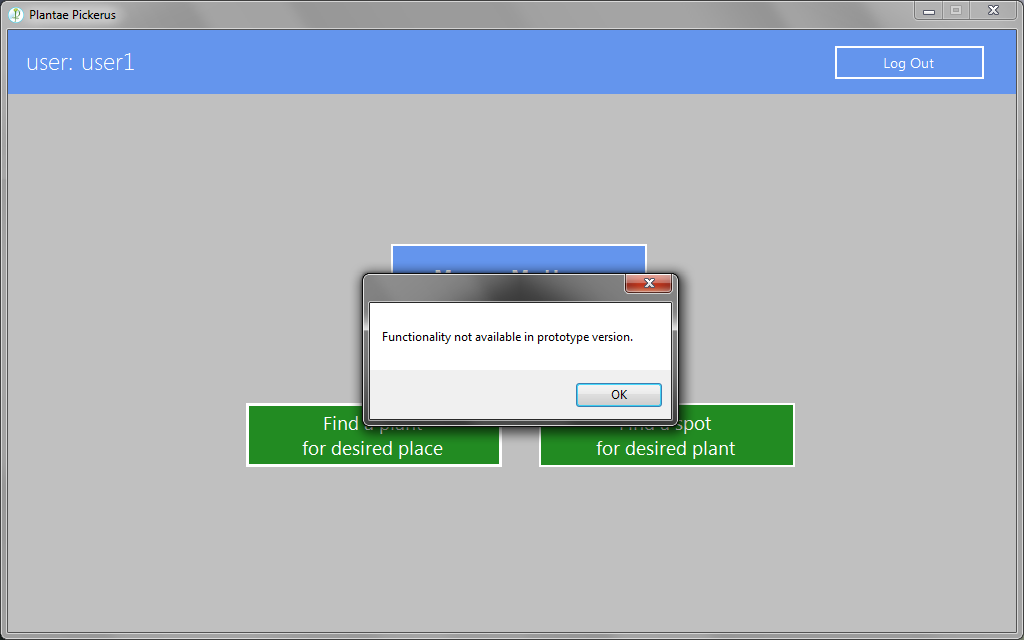
Ensure that the user has access to the Manage My House functionality   
and the information about his: rooms, spots, plants.

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | Expected Results | Actual Results | PASS? |
| Login to the application | After correct login, the Main Menu should be displayed, including buttons: “Log Out”, “Manage My House”, “Find a plant…”, Find a spot”.  The button should be clickable and should run corresponding activities. | Manage My House and Log Out buttons work correctly. Button to “Find a plant” and “Find a spot” work, but only show the message that this functionality isn’t included in the  prototype version | **Yes (partially)** |
| 1. Login to the application 2. Click on the “Manage My House” button | Window “Manage My House” should display a list with: rooms, spots, plants | Manage My House screen appears with the list of rooms, but not with the list of spots or plants. There is info to double-click the room name to see these information. | **Yes (partially)** |

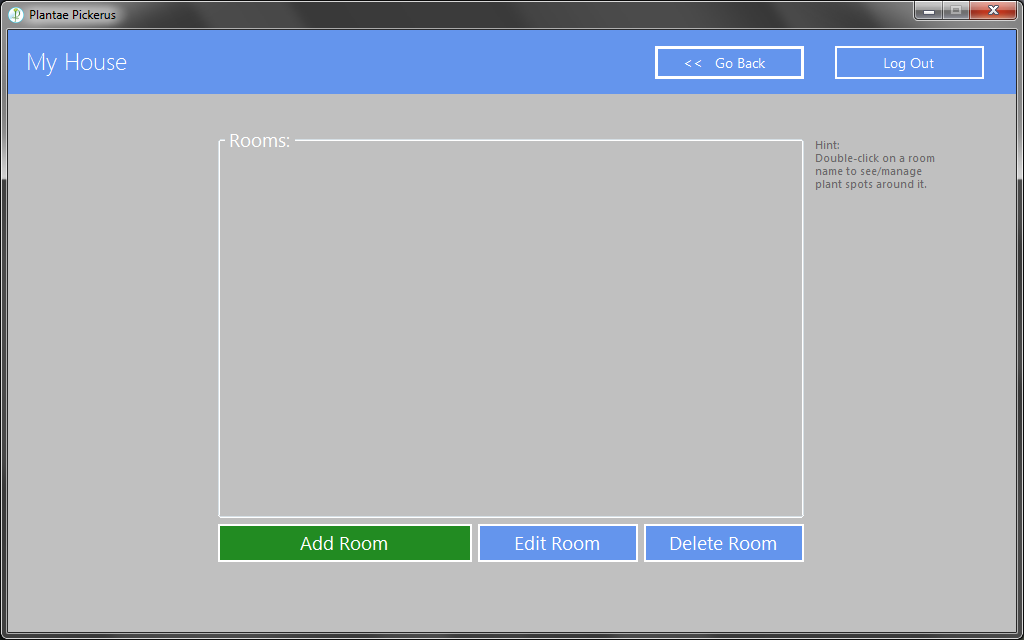
### Testing Screenshots:

Testing: Main Menu



**

Testing: Manage My House

**

## Test Case 6

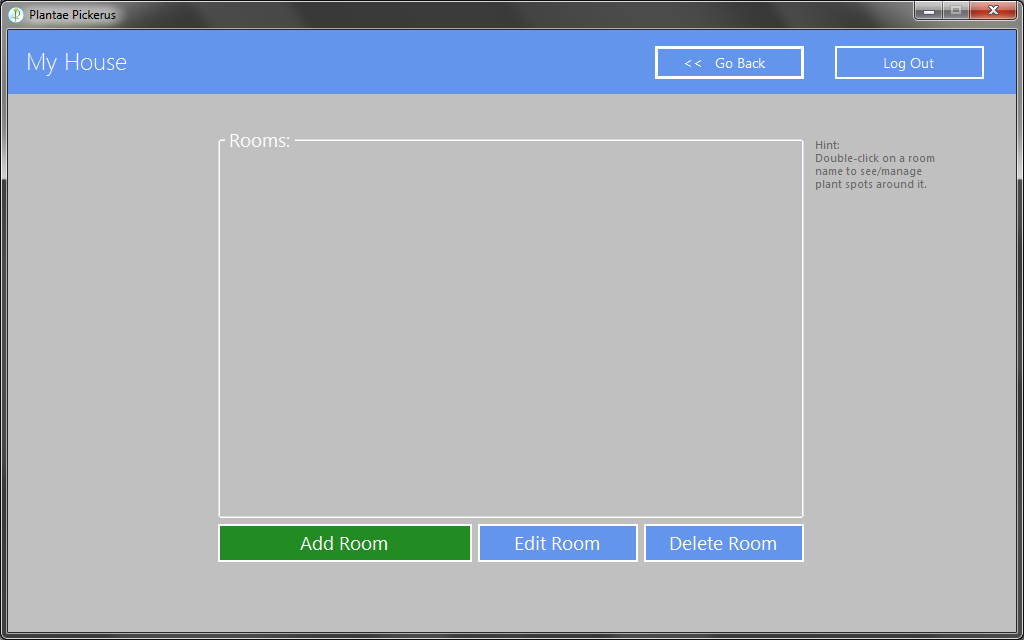
### User’s Rooms

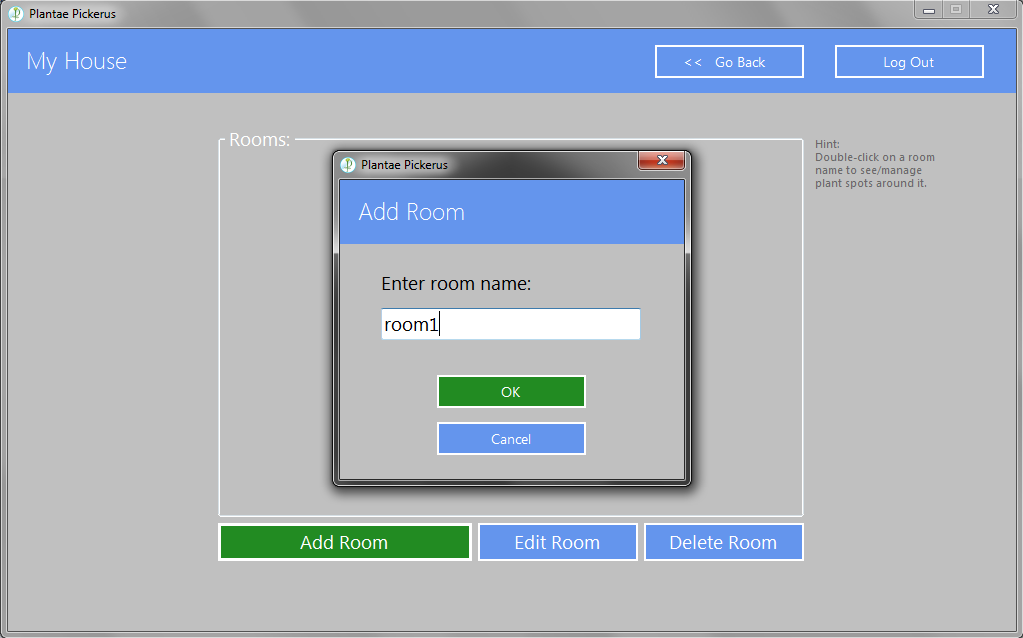
Ensure that the view shows *rooms* of logged-in user.  
Ensure that the *room* has been added successfully.  
Ensure that the *room* has been updated (edited and saved) successfully.  
Ensure that the *room* has been deleted successfully.

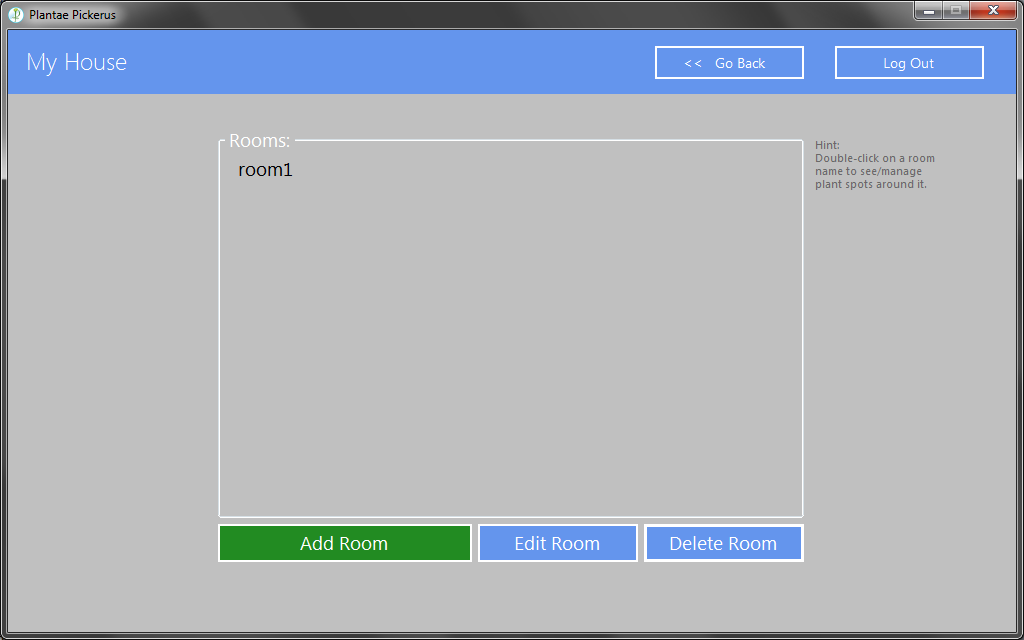
|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | Expected Results | Actual Results | PASS? |
| Go to: Manage My House menu/window | List of rooms should be displayed. | Empty list of rooms is displayed. | **Yes** |
| In Manage My House menu: click on “Add Room” button, fill the fields with any values and confirm. | Room should be added to the list of rooms. | Room has been added. The list of the rooms updated automatically. | **Yes** |
| In Manage My House menu: choose a room and click on “Edit Room” button, change the values and confirm. | Edited room should appear on the list as an updated room (updated name). | Room has been edited and the list of the rooms updated automatically. | **Yes** |
| In Manage My House menu: choose a room and click on “Delete Room” button and confirm. | The room should be removed from the list. | Room has been deleted from the list of the rooms. | **Yes** |
| Go to: Manage My House menu/window | List of rooms should display current user’s rooms. | List of current rooms is displayed. | **Yes** |

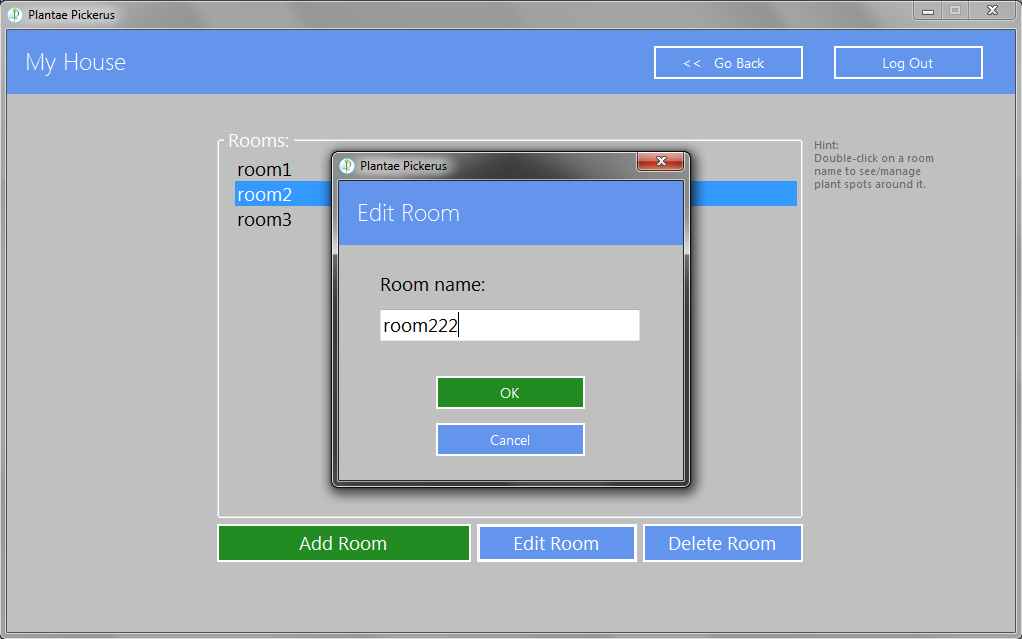
### Testing Screenshots:

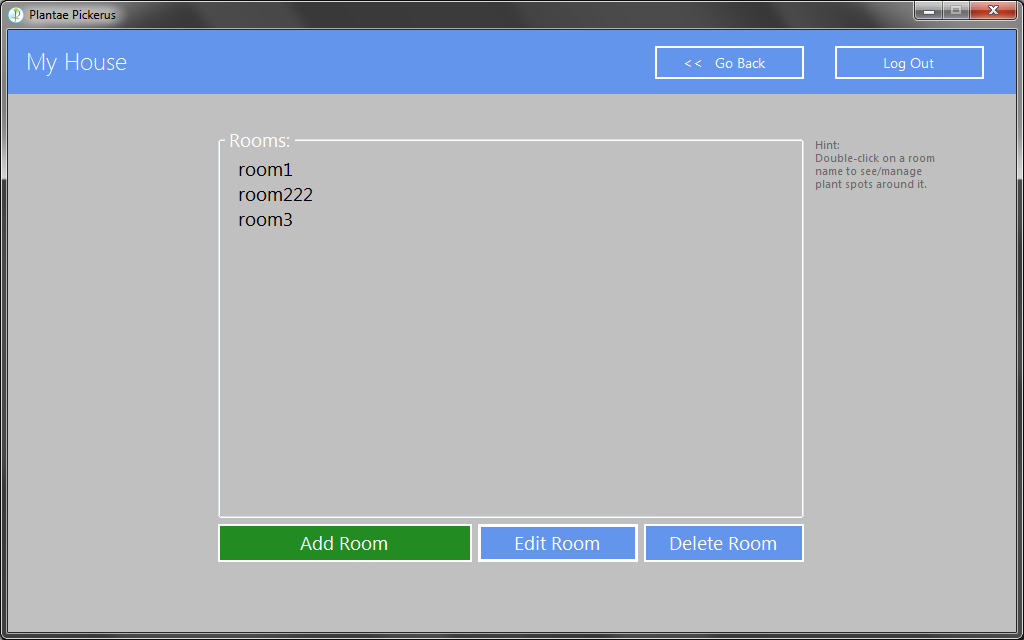
Testing: Show/Add/Edit/Delete/Show Rooms

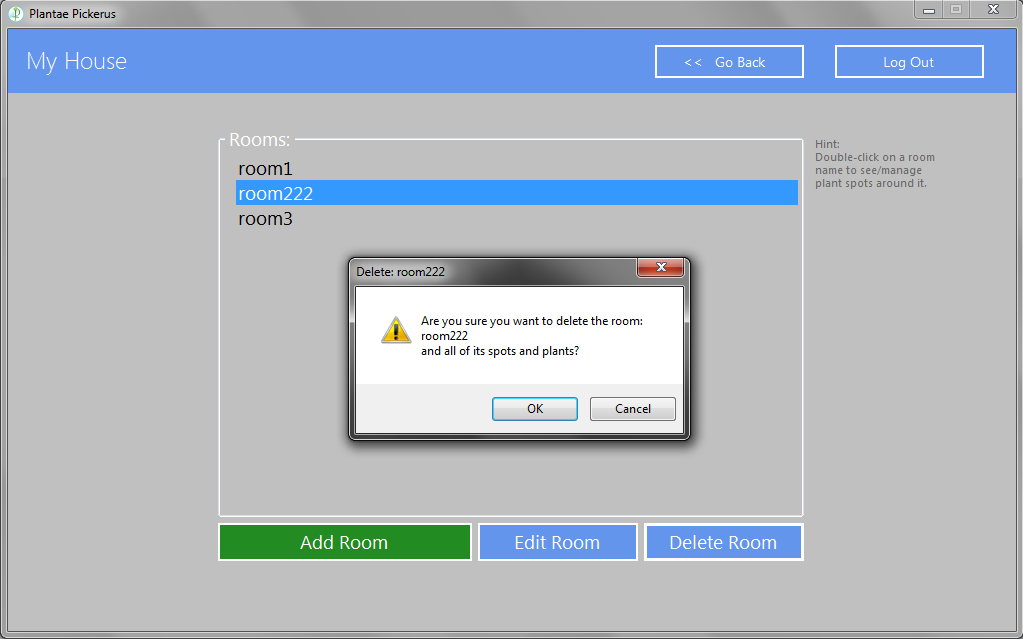


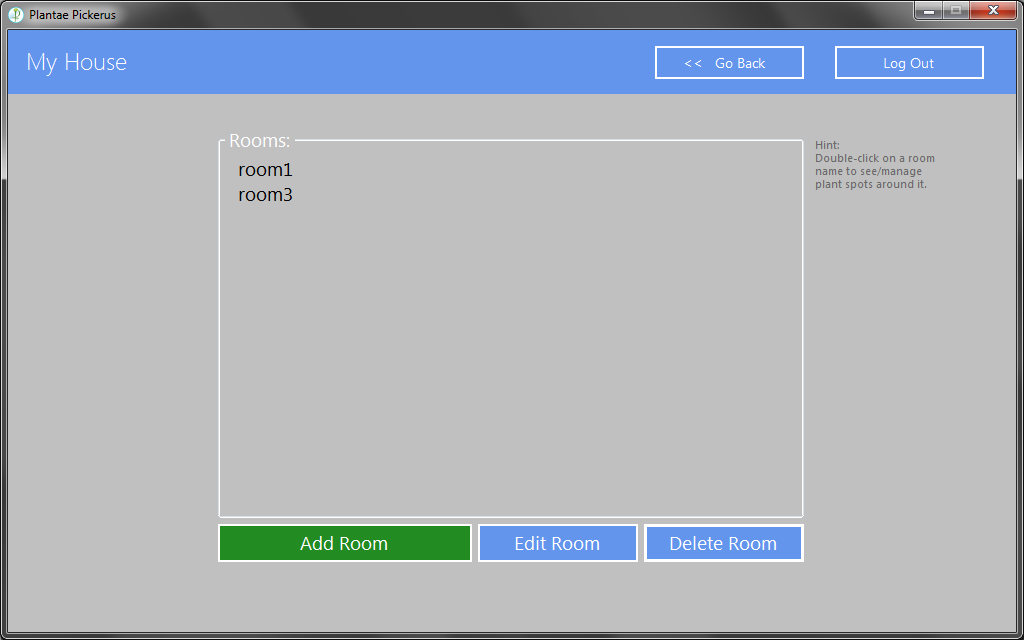
**

**









## Test Case 7

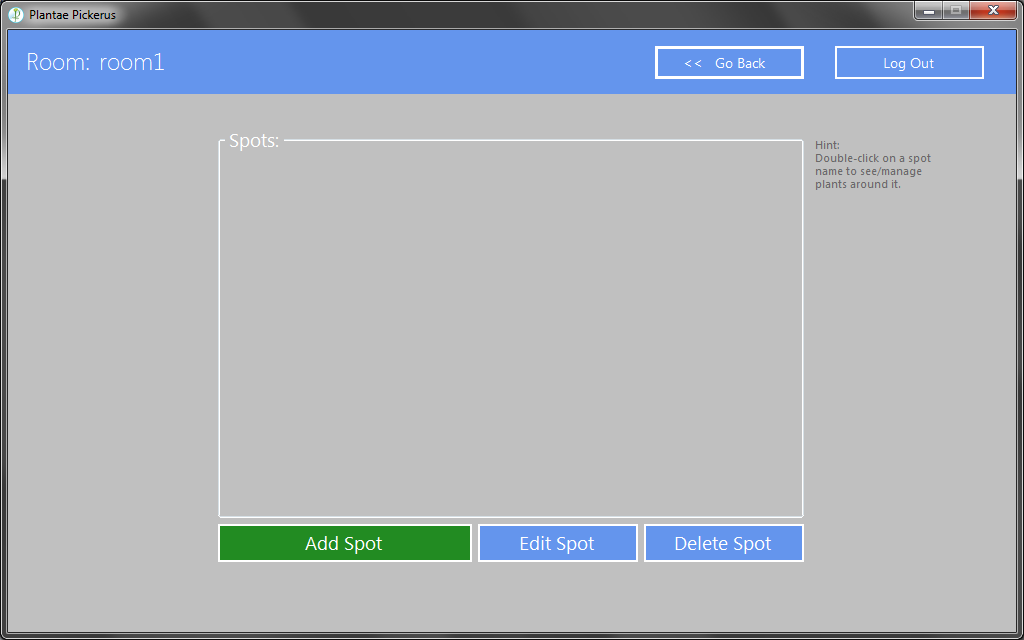
### User’s Spots

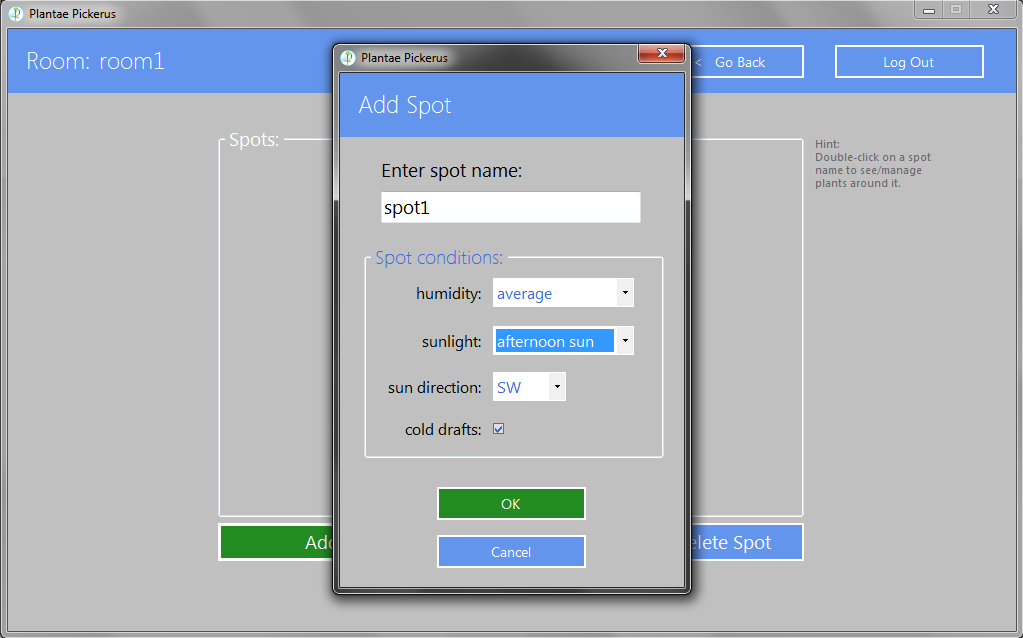
Ensure that the view shows *spots* of logged-in user.  
Ensure that the *spot* has been added successfully.  
Ensure that the *spot* has been updated (edited and saved) successfully.  
Ensure that the *spot* has been deleted successfully.

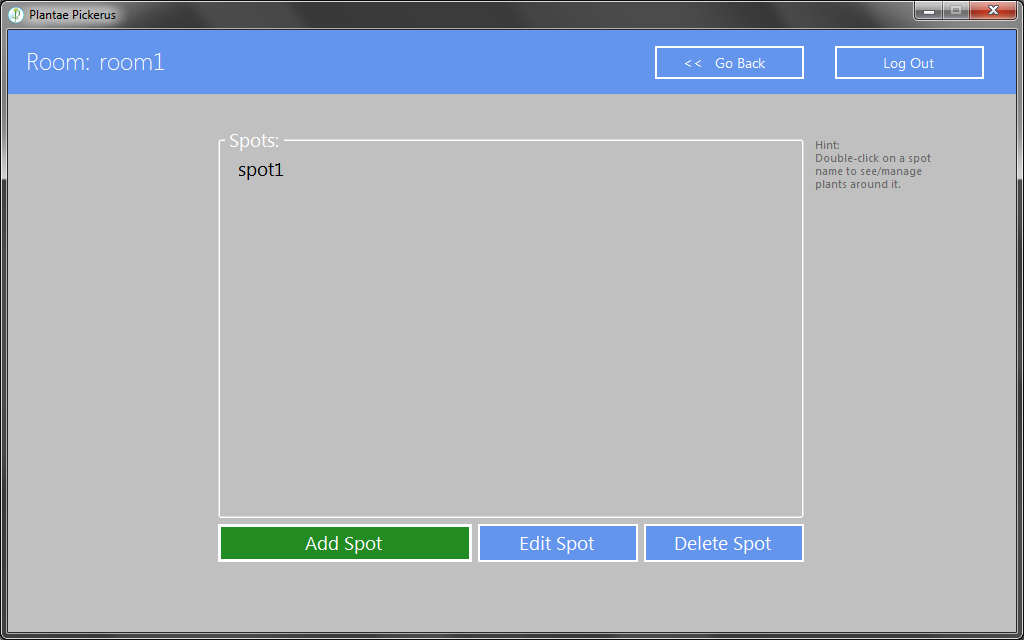
|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | Expected Results | Actual Results | PASS? |
| Go to: Manage My House menu/window. Double-click on any room. | List of spots should be displayed. | Empty list of spots is displayed. | **Yes** |
| In Manage My Room menu: click on “Add Spot” button, fill the fields with any values and confirm. | Spot should be added to the list of spots. | Spot has been added. The list of the spots updated automatically. | **Yes** |
| In Manage My Room menu: choose a spot and click on “Edit Spot” button, change the values and confirm. | Edited spot should appear on the list as an updated spot (updated name). | Spot has been edited and the list of the spots updated automatically. | **Yes** |
| In Manage My Room menu: choose a spot and click on “Delete Spot” button and confirm. | The spot should be removed from the list. | Spot has been deleted from the list of the spots. | **Yes** |
| Go to: Manage My House menu/window. Double-click on any room. | List of rooms should display current user’s spots. | List of current spots is displayed. | **Yes** |

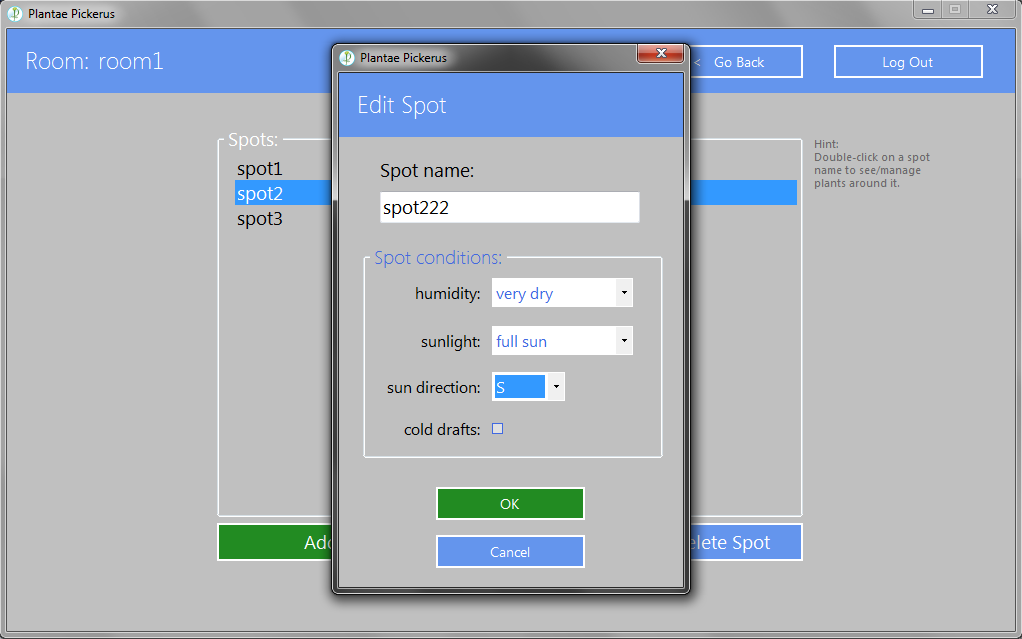
### Testing Screenshots:

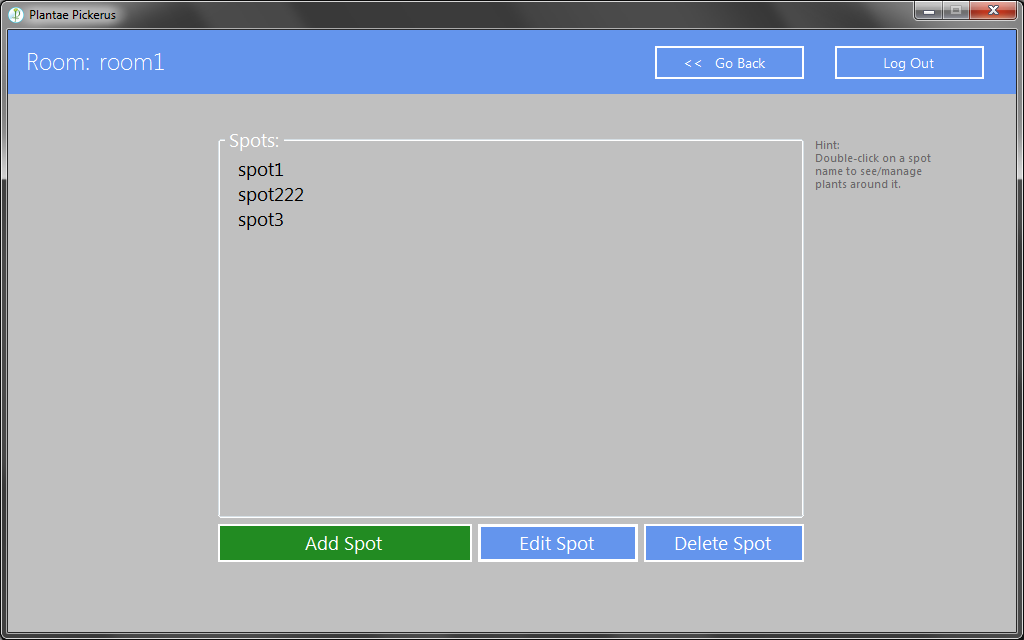
Testing: Show/Add/Edit/Delete/Show Spots

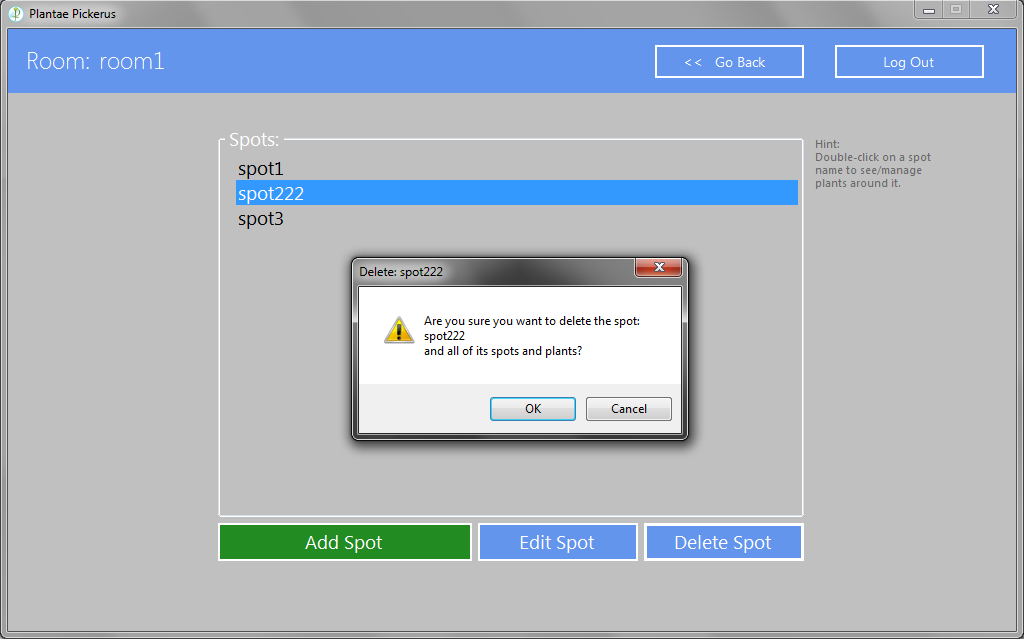


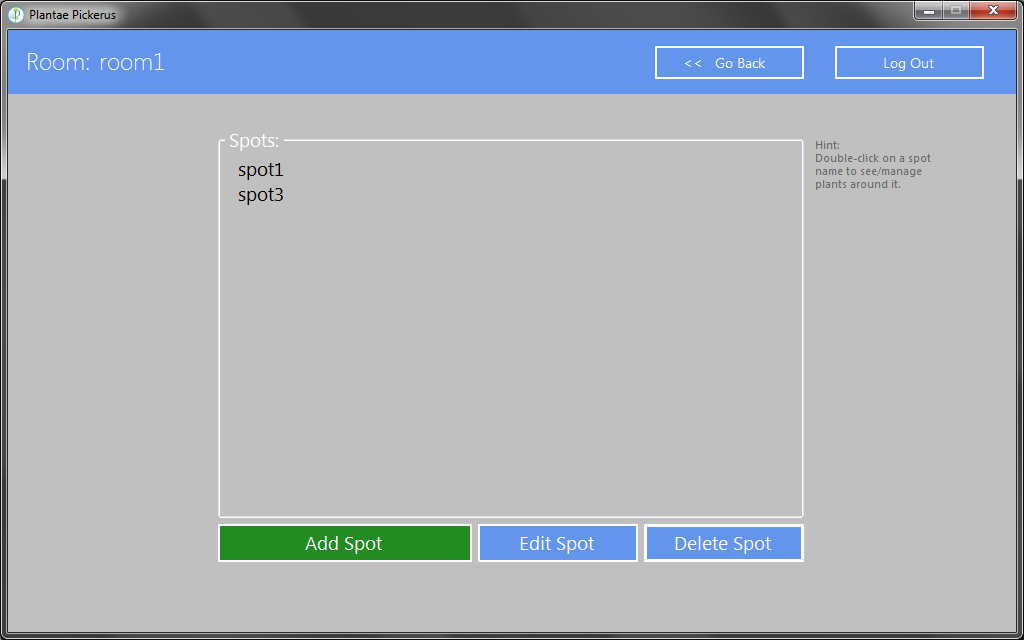












## Test Case 8

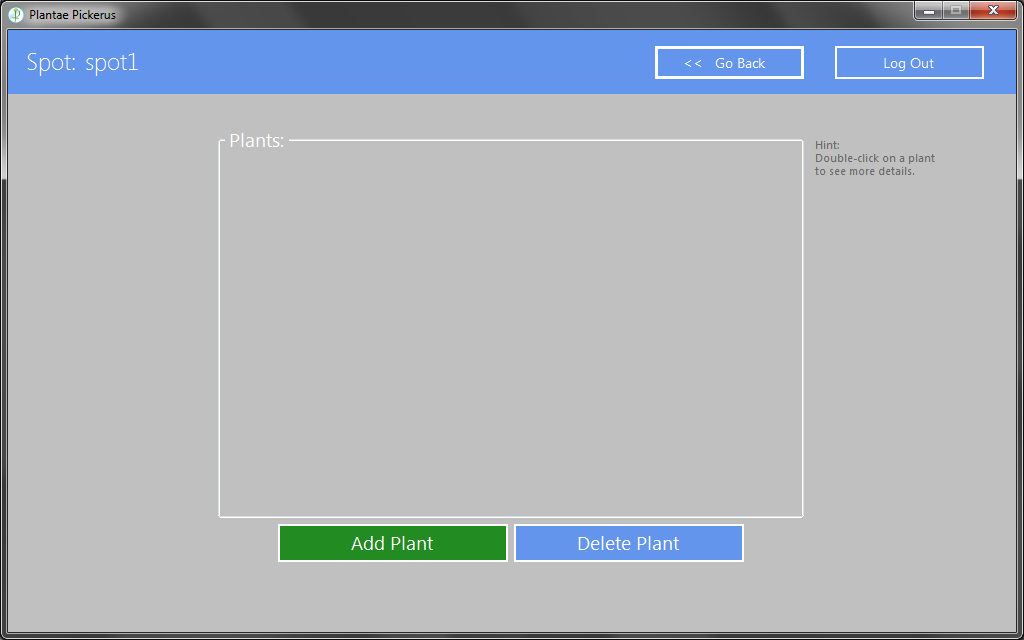
### User’s Plants

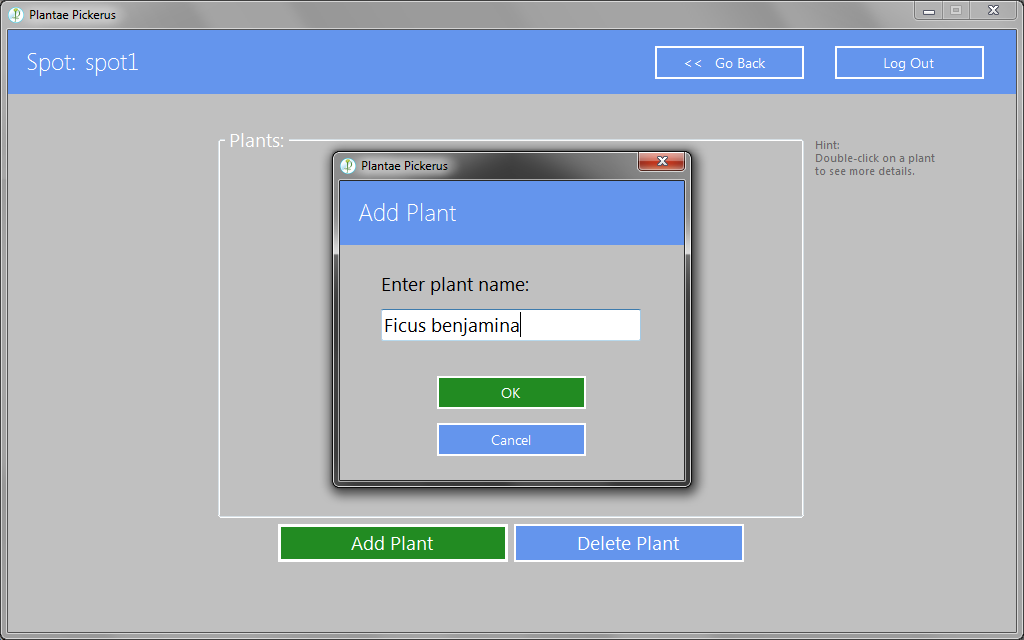
Ensure that the view shows *plants* of logged-in user.  
Ensure that the *plant* has been added successfully.  
Ensure that the *plant* has been deleted successfully.

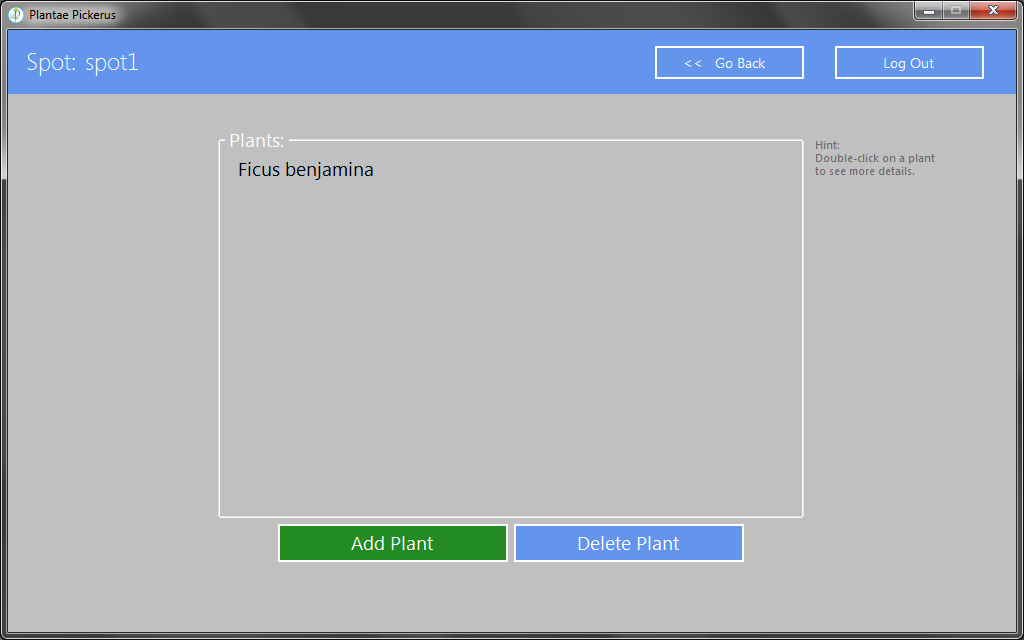
|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | Expected Results | Actual Results | PASS? |
| Go to: Manage My House menu/window. Double-click on any room then double-click on any spot. | List of plants should be displayed. | Empty list of plants is displayed. | **Yes** |
| In Manage My Spot menu: click on “Add Plant” button, fill the fields with any values and confirm. | Plant should be added to the list of plants. | Plant has been added. The list of the plants updated automatically. | **Yes** |
| In Manage My Spot menu: choose a plant and click on “Delete Plant” button and confirm. | The plant should be removed from the list. | Plant has been deleted from the list of the plants. | **Yes** |
| Go to: Manage My House menu/window. Double-click on any room then double-click on any spot. | List of plants should display current user’s spots. | List of current plants is displayed. | **Yes** |

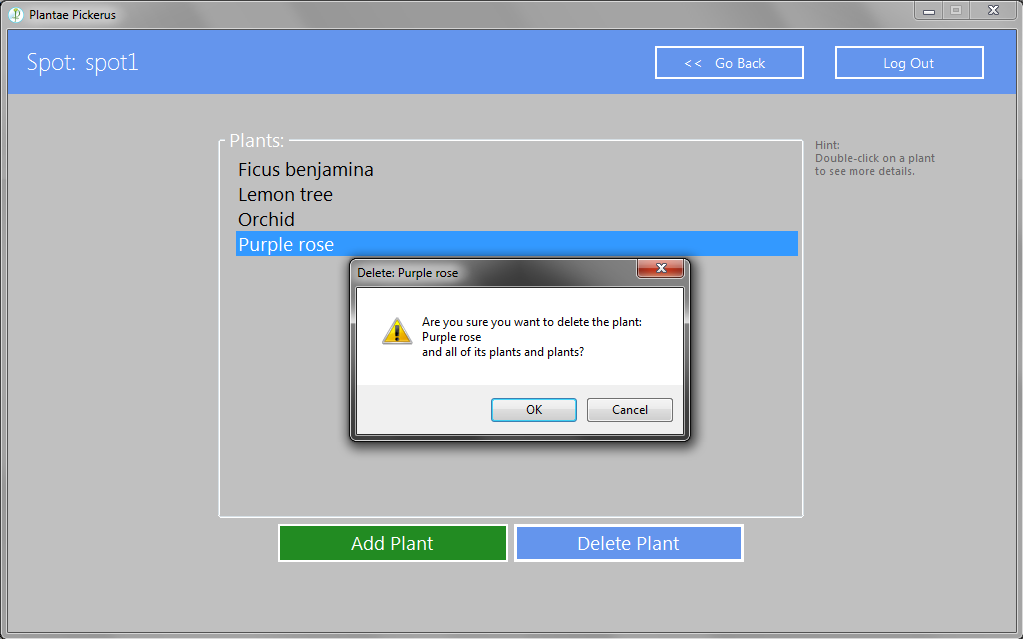
### Testing Screenshots:

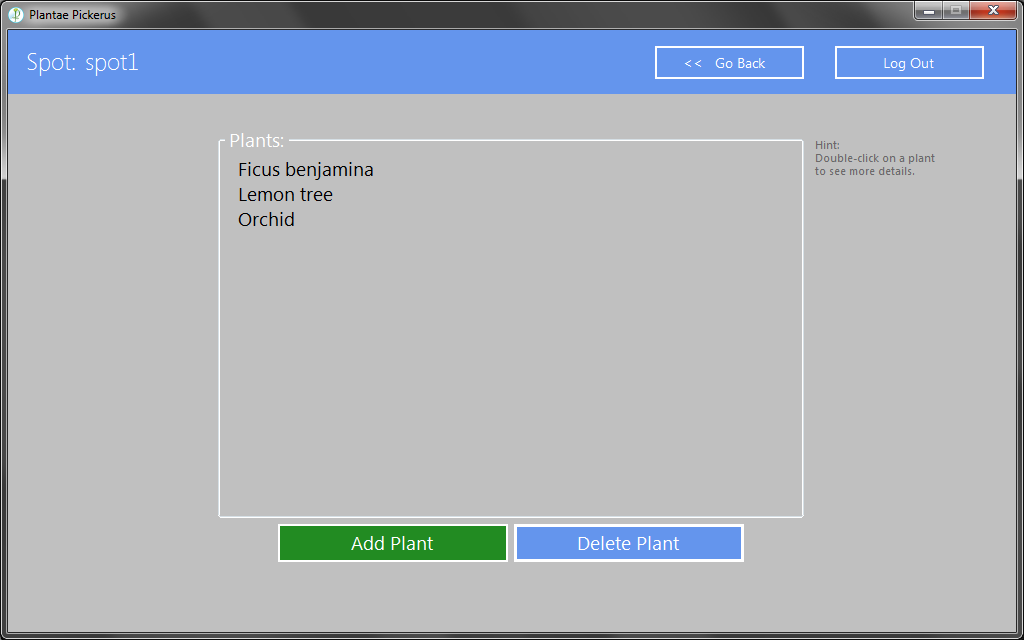
Testing: Show/Add/Delete/Show Plants











## Test Case 9

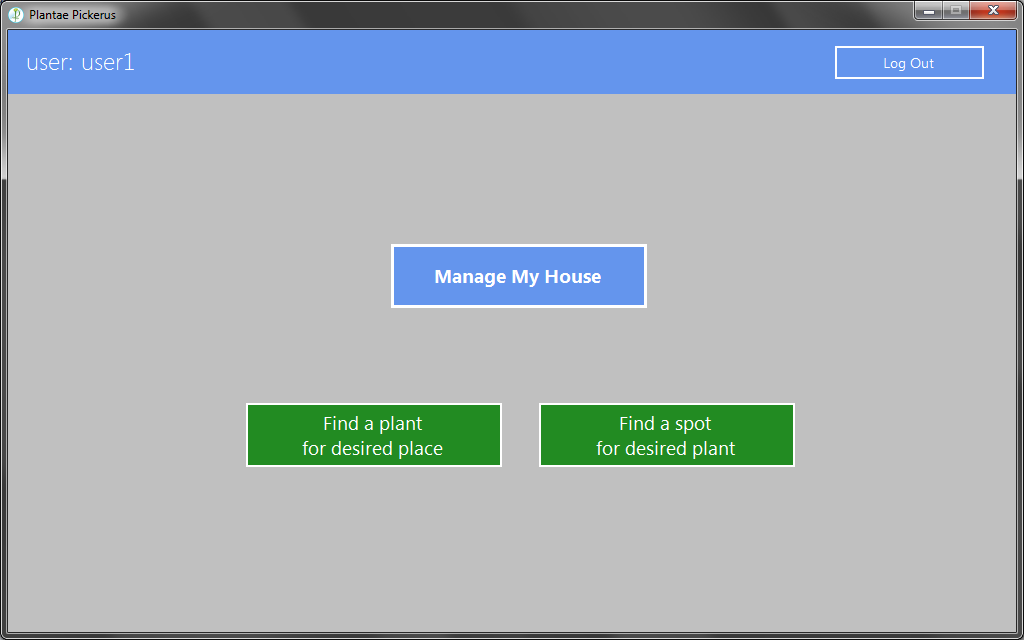
### Logout (from many windows)

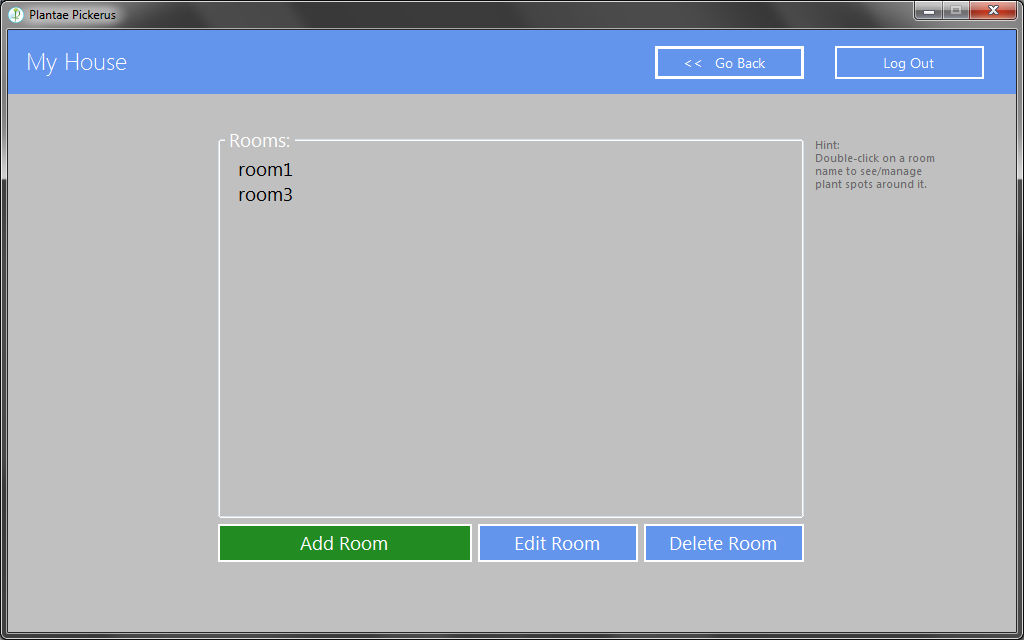
Ensure that Logout button logs out current user and displays Login Screen after.

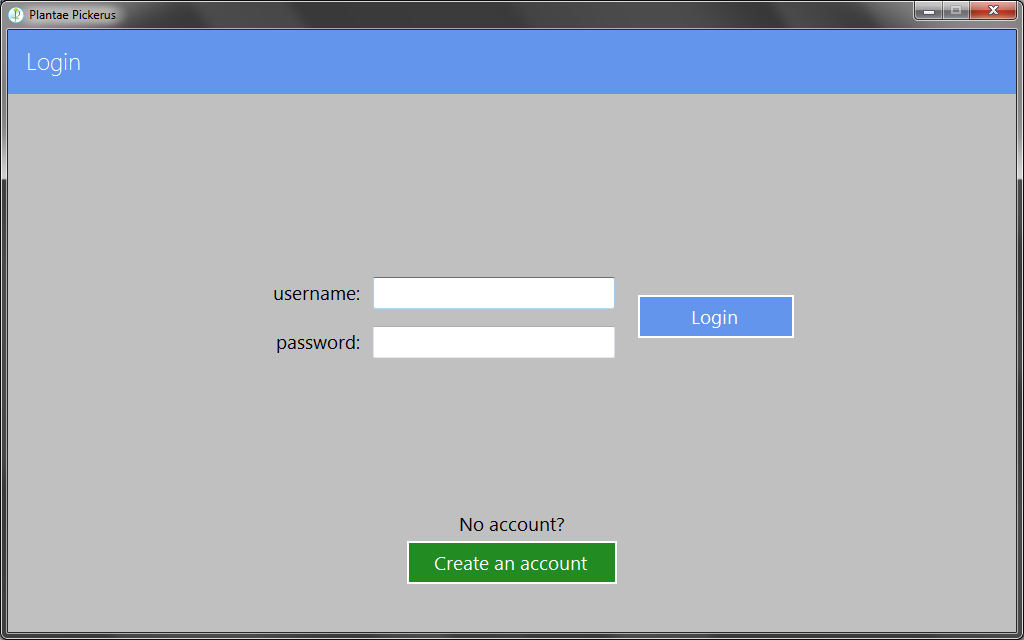
|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | Expected Results | Actual Results | PASS? |
| 1. Login to the application 2. Go to: Manage My House 3. Log out | The user should be logged out and the Login Screen should appear. | Login Screen appears correctly.  Other windows are closed. | **Yes** |
| 1. Login to the application 2. Go to: Manage My House 3. Double-click one of the rooms 4. Double-click one of the spots 5. Log out | The user should be logged out and the Login Screen should appear. | Login Screen appears correctly.  Other windows are closed. | **Yes** |

### Testing Screenshots:

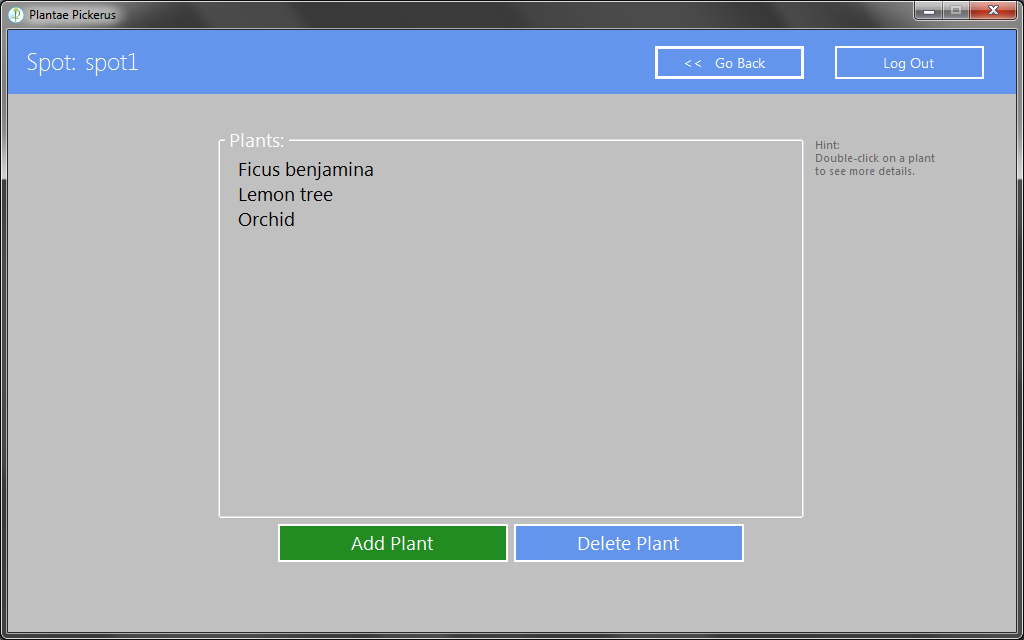
Testing: Logout from Manage My House window

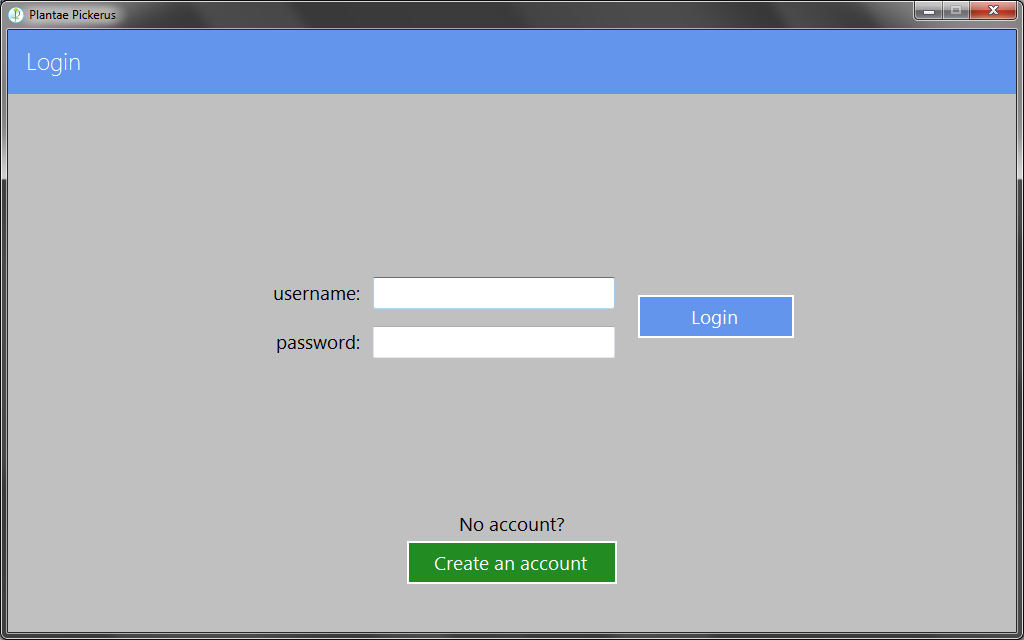






Testing: Logout from Manage My Spots window





## Test Case 10

### Quit/Exit Application (from many windows)

Ensure that the quit option (X button) exits the application.

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | Expected Results | Actual Results | PASS? |
| 1. Login to the application 2. Go to: Manage My House 3. Press [X] button to close | The application should be closed. | The program exits/is closed. | **Yes** |
| 1. Login to the application 2. Go to: Manage My House 3. Double-click one of the rooms 4. Double-click one of the spots 5. Press [X] button to close | The application should be closed. | The program exits/is closed. | **Yes** |
| 1. Login to the application 2. Press [X] button to close | The application should be closed. | The program exits/is closed. | **Yes** |
| 1. Run the application 2. Press [X] button to close | The application should be closed. | The program exits/is closed. | **Yes** |