Testing was done on release 3 11/03/2020  
Testing environment was done from a completely blank database testing the possible regenerative properties of the application.

All test where done with one a pc running Firefox (user 1) and a mobile phone running google chrome (user 2).

Both instances of browsers where launched in after removing cookies.

User 1: [harrisonjordanjames@gmail.com](mailto:harrisonjordanjames@gmail.com)

User2: jordanharrisontester@gmail.com

Testing was done on the “live” environment.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Task | Expected Result | Actual Result | Pass/Fail |
| 1 | User Login | Google authentication triggers state change and triggers page load of teams, tasks and notifications. Database object, representing the user is created along with the accounts collection. | As expected for both users in test | Pass |
| 2 | Cookie policy message acceptance | Cookie policy triggers on the initial load. Any subsequent refreshes trigger auto login and no cookie message appears. | As expected for both users in test | Pass |
| **1 Creating Teams** | | | | |
| 1a | Create New Team on user 1, With the following values:  name: test team 1  Description: test team 1  Allow Members to add tasks?: true  Allow Members to Edit Team?: true  Allow Members to Invite?: true  Personal skills: null | Team is created along with the teams collection. Users 1 team list updates the DOM user 2 receives no change  Team is created with following values:  name: test team 1  Description: test team 1  Allow Members to add tasks?: true  Allow Members to Edit Team?: true  Allow Members to Invite?: true  Personal skills: null  Owner: User1id  Members:{User1id} | As expected for both users in test  Data on database is as expected | Pass |
| 1b | Create New Team on user 2, With the following values:  name: test team 2  Description: test team 2  Allow Members to add tasks?: true  Allow Members to Edit Team?: true  Allow Members to Invite?: true  Personal skills: null | Team is created along with the teams collection. Users 2 team list updates the DOM user 1 receives no change  Team is created with following values:  name: test team 2  Description: test team 2  Allow Members to add tasks?: true  Allow Members to Edit Team?: true  Allow Members to Invite?: true  Personal skills: null  Owner: User2id  Members:{User2id} | As expected for both users in test  Data on database is as expected | pass |
| 1c | Create New Team user2, with the following values:  name: test team 3  Description: test team 3  Team type: vertical  Allow Members to add tasks?: false  Allow Members to Edit Team?: true  Allow Members to Invite?: true  Personal skills: { test skill 1} | Team is created under the teams collection. User 2 team list updates the DOM user 1 receives no change  Team is created with following values:  name: test team 3  Description: test team 3  Allow Members to add tasks?: false  Allow Members to Edit Team?: true  Allow Members to Invite?: true  Personal skills: { test skill 1}  Owner: User2id  Members:{User2id} | As expected for both users in test  Data on database is as expected | Pass |
| 1d | Create New Team user2, with the following values:  name: test team 4  Description: test team 4  Team type: vertical  Allow Members to add tasks?: false  Allow Members to Edit Team?: false  Allow Members to Invite?: true  Personal skills: null | Team is created under the teams collection. User 2 team list updates the DOM user 1 receives no change  Team is created with following values:  name: test team 4  Description: test team 4  Allow Members to add tasks?: false  Allow Members to Edit Team?: false  Allow Members to Invite?: true  Personal skills: null  Owner: User2id  Members:{User2id} | As expected for both users in test  Data on database is as expected |  |
| 1e | Create New Team user2, with the following values:  name: test team 5  Description: test team 5  Team type: vertical  Allow Members to add tasks?: false  Allow Members to Edit Team?: false  Allow Members to Invite?: false  Personal skills: null | Team is created under the teams collection. Users 2 team list updates the DOM user 1 receives no change  Team is created with following values:  name: test team 5  Description: test team 5  Allow Members to add tasks?: false  Allow Members to Edit Team?: false  Allow Members to Invite?: false  Personal skills: null  Owner: User2id  Members:{User2id} | As expected for both users in test  Data on database is as expected |  |
| **2 Inviting to team** | | | | |
| 2a | Team Invite – On user2 scan user 1s QR code and trigger invite to “test team 2”. Close the invite form. | User 1 receives an invite notification, invite triggers notify to popup, notification list has a new invite card on it. Database has notifications collection with a notification with the following fields:  For:user1id  is-read false  message "you have been invited to: test team 2"  team "teams/testteam2id"  type "team-invite"  test team 2 data has: pending-invites:{user1id} | As expected for both users in test  Data on database is as expected | pass |
| 2b | Team Invite – On user2 scan user 1s QR code and trigger invite to “test team 3”. Close the invite form. | User 1 receives an invite notification, invite triggers notify to popup, notification list has a new invite card on it. Database has notifications collection with a notification with the following fields:  For:user1id  is-read false  message "you have been invited to: test team 3"  team "teams/testteam3id"  type "team-invite"  test team 3 data has: pending-invites:{user1id} | As expected for both users in test  Data on database is as expected | pass |
| 2c | Team Invite – On user2 scan user 1s QR code and trigger invite to “test team 4”. Close the invite form. | User 1 receives an invite notification, invite triggers notify to popup, notification list has a new invite card on it. Database has notifications collection with a notification with the following fields:  For:user1id  is-read false  message "you have been invited to: test team 4"  team "teams/testteam4id"  type "team-invite"  test team 4 data has: pending-invites:{user1id} | As expected for both users in test  Data on database is as expected | pass |
| 2c | Team Invite – On user2 scan user 1s QR code and trigger invite to “test team 5”. Close the invite form. | User 1 receives an invite notification, invite triggers notify to popup, notification list has a new invite card on it. Database has notifications collection with a notification with the following fields:  For:user1id  is-read false  message "you have been invited to: test team 4"  team "teams/testteam5id"  type "team-invite"  test team 5 data has: pending-invites:{user1id} | As expected for both users in test  Data on database is as expected | pass |
| 2d | Accept Invite to “test team 2” on user 1 | User 1: team page updates to now contain “test team 4”. User 1can edit the team and can invite new users. And can add and edit tasks.  User 2: team page updates to now contain “test team 4”. User 2 can edit the team and invite new users. And can add and edit tasks.  test team 2 now has: members = {user2id, user1id} | As expected for both users in test  Data on database is as expected | pass |
| 2e | Accept Invite “test team 3” on user 1 | User 1: team page updates to now contain “test team 4”. User 1can edit the team and can invite new users. User 1 **cannot** add new tasks.  User 2: team page updates to now contain “test team 4”. User 2 can edit the team and invite new users. And can add and edit tasks.  test team 4 now has: members = {user2id, user1id} | As expected for both users in test  Data on database is as expected | pass |
| 2f | Accept Invite “test team 4” on user 1 | User 1: team page updates to now contain “test team 4”. User 1 **cannot** edit the team but can invite new users. User 1 **cannot** add new tasks.  User 2: team page updates to now contain “test team 4”. User 2 can edit the team and invite new users. And can add and edit tasks.  test team 4 now has: members = {user2id, user1id} | As expected for both users in test  Data on database is as expected | pass |
| 2g | Accept Invite “test team 5” on user 1 | User 1: team page updates to now contain “test team 5”. User 1 **cannot** edit the team or invite new users. User 1 **cannot** add or edit any tasks.  User 2: team page updates to now contain “test team 5”. User 2 can edit the team and invite new users. And can add and edit tasks.  test team 4 now has: members = {user2id, user1id} | As expected for both users in test  Data on database is as expected | pass |
| **3 Task Creation** | | | | |
| 3a | Both users view test team 2 tasks  User 1 Add new task to test team 2 with the following values:  Name: task test 1;  Deadline:11/03/2020  Urgency: 1  Impact: 1  Importance: 1  Requirements:  Intelligence: 0,  strength: 0 | Both User 1 and user 2 receive DOM update to their tasks list.  User 1 **cannot** edit the task but User 2 can. Task collection is created under test team 2  New task object with the following parameters:  Name: task test 1;  Deadline:11/03/2020 at 12AM  Urgency: 1  Impact: 1  Importance: 1  Requirements:  Intelligence: 0,  strength: 0  status: Active;  experience-rewards:  Intelligence: 1935,  Strength: 1935  Team: testteam2id  Owner: User1id | As expected for both users in test  Data on database is as expected | pass |
| 3b | Both users view test team 2 tasks  User 2 add new task to test team 2 with the following values:  Name: task test 2;  Deadline:11/03/2020  Urgency: 1  Impact: 1  Importance: 1  Requirements:  test skill 1: 0, | Both User 1 and user 2 receive DOM update to their tasks list.  Both users should be able to edit the task Task collection is created under test team 2  New task object with the following parameters:  Name: task test 2;  Deadline:11/03/2020 at 12AM  Urgency: 1  Impact: 1  Importance: 1  Requirements: {test skill 1: 0},  status: Active;  experience-rewards: {test skill 1: 1935},  Team: testteam2id  Owner: User2id | As expected for both users in test  Data on database is as expected | pass |
| 3c | Both users view test team 3 tasks  User 2 Add new task to test team 3 with the following values:  Name: task test 3;  Deadline:11/03/2020  Urgency: 1  Impact: 1  Importance: 1  Requirements:  Intelligence: 0,  strength: 0 | Both User 1 and user 2 receive DOM update to their tasks list. Task collection is created under test team 3  New task object with the following parameters:  Name: task test 3;  Deadline:11/03/2020 at 12AM  Urgency: 1  Impact: 1  Importance: 1  Requirements:  Intelligence: 0,  strength: 0  status: Active;  experience-rewards:  Intelligence: 1935,  Strength: 1935  Team: testteam3id | As expected for both users in test  Data on database is as expected |  |
| 3c | Both users view test team 3 tasks  User 2 Add new task to test team 3 with the following values:  Name: task test 4;  Deadline:11/03/2020  Urgency: 1  Impact: 1  Importance: 1  Requirements:  Intelligence: 4,  strength: 4 | Both User 1 and user 2 receive DOM update to their tasks list.  User 1 **cannot** edit the task but User 2 can. Task collection is created under test team 3  New task object with the following parameters:  Name: task test 4;  Deadline:11/03/2020 at 12AM  Urgency: 1  Impact: 1  Importance: 1  Requirements:  Intelligence: 4,  Strength: 4  status: Active;  experience-rewards:  Intelligence: 2012,  Strength: 2012  Team: testteam3id | As expected for both users in test  Data on database is as expected | pass |
| 3d | User 1 navigate to team 1 and add new task to test team 3 with the following values:  Name: task test 5;  Deadline:11/03/2020  Urgency: 1  Impact: 1  Importance: 1  Requirements:  Intelligence: 4,  strength: 4 | Only User 1 can access test team 1 and any tasks Task collection is created under test team 1  New task object with the following parameters:  Name: task test 5;  Deadline:11/03/2020 at 12AM  Urgency: 1  Impact: 1  Importance: 1  Requirements:  Intelligence: 4,  strength: 4  status: Active;  experience-rewards:  Intelligence: 2012,  Strength: 2012  Team: testteam1id | As expected for both users in test  Data on database is as expected | pass |
| **4 Task Completion** | | | | |
| 4a | User 2 attempt to complete task test4. | Message appears to say the requirements havn’t been met. Task object doesn’t change, user object doesn’t change. | As expected for both users in test  Data on database is as expected | pass |
| 4b | User 2 complete task test1 navigate to user experience bars. | Both users have task test 1 remove from the DOM.  User 2 db object now has :  skill-levels:  Intelligence: 1935,  Strength: 1935    User page bars show:  Intelligence level 4 at 335/2500  Strength level 4 at 335/2500  Task changes:  Status: complete  Complete by: User2id | As expected for both users in test  Data on database is as expected | pass |
| 4c | User 1 complete task test 2 navigate to user experience bars. | Both users have task test 2 remove from the DOM.  User 1 db object now has :  skill-levels:  Test skill 1: 1935    User page bars show:  Test skill 1 level 4 at 335/2500  Task changes:  Status: complete  Complete by: User1id | As expected for both users in test  Data on database is as expected |  |
| 4d | User 1 complete task test 3 navigate to user experience bars. | Both users have task test 3 remove from the DOM.  User 2 db object now has :  skill-levels:  Intelligence: 1935,  Strength: 1935  Agility : 1935,  Test skill 1: 1935    User page bars show:  Intelligence level 4 at 335/2500  Strength level 4 at 335/2500  Agility level 4 at 335/2500  Test skill 1 level 4 at 335/2500  Task changes:  Status: complete  Complete by: User1id | As expected for both users in test  Data on database is as expected | pass |
| 4e | User 1 complete task test 4 navigate to user experience bars. | Requirements met task should now complete.  Both users have task test 4 remove from the DOM.  User 2 db object now has :  skill-levels:  Intelligence: \*\*,  Strength: \*\*  Agility : 1935,  Test skill 1: 1935    User page bars show:  Intelligence level 4 at 335/2500  Strength level 4 at 335/2500  Agility level 4 at 335/2500  Test skill 1 level 4 at 335/2500  Task changes:  Status: complete  Complete by: User1id | As expected for both users in test  Data on database is as expected |  |