



Software Testing Document (STD)

1. Introduction

In this document, we verify through testing, the different functionalities that were discussed in the SDD by referring to what the user is expected to input, and what the user should expect from the App to do. In addition to that, we assume several test cases (not just the ones typically inputted by the user) that would cover as much scenarios as possible (and thus verifying that the underlying defined events and conditionals work fully as desired). This is also the best way to ensure robustness of our application in front of special situations such as when a user enters unexpected, invalid information, or when boundary values are involved. We are aware that our testing approach could mostly be considered a black-box one, rather than both white and black, but the test sets we worked with guaranteed a high percentage of condition coverage (from the high-level coding point of view); and this is related to the fact that the white-box approach requires a high level knowledge of internals of the software and under test and therefore certainly requires full access to the underlying code (this is something we lack in our case since the lower level programming is hidden from developers working on the **Bubble** environment).

2. Testing Procedure

Here we show all the test cases that were used per functionality (or module), while also displaying the generated result of the test from the User GUI's (whenever necessary).

Signing Up

➤ Test case 1

`<username = "jp", polimi email = "jeanpierre.sleiman@mail.polimi.it", password="123">`

It is assumed there are no other registered users with either the same email or the same username.

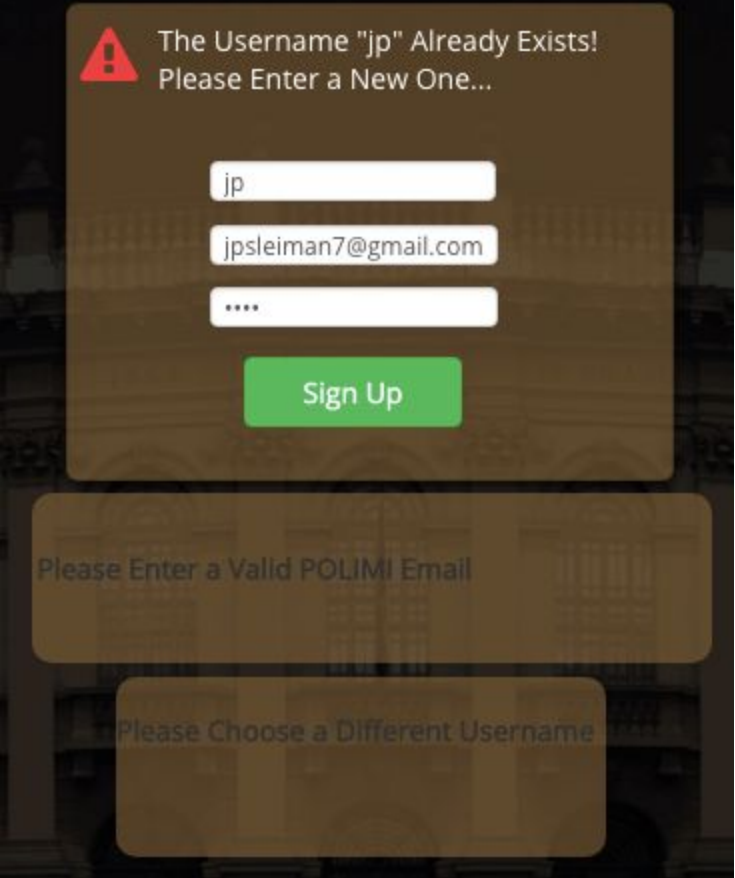
SIGNUP SUCCEEDS

➤ Test case 2

<username = "jp", polimi email = "jpsleiman7@gmail.com", password="1234">

A user already exists with the same username.

SIGNUP FAILS (invalid username and invalid email)



The screenshot shows a dark-themed signup form. At the top, a red warning icon is next to the text: "The Username 'jp' Already Exists! Please Enter a New One...". Below this, there are three input fields: the first contains "jp", the second contains "jpsleiman7@gmail.com", and the third contains four dots. A green "Sign Up" button is positioned below the input fields. Below the button, there are two more error messages in separate boxes: "Please Enter a Valid POLIMI Email" and "Please Choose a Different Username".

➤ Test case 3

<username = "mkk", polimi email = "michael.khayyat@mail.polimi.it", password="1234">

A user already exists with the same polimi email.

SIGNUP FAILS (email already in use)



➤ Test case 4

<username = " ", polimi email = "jeanpierre.sleiman@mail.polimi.it", password="123">

Or if any of the other input components are empty.

SIGNUP FAILS (empty component)

Logging in

➤ Test case 1

<polimi email = "jeanpierre.sleiman@mail.polimi.it", password="123">

It is assumed this email is already in the database (signed up) and the inputted password is the same as the one used during signup.

Note that the "Remember me" checkbox does not affect the Login functionality.

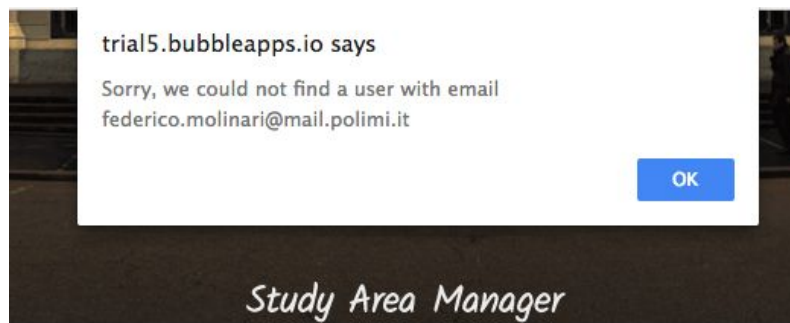
LOGIN SUCCEEDS

➤ Test case 2

<polimi email = "federico.molinari@mail.polimi.it", password="123">

Non-existing polimi email (in the database) is entered.

LOGIN FAILS (email can't be found)

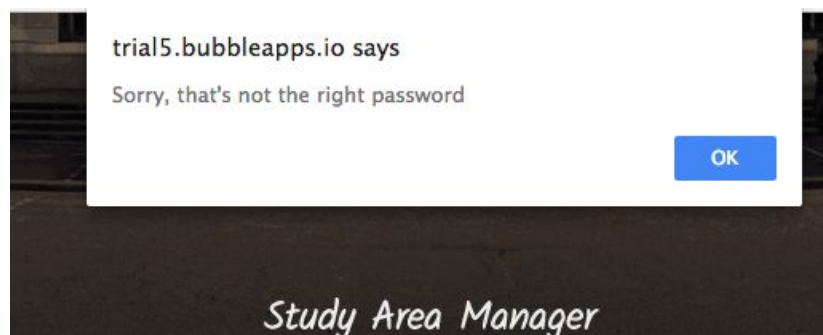
A screenshot of the login form in the mobile application. The form has a light brown background. It contains two input fields: the first for email, which has "federico.molinari@mail.polimi.it" entered, and the second for password, which has three dots "..." entered. Below the password field is a checkbox labeled "Remember me". At the bottom of the form is a blue button labeled "Login".

➤ Test case 3

<polimi email = "jeanpierre.sleiman@mail.polimi.it", password="hello">

Wrong password is entered.

LOGIN FAILS (wrong password)

A screenshot of the login form in the mobile application. The form has a light brown background. It contains two input fields: the first for email, which has "jeanpierre.sleiman@mail.polimi.it" entered, and the second for password, which has five dots "*****" entered. Below the password field is a checkbox labeled "Remember me". At the bottom of the form is a blue button labeled "Login".

Filtering

➤ Test case 1

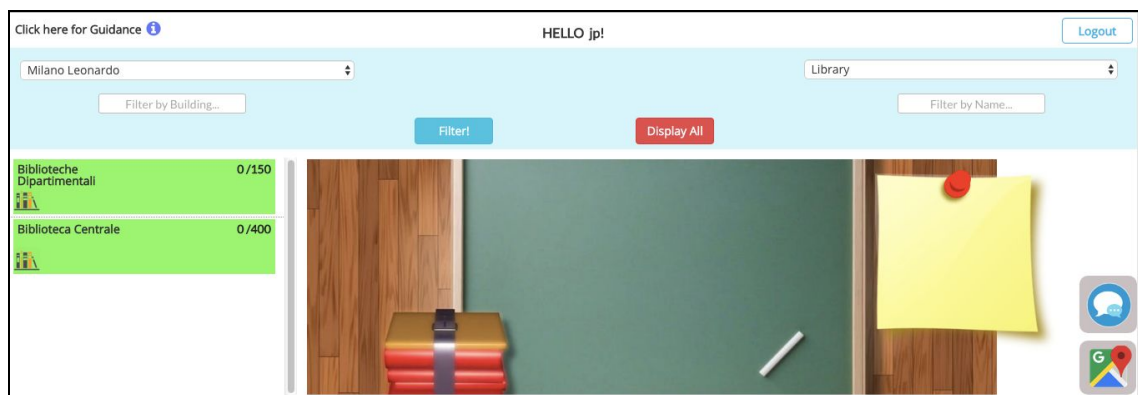
<Campus = x, Category = y , Building = z, Name = w>

It is assumed in this test case that the different combination of choices among x, y, z, w (they can also be empty), yields a Study Area that can be found in the database.

For example:

<Campus = "Milano Leonardo", Category = "Library", Building = " ", Name = " ">

FILTERING SUCCEEDS (the appropriate study-areas appear in the left column)



➤ Test case 2

<Campus = x, Category = y , Building = z, Name = w>

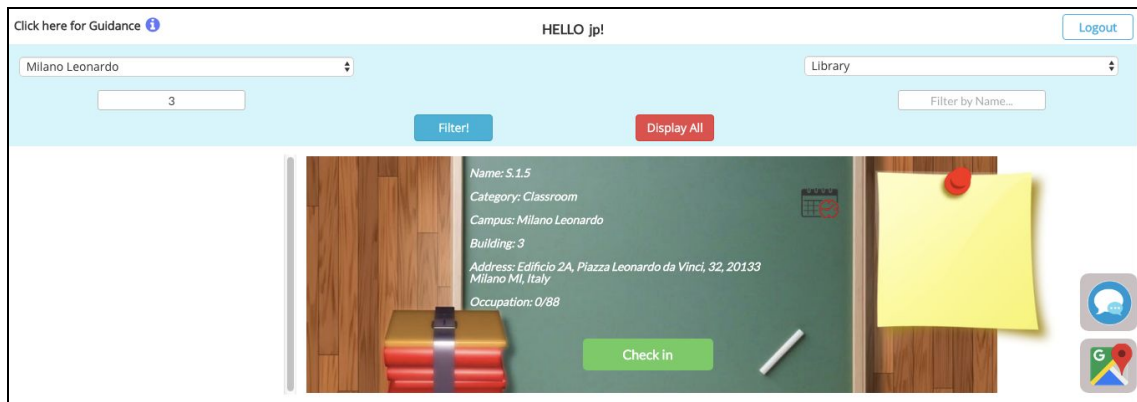
It is assumed in this test case that the different combination of choices among x, y, z, w yields a Study Area that cannot be found in the database.

For example:

<Campus = "Milano Leonardo", Category = "Library", Building = "3", Name = " ">

<Campus = " ", Category = " ", Building = " ", Name = "W.P.2">

FILTERING SUCCEEDS (No Study Areas appear in the left column)



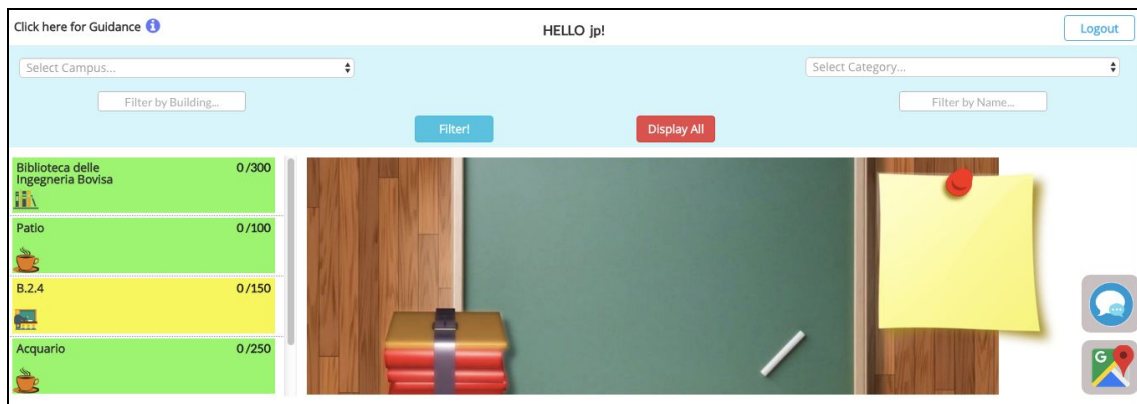
Checking In/Out

➤ Test case 1

<Selected Study Area = “ ”>

No study area has been selected yet.

CHECK IN FAILS (check in button doesn't even appear yet)

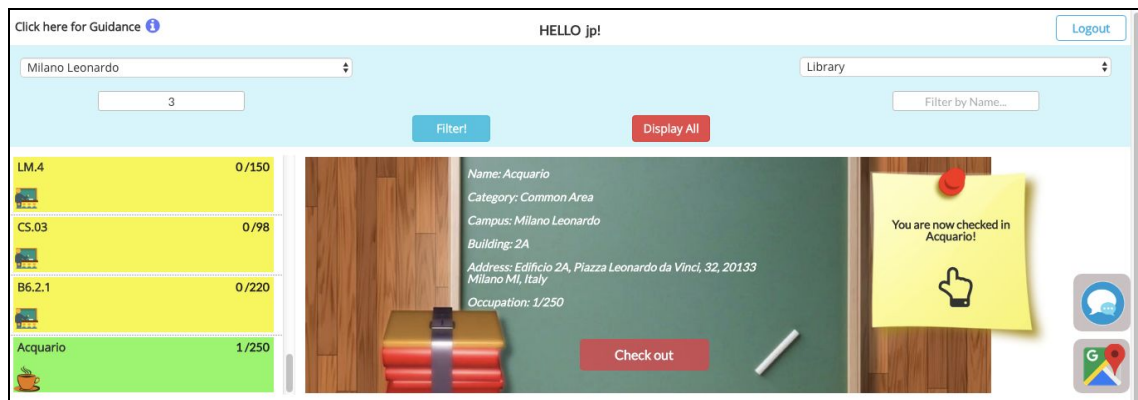


➤ Test case 2

<Selected Study Area = “Acquario”, Check in = true>

It is assumed that the user is not checked-in in any Study Area, and the number of occupants is 0 or any number less than the full capacity.

CHECK IN SUCCEEDS (as described in the design document)

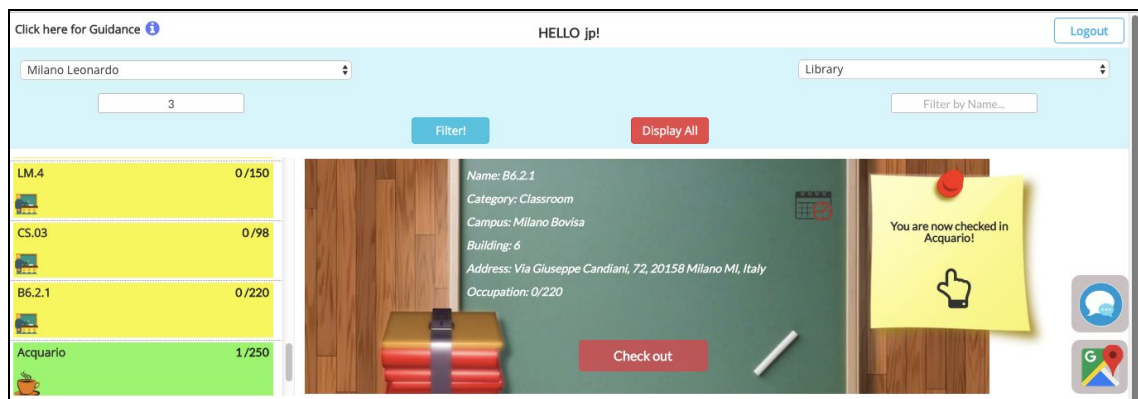


➤ Test case 3

<Selected Study Area = “B6.2.1”>

The user is already checked in Acquario.

CHECK IN SUCCEEDS (the student is still checked in Acquario but the information of B6.2.1 appears in the middle of the screen)



➤ Test case 4

<Selected Study Area = "B6.2.1", check out = true">

The user is already checked in Acquario.

CHECK OUT FAILS (no changes occur to the screen)

➤ Test case 5

<Selected Study Area = "B6.2.1", reload page = true">

The user reloads the page while already being checked in Acquario

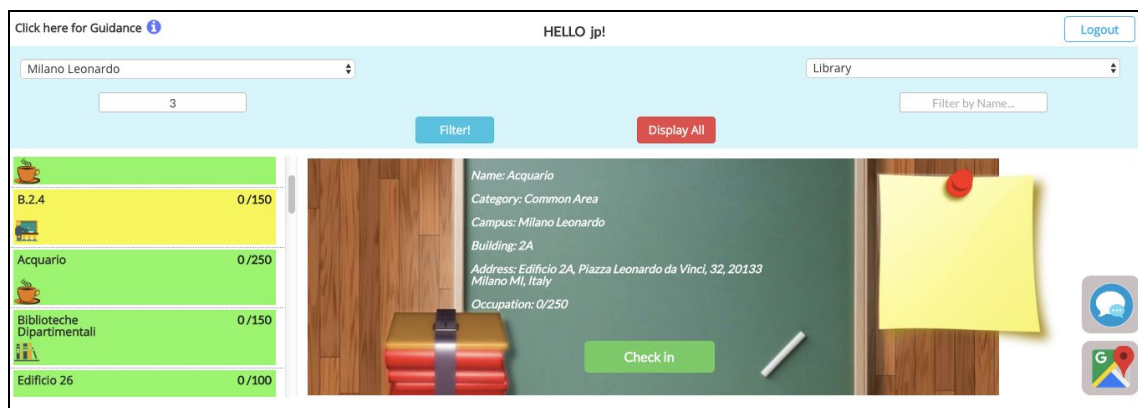
CHECK IN SUCCEEDS (the student is still checked in Acquario and its information appears in the middle of the screen)

➤ Test case 6

<Selected Study Area = "Acquario", check out = true">

The user is already checked in Acquario.

CHECK OUT SUCCEEDS (as described in the design document)

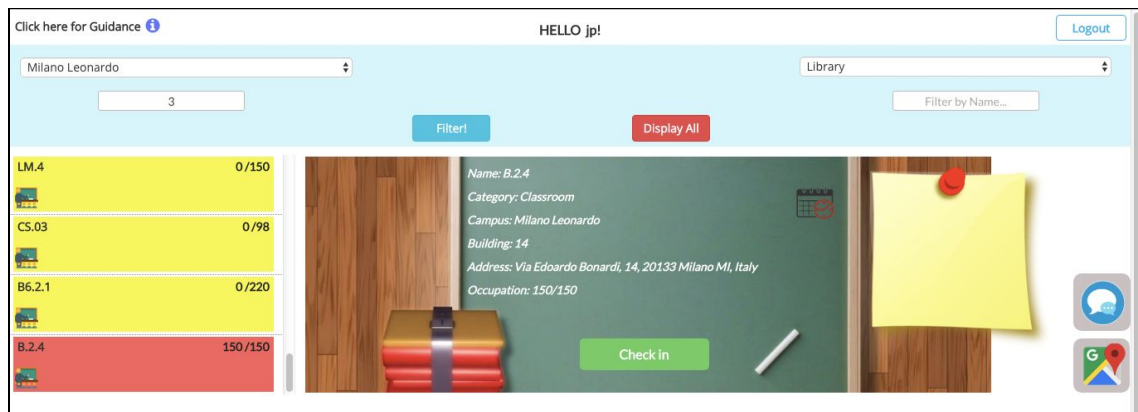


➤ Test case 7

<Selected Study Area = “B.2.4”, check in = true>

It is assumed in this case that the study area has reached its full capacity (boundary value)

CHECK IN FAILS



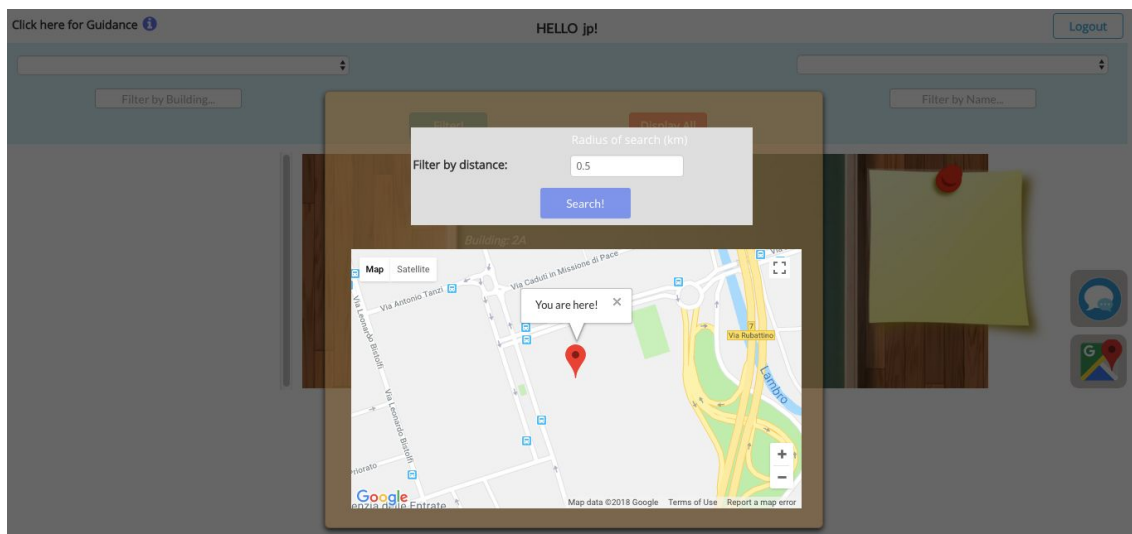
Searching by Distance

➤ Test case 1

<Radius of search = x>

It is assumed that x is not big enough to encompass any of the listed study areas in the database.

SEARCH SUCCEEDS (nothing appears on the map except for the user's current location, and nothing appears in the left column of the GUI)

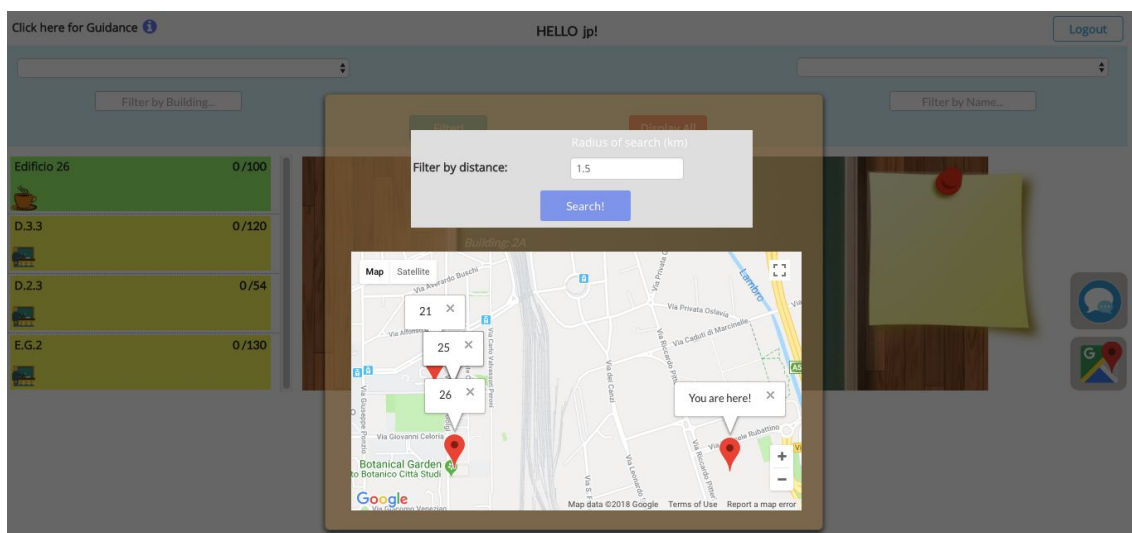


➤ Test case 2

<Radius of search = x>

It is assumed that x is big enough to encompass some of the listed study areas in the database.

SEARCH SUCCEEDS

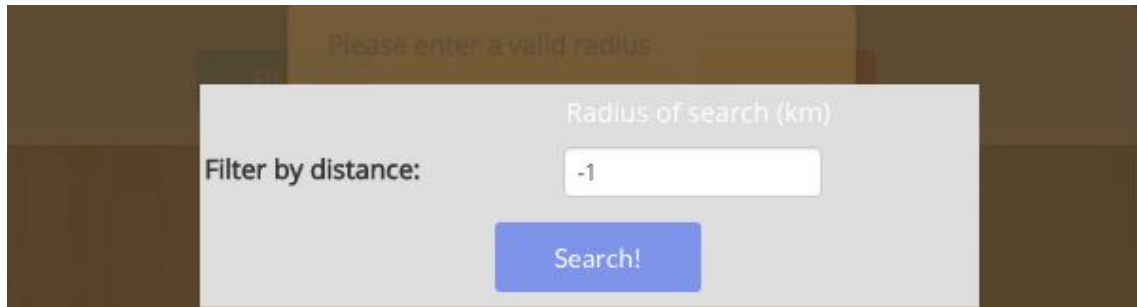


➤ Test case 3

<Radius of search = x>

It is assumed that x is invalid (i.e. less than or equal to zero)

SEARCH FAILS (an error message appears)



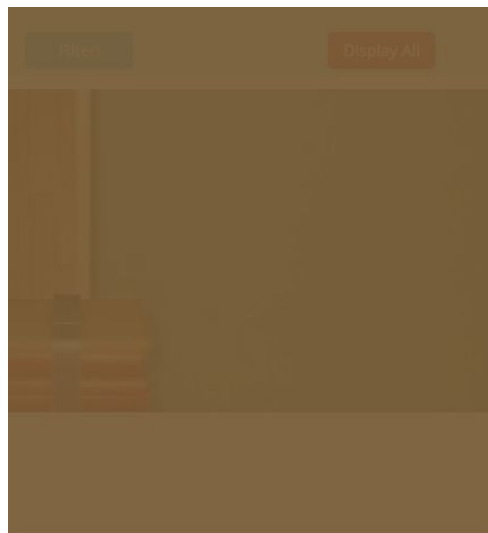
Posting/Deleting Messages

➤ Test case 1

<message = "Hello", student = "jp", selected post place = " ">

No study area has been selected yet.

POSTING FAILS (no option for posting messages appears)

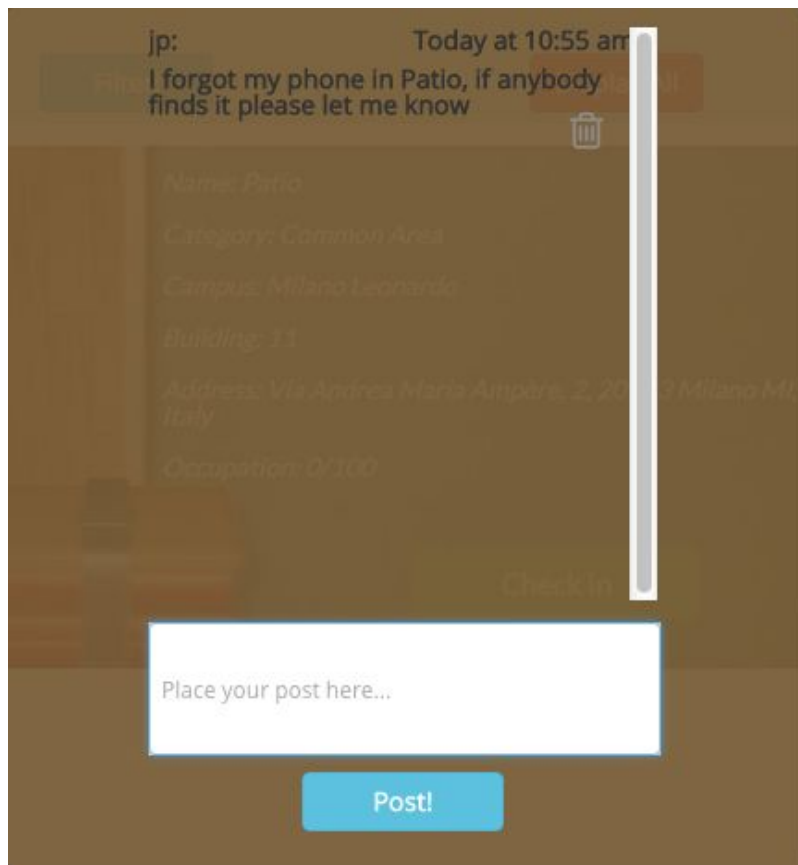


➤ Test case 2

<message = "I forgot my phone in Patio, if anybody finds it please let me know?",
student = "jp", selected post place = "Patio", post = true>

It is assumed the current user is "jp" ofcourse.

POSTING SUCCEEDS



➤ Test case 3

*<message = "I forgot my phone in Patio, if anybody finds it please let me know?",
student = "jp", selected post place = "Patio", delete = true>*

It is assumed the current user is "jp" ofcourse.

DELETING SUCCEEDS

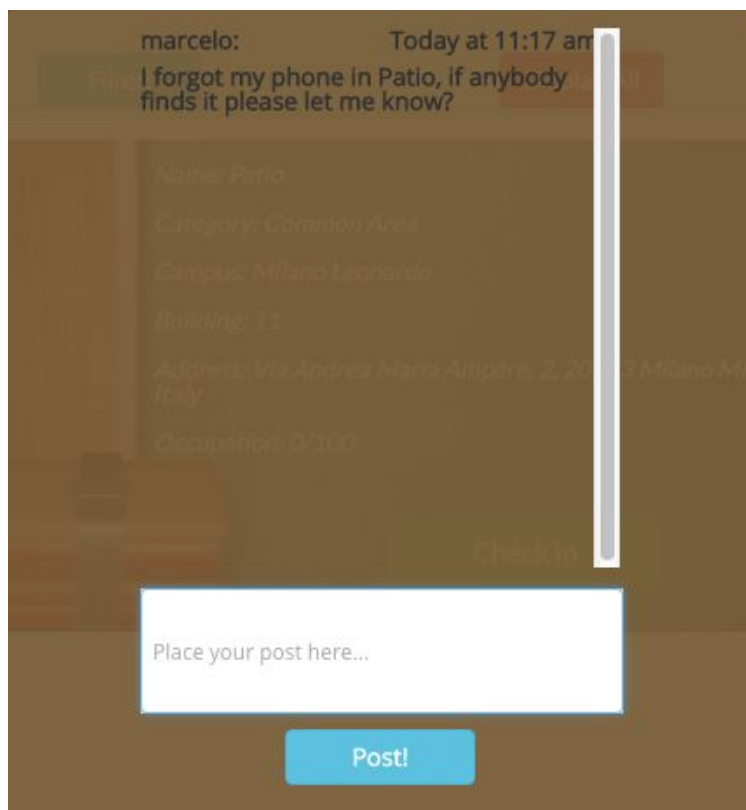
The screenshot shows a web application interface with a brown background. At the top, there are two buttons: "Filter" on the left and "Display All" on the right. Below these, a list of items is displayed, each with a small square icon to its left. The items are: "Name: Patio", "Category: Common Area", "Campus: Milano Leonardo", "Building: 11", "Address: Via Andrea Maria Ampère, 2, 20129 Milano MI, Italy", and "Occupation: 0/100". To the right of the list, there is a vertical scrollbar. Below the list, there is a "Check in" button. At the bottom, there is a white text input field with the placeholder text "Place your post here..." and a blue "Post!" button below it.

➤ **Test case 4**

*<message = "I forgot my phone in Patio, if anybody finds it please let me know?",
student = "marcelo", selected post place = "Patio"*

It is assumed that the current user is "jp"

DELETING FAILS (the delete icon doesn't even appear since students are not allowed to delete other user's messages)



3. Experimental Trials with Students

Before Concluding this document, and therefore all our project work, it is worth mentioning that after all our work on PoliSAM was done, we asked a couple of student volunteers from Polimi to try our Web App and give us their feedback (we were also getting suggestions in the middle of the development process concerning what type of features would be considered interesting for the user). It was quite satisfactory for us to see how positively our colleagues reacted to it, and how helpful they were in their constructive feedback. In fact, all of the students who tried our application found the idea itself attractive as they can relate with us in terms of the motivation behind developing such an interface. Moreover, they found the GUI to be easy and intuitive to understand and operate with; also none of them complained about any problems related to the different modules performing their required functionalities according to the defined specifications. However, most of them thought that the graphical interface could be visually ameliorated (in terms of the front-end graphics); and so this is why we moved from the GUI 2 that was presented in the SDD document, to a more improved version which was displayed in the previous Section of this document.