
Unit Test Cases Document

for

PlaceMentor

Version 0

Prepared by 'CodeNatives'

Date : 26/03/2021

Contents

| | |
|---|----|
| 1. Feature: To register an account | 3 |
| 2. Login | 6 |
| 3. Logout | 7 |
| 4. Load Notifications | 7 |
| 5. Fetch User Names | 8 |
| 6. Search for user profile | 10 |
| 7. Writing Messages | 11 |
| 8. Capture Read Status of message/notice | 12 |
| 9. Applying to a position | 13 |
| 10. Uploading resume | 14 |
| 11. Get list of positions for a given recruiter | 15 |
| 12. Releasing updates of positions | 16 |
| 13. Give feedback on the positions | 17 |
| 14. Release Job Profile | 18 |
| 15. Get applications for a position | 18 |
| 16. Accept or Reject Applications | 19 |
| 17. Check if recruiters are genuine | 19 |

1. Feature: To register an account

| | |
|--------------------|---|
| Test Case 1 | Valid Credentials |
| Input | Username: test_user, User Email: test_user@test.com, User Type: Student, Password: test_pswrd |
| Response | HTTP Response Code: 200 Response data: { status: 'success', message: 'User successfully registered' } |
| Comment | This test case checks the scenario when a new user (student or alumni) registers with all valid credentials |

| | |
|--------------------|---|
| Test Case 2 | Valid Credentials for a Company Admin |
| Input | Username: test_comp_admin, User Email: test_admin@test.com, User Type: Institute Admin, Password: test_pswrd Admin_Gen_ID: @admin#compny_pm_001 |
| Response | HTTP Response Code: 200 Response data: { status: 'success', message: 'User successfully registered' } |
| Comment | This test case checks the scenario when a company admin registers with all valid credentials (and have a Genuine id) |

| | |
|--------------------|--|
| Test Case 3 | User name already existed |
| Input | Username: test_user, (that has already been used |

| | |
|-----------------|---|
| | before) User Email: test_user_2@test.com, User Type: Student, Password: test_pswrd |
| Response | HTTP Response Code: 406 Response_object: { status: 'invalid', message: 'User name already been taken' } |
| Comment | This test case checks the scenario when a new user registers with a username that has already been taken by someone. (The <i>test_user</i> has already been used to register in the test case 1, this is not accepted.) |

| | |
|--------------------|---|
| Test Case 4 | User email already existed |
| Input | Username: test_user_2, User Email: test_user@test.com , (that has already been used before) User Type: Alumni |
| Response | HTTP Response Code: 402 Response_object: { status: 'invalid', message: 'Email is already registered' } |
| Comment | This test case checks the scenario when a new user registers with an email that has already been registered with an account. (The <i>test_user</i> has already been used to register in the test case 1, this is not accepted.) |

| | |
|--------------------|---|
| Test Case 5 | Registers with Non Genuine ID |
| Input | Username: test_comp_admin, User Email: test_admin@test.com, User Type: Institute Admin, Password: test_pswrd |

| | |
|-----------------|---|
| | Admin_Gen_ID: fake_gen_id |
| Response | HTTP Response Code: 403 Response_object: { status: 'invalid', message: 'ID is not genuine']' } |
| Comment | This test case checks the scenario when a new company admin registers with a genuine id that has already been registered with an account. (The <i>genuine_id</i> has already been used to register in the test case 2, this is not accepted.) |

| | |
|--------------------|---|
| Test Case 6 | Admin already exists with that Genuine ID |
| Input | Username: test_comp_admin_2, User Email: test_admin_2@test.com, User Type: Institute Admin, Password: test_pswrd Admin_Gen_ID: @admin#compny_pm_001 |
| Response | HTTP Response Code: 409 Response_object: { status: 'invalid', message: 'Admin already exists with that Genuine ID' } |
| Comment | This test case checks the scenario when a new company admin registers with a genuine id that has already been registered with an account. (The <i>genuine_id</i> has already been used to register in the test case 2, this is not accepted.) |

2. Login

| | |
|--------------------|--|
| Test Case 1 | Correct Credentials |
| Input | Username: test_user, Password: test_pswrd |
| Response | HTTP Response Code: 200 Login Details: { User_ID: 1, username: test_user } |
| Comment | This test case checks the scenario when a registered user logs in with correct credentials. (Necessary users are registered before running the tests.) |

| | |
|--------------------|--|
| Test Case 2 | Incorrect Credentials |
| Input | Username: test_user, Password: test_incrct_pswrd |
| Response | HTTP Response Code: 401 Response_object: { status: 'fail', message: 'Incorrect Password' } |
| Comment | This test case checks the scenario when a registered user logs in with in-correct credentials (password). (Necessary users are registered before running the tests.) |

| | |
|--------------------|---|
| Test Case 3 | User does not exist |
| Input | Username: test_nonexist_user, Password: test_pswrd |
| Response | HTTP Response Code: 404 Response_object: { status: 'fail', message: 'No user with that username' |

| | |
|----------------|--|
| | } |
| Comment | This test case checks the scenario when a user logs in with username that has not been registered. |

3. Logout

| | |
|--------------------|---|
| Test Case 1 | Logout of Logged-in user |
| Input | --- |
| Response | HTTP Response Code: 200 Response_object: { status: 'success', message: 'User successfully logged out' } |
| Comment | This test case checks the scenario when a logged in user logs out. (Necessary users are registered and logged in before running the tests.) |

4. Load Notifications

| | |
|--------------------|---|
| Test Case 1 | Load Notifications for a Logged-in user |
| Input | --- |
| Response | HTTP Response Code: 200 Notifications_list: [{ title: "Title 1", time: "08:23 AM", content: "Content 1", sender: "Institute Admin", type_of_notification: Notice, id: 1, (Id of the notice) },] |

| | |
|----------------|---|
| | <pre> { title: "Title 3", time: "06:18 AM", content: "Content 3", sender: "Company_1", type_of_notification: Job Position, id: 1, (Id of the Job Position) },] </pre> <p>(sorted by date of arrival of notification)</p> |
| Comment | <p>This test case checks the scenario of loading notifications when a user is logged in. (Necessary users are registered and logged in and certain messages have already been sent and some of them are read as marked before running the tests.)</p> <p>For this test case, Notification with title 2 has been read by user and not loaded as notification</p> |

5. Fetch User Names

| | |
|--------------------|---|
| Test Case 1 | Search for users based on username, when the user is logged in |
| Input | Name_Search_Query: "Alum" |
| Response | <p>HTTP Response Code: 200</p> <p>Username_lists: [</p> <p style="padding-left: 40px;">"Alumni_1",</p> <p style="padding-left: 40px;">"Compaluminate"</p> <p>]</p> <p>(sorted in alphabetical order)</p> |
| Comment | <p>This test case checks the scenario of fetching list of usernames matching search query when a user is logged in. (Necessary users (with usernames) are registered and the requesting user is logged in</p> |

| | |
|--------------------|--|
| Test Case 2 | Search for users based on a valid user type, when the user is logged in |
| Input | Type_Search_Query: Company Admin |
| Response | HTTP Response Code: 200 Username_lists: ["Comp_admin_1", "Comp_admin_2",] (sorted in alphabetical order) |
| Comment | This test case checks the scenario of fetching list of usernames matching search query when a user is logged in. (Necessary users (with usernames) are registered and the requesting user is logged in |

| | |
|--------------------|---|
| Test Case 3 | Search for users based on a invalid user type, when the user is logged in |
| Input | Type_Search_Query: Professor |
| Response | HTTP Response Code: 400 Response_object: { status: 'fail', message: 'Invalid search Criteria' } |
| Comment | This test case checks the scenario of fetching list of usernames when an invalid search query when a user is logged in. (Necessary users (with usernames) are registered and requesting user is logged in |

6. Search for user profile

| | |
|--------------------|---|
| Test Case 1 | Search for user profile based on username, when the user is logged in |
| Input | Name_Search_Query: "Alumni_1" |
| Response | HTTP Response Code: 200 profile: [grad_year: 2012, Companies Worked for: ["Comp_1", "Comp_2"], degree: "B.Tech", dept: "CS",] |
| Comment | This test case checks the scenario of fetching the profile details(that are made public) of a given username (if it is public) when a user is logged in. (Necessary users (here Alumni_1) and privacy options are set(user email: private, else all are public) are registered and the requesting user is logged in |

| | |
|--------------------|--|
| Test Case 2 | Search for user profile (which is not accessible) based on username, when the user is logged in |
| Input | Name_Search_Query: "Student_1" |
| Response | HTTP Response Code: 200 Profile: {} |
| Comment | This test case checks the scenario of fetching the profile details(all are made public) of a given username , when a user is logged in. (Necessary users (here Student_1) and is made private to the requesting user type(Students) are registered and the requesting user(Student_2) is logged in |

| | |
|--------------------|--|
| Test Case 3 | Search for user profile of a username that does not exist |
| Input | Name_Search_Query: "Student_a" |
| Response | HTTP Response Code: 404 Response_object: { status: 'invalid', message: 'No user found with the username' } |
| Comment | This test case checks the scenario of searching the profile of a given username (that does not exist), when a user is logged in. (Necessary users are registered and the requesting user(Student_2) is logged in |

7. Writing Messages

| | |
|--------------------|--|
| Test Case 1 | Valid Recipient |
| Input | Recipient Username: "test_username", Message Content: "test content of the message", Date: 26-03-2021 |
| Response | HTTP Response Code: 200 Message: { Recipient Username: "test_username", Content: "test content of the message" } |
| Comment | This test case is to check if messages are being sent to other users. |

| | |
|--------------------|--|
| Test Case 2 | Invalid Recipient Type |
| Input | Recipient Username: "Company_1" Message Content: " test content of the message" |

| | |
|-----------------|--|
| Response | HTTP Response Code: 404 Response data: { status: 'invalid', message: 'Messages cannot be sent to this user type' } |
| Comment | This test case is to check whether the system shows an error if a user of user_type1 tries to send a message to another user of user_type2 (whom he cannot send a message). Example: a student to a company admin |

| | |
|--------------------|---|
| Test Case 3 | Non-existing recipient name |
| Input | Recipient Username: "test_non_existent_username" Message Content: "test content of the message" |
| Response | HTTP Response Code: 404 Response data: { status: 'fail', message: 'No user found with the username' } |
| Comment | This test case checks if the system shows an error if a user tries to send a message to a non-existent username. |

8. Capture Read Status of message/notice

| | |
|--------------------|---|
| Test Case 1 | Valid Message |
| Input | Message Id: 1 |
| Response | HTTP Response Code: 200 Response data: { status: 'success', |

| | |
|----------------|---|
| | message: 'Successfully captured Read Status' } |
| Comment | This test case checks if a user is able to check the read-status of his/her message and if the read status is accurate. |

| | |
|--------------------|---|
| Test Case 2 | Invalid Recipient |
| Input | Message Id: -1 |
| Response | HTTP Response Code: 404 Response data: { status: 'invalid', message: 'Cannot find a message with that id' } |
| Comment | This test case checks if a user is able to check the read-status of his/her message and if the read status is accurate. |

9. Applying to a position

| | |
|--------------------|--|
| Test Case 1 | Apply to a job profile before the deadline |
| Input | positionId = 1 Upload Resume: test_file.pdf Date = 26-03-2021 |
| Response | HTTP Response Code: 200 Response data: { status: 'success', message: 'Succesfully spplied to the Job position' } |
| Comment | This test case checks if a student is able to search for job positions and apply to a job position. Tested |

| | |
|--|--|
| | whether The application is sent to the respective company admin. |
|--|--|

| | |
|--------------------|---|
| Test Case 2 | Apply to a job profile post-deadline |
| Input | positionId = 1 Upload Resume: test_file.pdf Date = 26-03-2021 |
| Response | HTTP Response Code: 406 Response data: { status: 'fail', message: 'Deadline of the application expired' } |
| Comment | This test case checks if the system shows an error if a student tries to apply for a job position whose deadline is passed. |

| | |
|--------------------|--|
| Test Case 3 | Invalid Job Position |
| Input | Job Position = -1 |
| Response | HTTP Response Code: 404 Response data: { status: 'invalid', message: 'No Job Position found with the ID' } |
| Comment | This test case checks if no job offers are shown when a user searches for an invalid job position. |

10. Uploading resume

| | |
|--------------------|---------------------------|
| Test Case 1 | Resume in a proper format |
|--------------------|---------------------------|

| | |
|-----------------|---|
| Input | Resume file: test_file.pdf |
| Response | HTTP Response Code: 200 Response data: { status: 'success', message: 'Successfully uploaded the resume' } |
| Comment | This test case checks if a student is able to submit his/her resume (pdf file). Tested whether the resume file will be uploaded to the database. |

| | |
|--------------------|---|
| Test Case 2 | Resume in an improper format |
| Input | Resume file: test_file.png |
| Response | HTTP Response Code: 404 Response data: { status: 'invalid', message: 'File of this format cannot be uploaded as a resume' } |
| Comment | This test case checks if the system detects improper file format of a resume and rejects the submission. |

11. Get list of positions for a given recruiter

| | |
|--------------------|--|
| Test Case 1 | Valid Recruiter |
| Input | Company_ID = 2 |
| Response | HTTP Response Code: 200 Job Positions: [{ profileName = "SE_Lead", |

| | |
|----------------|---|
| | <pre> companyName = "ABC", deadline = 27-03-2021, CTC = 1200000, Release Date = 13-03-2021, Description = "description", Dept = ['CS','EC'] Degree = ['B.Tech',' M.Tech'] }, { profileName = "SE_Manager", companyName = "ABC", deadline = 27-03-2021, CTC = 2400000, Release Date = 13-03-2021, Description = "description", Dept = ['CS','EC'] Degree = ['B.Tech',' M.Tech'] },]</pre> |
| Comment | This test case checks if the search operation for recruiters is properly working or not. |

12. Releasing updates of positions

| | |
|--------------------|---|
| Test Case 1 | Valid Job Position |
| Input | positionId = 1 |
| Response | HTTP Response Code: 200 Response data: { status: 'success', message: 'Position successfully released' } |
| Comment | This test case checks if an update associated with the job position is released into the InstiBoard. |

| | |
|--------------------|----------------------|
| Test Case 2 | Invalid Job Position |
| Input | positionId = -1 |

| | |
|-----------------|---|
| Response | HTTP Response Code: 404 Response data: { status: 'invalid', message: 'Position not found' } |
| Comment | This test case checks if an update associated with the job position is released into the InstiBoard. |

13. Give feedback on the positions

| | |
|--------------------|---|
| Test Case 1 | Valid Job Position |
| Input | positionId = 1, date: 29-03-2021, author: alum_X, content: "XYZ", |
| Response | HTTP Response Code: 200 Response data: { status: 'success', message: 'Feedback successfully posted' } |
| Comment | This test case checks whether the feedback is sent to the associated job profile. |

| | |
|--------------------|---|
| Test Case 2 | Invalid Job Position |
| Input | positionId = 1, date: 29-03-2021, author: alum_X, content: "XYZ", |
| Response | HTTP Response Code: 404 Response data: { status: 'invalid', message: 'Position not found' } |
| Comment | This test case assures that feedback sent to such job positions shall not be accepted. |

14. Release Job Profile

| | |
|--------------------|---|
| Test Case 1 | Store job profile in the system |
| Input | profileName = "SE_Lead", companyName = "ABC", deadline = 27-03-2021, CTC = 1200000, Release Date = 13-03-2021, Description = "description", Dept = ['CS','EC'] Degree = ['B.Tech',' M.Tech'] |
| Response | HTTP Response Code: 200 Response data: { status: 'success', message: 'Job Profile successfully released' } |
| Comment | This test case checks whether the job position is released for all the users. |

15. Get applications for a position

| | |
|--------------------|--|
| Test Case 1 | Valid Job Position |
| Input | positionId = 1 |
| Response | HTTP Response Code: 200 Applicants: [Student_1, Student_2] (<list> of Job Applicants) |
| Comment | This test case successfully displays all the applicants associated with a job position. |

| | |
|--------------------|--|
| Test Case 2 | Invalid Job Position |
| Input | positionId = -1 |
| Response | HTTP Response Code: 404 Response data: { status: 'invalid', message: 'Position not found' |

| | |
|----------------|--|
| | } |
| Comment | This test case assures that job profiles which are: a) not stored in the system OR b) not yet released shall not have any applicants. |

16. Accept or Reject Applications

| | |
|--------------------|---|
| Test Case 1 | Select/Reject an applicant |
| Input | Application_id = 1 |
| Response | HTTP Response Code: 200 Response data: { status: 'success', message: 'Succesfully updates the status' } |
| Comment | This test case checks if the student is notified about his/her application status. |

17. Check if recruiters are genuine

| | |
|--------------------|--|
| Test Case 1 | Genuine Recruiter |
| Input | Govt_Id: type <String> |
| Response | True |
| Comment | This test case assures that a given recruiter is valid by checking the provided ID with the data stored. |

| | |
|--------------------|--|
| Test Case 2 | Non-genuine recruiter |
| Input | Govt_Id: type <String> |
| Response | False |
| Comment | This test case assures that a given recruiter is invalid by checking the provided ID with the data stored. |

