

## Limiting Controller Access:

```
1 class ApplicationController < ActionController::Base
2   #Functions for the application controller.
3   protect_from_forgery with: :exception
4   #Include SessionHelpers. Allows helpers to be executed.
5   include SessionHelpers
6
7   private
8   def logged_in_user
9     unless logged_in?
10       store_location
11       flash[:danger] = "Please log in."
12       redirect_to login_url
13     end
14   end
15
16   def correct_user
17     @user = User.find(params[:id])
18     redirect_to(root_url) unless current_user?(@user)
19   end
20
21   def admin_user
22     redirect_to(root_url) unless current_user.admin == true
23   end
24
25   def teacher_user
26     redirect_to(root_url) unless current_user.teacher?
27   end
28
29   def student_user
30     redirect_to(root_url) unless !current_user.teacher? && !current_user.admin?
31   end
32 end
```

Figure 1. Definition of functions used to determine before actions

```
1 class FreePlayRoomsController < ApplicationController
2
3   before_action :logged_in_user
4   before_action :student_user, :except => [:index]
5
```

Figure 2. Before actions for Free Play Rooms Controller

```
1 class InterviewsController < ApplicationController
2
3   before_action :logged_in_user
4   before_action :admin_user
5
```

Figure 3. Before actions for Interviews Controller

```
1 class SkillsController < ApplicationController
2
3   before_action :logged_in_user
4   before_action :teacher_user
5
```

Figure 4. Before actions for skills Controller

Limiting controller access is a fundamental aspect of the project. As defined in our users' stories: confidentiality and integrity of users' information are three major security assets that we strive to maintain. In order to insure the confidentiality of students, we want to limit who is able to view certain information. As seen in figure 2, logged in users are unable to view any aspect of the free play room. Unlogged in users are unable to view the times and dates of when students will be in the room. Additionally, teacher and admins are only able to view the index of the free play room. Thus, are unable to edit or delete student bookings – ensuring user integrity of their information. If an unlogged in user attempts to view the free play room controller, or in fact any controller except static pages and new user page, they will be redirected to the login page. Just like with the free play room, any logged in user who attempts to view information not relevant to themselves (unauthorised) will be redirected to the home page. This functions responsible for redirecting certain users are found in the application controller as seen in Figure 1, therefore all controllers in the project inherit these functions.

In addition to the control above, I have implemented other additional control methods to prevent information about users being disclosed to unauthorised users. An example of this is the show page of users. If the correct users is logged in they can see personal information such as email, parent name, etc. However, users are only able to see name, birth date and Facebook address of other users.

## Test fixtures and Helpers:

```
1 # Test users currently used for website.
2 mika:
3   id: 1
4   name: Mika
5   last_name: Williams
6   email: mika@music.com
7   facebook_id: www.facebook.com/mikaMusic
8   address: '450 Queens St, QLD'
9   dob: 11/12/1980
10  gender: Female
11  password_digest: <%= User.digest('password') %>
12  admin: true
13
14 teacher1:
15   id: 2
16   name: Teacher
17   last_name: num1
18   email: teachernum1@example.gov
19   facebook_id: www.facebook.com/teachernum1
20   address: '40283 Dunbar St, Grafton, QLD'
21   dob: 28/08/1989
22   gender: Male
23   password_digest: <%= User.digest('password') %>
24   teacher_qualifications: Graduated Queensland University of Technology
25   user_recieve_emails: true
26   teacher: true
27   admin: false
28
29 teacher2:
30   id: 3
31   name: Teacher
32   last_name: num2
33   email: techteacher@teacher.org
34   facebook_id: www.facebook.com/teacher1
35   address: '100 Railway Ave, Tingalpa, QLD'
36   dob: 05/06/1991
37   gender: Female
38   password_digest: <%= User.digest('password') %>
39   teacher_qualifications: Graduated University Queensland
40   user_recieve_emails: false
41   teacher: true
42   admin: false
43
44 student1:
45   name: student
46   last_name: number1
47   email: student@example.gov
48   facebook_id: www.facebook.com/studentNum1
49   address: '15 Collins St, Boondal, QLD'
50   dob: 10/01/1995
51   gender: Male
52   password_digest: <%= User.digest('password') %>
53   user_recieve_emails: false
54   teacher: false
55   admin: false
```

Figure 5. Test fixture for users

```
1 one:
2   user_id: 2
3   user_email: teachernum1@example.gov
4   day: Monday
5   time: 09:30
6   duration: 2
7
8 two:
9   user_id: 3
10  user_email: techteacher@teacher.org
11  day: Friday
12  time: 09:30
13  duration: 5
```

Figure 6. Test fixture for availabilities

```
1 one:
2   user_id: 4
3   user_email: student@example.gov
4   preferred_day: Monday
5   preferred_time: 12:30
6   instrument: Guitar
7   preferred_teacher_language: English
8   preferred_teacher_gender: Male
9
10 two:
11   user_id: 5
12   user_email: num2@student.gov
13   preferred_day: Friday
14   preferred_time: 30:11
15   instrument: Drums
16   preferred_teacher_language: English
17   preferred_teacher_gender: None
```

Figure 7. Test fixture for preferences

Above are some examples of test fixtures used for this project. As seen in Figure 5 compared to artefact 5, the user fixtures have been significantly modified and updated to reflect plans for release 2. The admin user has been named as Mika and her test credentials have been updated to reflect her. There are two fixtures for teachers, each with slightly different, yet both valid, credentials. This is the same two student fixtures.

Two test fixtures were created for each table/controller created. An example of this can be seen in Figure 6 and 7. Both fixtures for each table/controller are valid however, each fixture belongs to a separate individual. This is so it can be tested whether or not a user can manipulate an entry that does not belong to themselves. The test helper then assigns each of these fixtures to an object, thus allowing each test controller to access these objects without repetitive code.

```
1 ENV['RAILS_ENV'] ||= 'test'
2 require File.expand_path('.../config/environment', __FILE__)
3 require 'rails/test_help'
4 require 'minitest/reporters'
5 Minitest::Reporters.use!
6 class ActiveSupport::TestCase
7   fixtures :all
8   include ApplicationHelper
9
10  def setup
11    @admin = users(:mika)
12    @teacher1 = users(:teacher1)
13    @teacher2 = users(:teacher2)
14    @student1 = users(:student1)
15    @student2 = users(:student2)
16    @availability1 = availabilities(:one)
17    @availability2 = availabilities(:two)
18    @free_play_room1 = free_play_rooms(:one)
19    @free_play_room2 = free_play_rooms(:two)
20    @interview1 = interviews(:one)
21    @interview2 = interviews(:two)
22    @preference1 = preferences(:one)
23    @preference2 = preferences(:two)
24    @skill1 = skills(:one)
25    @skill2 = skills(:two)
26    @base_title = 'Mika Music School'
27  end
28
29  # Returns true if a test user is logged in.
30  def is_logged_in?
31    !session[:user_id].nil?
32  end
33
34  # Log in as a particular user.
35  def log_in_as(user)
36    session[:user_id] = user.id
37  end
38
39  class ActionDispatch::IntegrationTest
40    # Log in as a particular user.
41    def log_in_as(user, password: 'password', remember_me: '1')
42      post login_path, params: { session: { email: user.email,
43                                           password: password,
44                                           remember_me: remember_me } }
45    end
46  end
```

Figure 8. Test Helper

## Test controllers:

```
1 require 'test_helper'
2
3 class InterviewsControllerTest < ActionDispatch::IntegrationTest
4
5   test "admin should get index" do
6     log_in_as(@admin)
7     get my_interviews_path
8     assert_response :success
9     assert_select "title", "My Interviews | #{@base_title}"
10  end
11
12   test "admin should get new" do
13     log_in_as(@admin)
14     get new_interview_path
15     assert_response :success
16     assert_select "title", "New Interview | #{@base_title}"
17  end
18
19   test "admin should get edit" do
20     log_in_as(@admin)
21     get edit_interview_path(@interview1)
22     assert_response :success
23     assert_select "title", "Edit Interview | #{@base_title}"
24  end
25
26   test "admin should get delete" do
27     log_in_as(@admin)
28     get delete_interview_path(@interview1)
29     assert_response :success
30     assert_select "title", "Delete Interview | #{@base_title}"
31  end
32
33   test "teacher should not get index" do
34     log_in_as(@teacher1)
35     get my_interviews_path
36     assert_redirected_to root_url
37  end
38
39   test "teacher should not get new" do
40     log_in_as(@teacher1)
41     get new_interview_path
42     assert_redirected_to root_url
43  end
44
45   test "teacher should not get edit" do
46     log_in_as(@teacher1)
47     get edit_interview_path(@interview1)
48     assert_redirected_to root_url
49  end
50
51   test "teacher should not get delete" do
52     log_in_as(@teacher1)
53     get delete_interview_path(@interview1)
54     assert_redirected_to root_url
55  end
56
57   test "student should not get index" do
58     log_in_as(@student1)
59     get my_interviews_path
60     assert_redirected_to root_url
61  end
62
63   test "student should not get new" do
64     log_in_as(@student1)
65     get new_interview_path
66     assert_redirected_to root_url
67  end
68
69   test "student should not get edit" do
70     log_in_as(@student1)
71     get edit_interview_path(@interview1)
72     assert_redirected_to root_url
73  end
74
75   test "student should not get delete" do
76     log_in_as(@student1)
77     get delete_interview_path(@interview1)
78     assert_redirected_to root_url
79  end
80
81   test "unlogged user should not get index" do
82     get my_interviews_path
83     assert_redirected_to login_url
84  end
85
86   test "unlogged user should not get new" do
87     get new_interview_path
88     assert_redirected_to login_url
89  end
90
91   test "unlogged user should not get edit" do
92     get edit_interview_path(@interview1)
93     assert_redirected_to login_url
94  end
95
96   test "unlogged user should not get delete" do
97     get delete_interview_path(@interview1)
98     assert_redirected_to login_url
99  end
100 end
```

Figure 9. Test Controller for Interviews part 1

```
1 require 'test_helper'
2
3 class AvailabilitiesControllerTest < ActionDispatch::IntegrationTest
4
5   test "should get index" do
6     log_in_as(@teacher1)
7     get my_availabilities_path
8     assert_response :success
9     assert_select "title", "My Availabilities | #{@base_title}"
10  end
11
12   test "should get new" do
13     log_in_as(@teacher1)
14     get new_availability_path
15     assert_response :success
16     assert_select "title", "Add New Availability | #{@base_title}"
17  end
18
19   test "should get edit" do
20     log_in_as(@teacher1)
21     get edit_availability_path(@availability1)
22     assert_response :success
23     assert_select "title", "Edit Availability | #{@base_title}"
24  end
25
26   test "should get delete" do
27     log_in_as(@teacher1)
28     get delete_availability_path(@availability1)
29     assert_response :success
30     assert_select "title", "Delete Availability | #{@base_title}"
31  end
32
33   test "should redirect student to home page on index attempt" do
34     log_in_as(@student1)
35     get my_availabilities_path
36     assert_redirected_to root_url
37  end
38
39   test "should redirect student to home page on new attempt" do
40     log_in_as(@student1)
41     get new_availability_path
42     assert_redirected_to root_url
43  end
44
45   test "should redirect student to home page on edit attempt" do
46     log_in_as(@student1)
47     get edit_availability_path(@availability1)
48     assert_redirected_to root_url
49  end
50
51   test "should redirect student to home page on delete attempt" do
52     log_in_as(@student1)
53     get delete_availability_path(@availability1)
54     assert_redirected_to root_url
55  end
56 end
```

Figure 11. Test Controller for Availabilities

```
50
51 test "student should not get index" do
52   log_in_as(@student1)
53   get my_interviews_path
54   assert_redirected_to root_url
55 end
56
57 test "student should not get new" do
58   log_in_as(@student1)
59   get new_interview_path
60   assert_redirected_to root_url
61 end
62
63 test "student should not get edit" do
64   log_in_as(@student1)
65   get edit_interview_path(@interview1)
66   assert_redirected_to root_url
67 end
68
69 test "student should not get delete" do
70   log_in_as(@student1)
71   get delete_interview_path(@interview1)
72   assert_redirected_to root_url
73 end
74
75 test "unlogged user should not get index" do
76   get my_interviews_path
77   assert_redirected_to login_url
78 end
79
80 test "unlogged user should not get new" do
81   get new_interview_path
82   assert_redirected_to login_url
83 end
84
85 test "unlogged user should not get edit" do
86   get edit_interview_path(@interview1)
87   assert_redirected_to login_url
88 end
89
90 test "unlogged user should not get delete" do
91   get delete_interview_path(@interview1)
92   assert_redirected_to login_url
93 end
94
95 test "unlogged user should not get delete" do
96   get delete_interview_path(@interview1)
97   assert_redirected_to login_url
98 end
99
100 end
```

Figure 10. Test Controller for Interviews part 2

As displayed in Figures 9 and 10, the test controller for Interviews has 16 tests and 20 assertions. The first four tests check to make sure that the admin is able to access all intended pages of the interview controller – on successful access then tests to make sure that the title is correctly displaying the appropriate information. There are then eight tests to test if both teachers and students are unable to access the interview controller and are redirected to the root url. Lastly are four tests, testing if un logged users are redirected to the sign up page.

This pattern is used almost identically for all test controllers, only changing the users that should be able to access that specific controller.