# Away Team Test Cases

| Status | Test Case Name | Supporting tests |
|--------|----------------|------------------|
| Passed | Login | Successful login and follow on token authentication<br>Android test cases #1-6 |
| Passed | Create User | New users registered with form validation<br>Server test cases #3<br>Android test cases #8-15 |
| Passed | Get User | Successfully retrieve and display user info<br>Server test case #3<br>Android test case #20 |
| Passed | Create Managed Team | Create private teams and assigned as manager<br>Android test case #18 |
| Passed | Create Unmanaged Team | Create unmanaged team complete<br>API test case #7<br>Android test case #18 |
| Passed | Handle Access Removal | Access removal tested<br>Android test cases #19,21 |
| Passed | Join Public Team | Able to view and join public team<br>API test case #17<br>Android test case #19 |
| Passed | Joining Private Team | Able to view, submit application, manager accept or deny application, and view private team details after accepted<br>Android test case #19 |
| Passed | Expense Retrieval | Able to retrieve user expenses for team, display details and retrieve receipt image<br>Android test case #23 |

# API Level Tests

| # | Status | Name | Explanation |
|---|--------|------|-------------|
| 1 | Pass | testUserInit | test initialization of user model in API |
| 2 | Pass | testPasswordHashAccuracy | tests generation of scripts and validate password hash |
| 3 | Pass | testGetUser | Inserts User into DB then Pulls User from DB to verify Object is Equal |
| 4 | Pass | testLoginIDExist | User is entered into db first, then test function checks function to see if a username exists in the system or not. |
| 5 | Pass | testEmailExist | User is entered into db first with specific email address, then test function checks to see if an email address is already used in the system or not |
| 6 | Pass | testTeamInit | test initialization of team model in API |
| 7 | Pass | testInsertUnmanagedTeam | test insert team function using an unmanaged team |
| 8 | Pass | testInsertManagedTeam | test insert team function using a managed team |
| 9 | Pass | testSelectAllTeams | test selection of all teams |
| 10 | Pass | testSelectTeamFromId | test select team based on team id |
| 11 | Pass | testSelectTeamFromTeamName | test select team based on team name |
| 12 | Pass | testGetTeamList | test getting team list for a specific user |
| 13 | Pass | testSearchTeams | test the ability to search teams based on team name consisting some string phrase |

| 14 | Pass | testIsTeamManaged | test the ability to determine if a team is managed |
|----|------|-------------------|-----------------------------------------------------|
| 15 | Pass | testModifyTeamModel | test modifying team |
| 16 | Pass | testModifyTeamNameModel | test modify name of team |
| 17 | Pass | testDeleteTeam | test team deletion |
| 18 | Pass | testTeamMembersInit | test initialization of team members in API |
| 19 | Pass | testInsertTeamMember | test inserting team member into database. Also indirectly tests AddFirstTeamMember function as the creator of the team is the first team member |
| 20 | Pass | testAddTeamMemberNonManager | test adding a user to a team that is already created |
| 21 | Pass | testSelectTeamMemberFromId | test selecting a team member based on teamMemberId |
| 22 | Pass | testSelectTeamMemberFromTeamId | test selecting a team member based on team id |
| 23 | Pass | testModifyTeamMember | test the ability to modify a team member |
| 24 | Pass | testModifyManagerAttribute | test the ability to modify the manager attribute of a team member |
| 25 | Pass | testModifyPendingApproval | test the ability to modify the pending approval of a team member |
| 26 | Pass | testModifyTeamMemberTeamId | test the ability to modify the team id of a team member |
| 27 | Pass | testVerifyTeamMemberExist | test the ability to verify if a team member exists |
| 28 | Pass | testDeleteTeamMember1 | test the ability to delete a team member |
| 29 | Pass | testDeleteTeamMember2 | test the ability to delete a team member |

| 30 | Pass | testDeleteTeamMember3 | test the ability to delete a team member where team member is the last one, so the member isn't deleted instead it will trigger a pop warning to end user. Warning will warn user team will be dleted |
|---|---|---|---|
| 31 | Pass | testDeleteTeamMemberTeamRemove | test the ability to delete a team due to no team members |
| 32 | Pass | testVerifyManagerForUser | test the ability to verify if user is manager or not |
| 34 | Pass | testGetNumberOfTeamMembersRemaining | test the ability to get number of team members left on a team |
| 35 | Pass | testGetNumberOfTeamManager | test the ability to get the number of managers that are part of a team |
| 36 | Pass | testInsertTeamTask | test the ability to insert a team task |
| 37 | Pass | testModifyTeamTask | test the ability to modify a team task |
| 38 | Pass | testMarkTeamTaskComplete | test the ability to mark a task complete |
| 40 | Pass | testDeleteTeamTask | test the ability to delete a task |
|  | Pass | testSelectTeamTasks | test the ability to select team tasks |
| 42 | Pass | testValidateDateTime0 | test the ability to validate a date time for an event. This test the case when date time is valid. |
| 43 | Pass | testValidateDateTime1 | test the ability to validate a date time for an event. This test the case when date time is invalid |
| 44 | Pass | testInsertEvent | test the ability to insert an event |
| 45 | Pass | testSelectAllEvents | test the ability to select all events |
| 46 | Pass | testSelectEventFromEventId | test the ability to select events from event id |
| 47 | Pass | testSelectEventFromEventName | test the ability to select events from event name |
| 48 | Pass | testModifyEvent | tests the ability to modify an event |

| 49 | Pass | testModifyEventName | test the ability to modify event name |
| 50 | Pass | testDeleteEvent | test the ability to delete an event |

# Android Tests

| # | Status | Name | Explanation |
|---|--------|------|-------------|
| 1 | Pass | testLoginFormVisibility | display is formatted correctly |
| 2 | Pass | testPasswordValidation | client side validation complete |
| 3 | Pass | testUsernameValidation | client side validation complete |
| 4 | Pass | testAttemptCreate | accepts commands to create a new account |
| 5 | Pass | testAttemptLogin_incorrect | all login failures caught |
| 6 | Pass | testAttemptLogin_correct | logs in when given valid account credentials |
| 7 | Pass | testPagerNavigation | tab navigation is set up and displays correctly |
| 8 | Pass | testAttemptCreateLogin | creates new users |
| 9 | Pass | testAttempteCreate_invalid | checks for username and email errors |
| 10 | Pass | testLoginFormVisibility | form is laid out correctly |
| 11 | Pass | testNameInterpreter1 | able to parse "John Doe" |
| 12 | Pass | testNameInterpreter2 | able to parse "John A Doe" |
| 13 | Pass | testNameInterpreter3 | able to parse "Doe, John A" |
| 14 | Pass | testNameInterpreter4 | able to parse "Johnny" |
| 15 | Pass | testInputValidation | implemented |
| 16 | Pass | testAuthentication | successfully passed server authentication token |
| 17 | Pass | testRememerLogin | remembered user for fast login and forget user after logout |
| 18 | Pass | testCreateTeam | ensure that a new team can be created and displayed correctly |
| 19 | Pass | testJoinTeam | ensure that a new team can be joined and displayed correctly |
| 20 | Pass | testEditAccount | test that user information and password can be changed |

| 21 | Pass | testTeamSpinner | test that changing teams successfully updates the display |
| 22 | Pass | testLogout | when user logs out, all info is removed |
| 23 | Pass | testExpenseReport | collects all expenses for a member on a team |

# Web Client Tests

| # | Status | Name | Explaination |
|---|--------|------|--------------|
| 1 | Pass | testLoginFormVisibility | Login/Logout buttons are displayed correctly based on login state. |
| 2 | Pass | testPasswordValidation | Client side validation works correctly |
| 3 | Pass | testUsernameValidation | Client side validation works correctly |
| 5 | Pass | testAttemptLogin | User is able to correctly login |
| 6 | Pass | testPagerNavigation | Tab navigation on the page works correctly and displays the correct URL. |
| 7 | Pass | testAttemptCreateLogin | User can correctly register a user |
| 8 | Pass | testUniqueUsername | Registration form validates that the username is not already in use |
| 9 | Pass | testEditUser | User can correctly edit their account information. |
| 10 | Pass | testGetUsersTeam | Client successfully pulls a list of teams that are assigned to the user that is currently logged in. |
| 11 | Pass | testCreateTeam | User can successfully create team after he is presented with the form to do so. |
| 12 | Pass | testGetTeam | User can successfully select the team which they would like to view/edit from the dropdown in the webclient's header |
| 13 | Pass | testChangePassword | User can successfully change their password with the given popup interface |
| 14 | Pass | testJoinTeam | User can successfully join a team through the web interface and have it confirmed through notifications and the presence of the team in the user's team dropdown. |
| 15 | Pass | testCreateExpense | User can successfully create an expense for a team he is a member of via the web client. |
| 16 | Pass | testGetTeamExpenses | The web client successfully displays the current expenses on the selected team on the Team Expense page. |

| 17 | Pass | testDeleteTeamExpense | The user is able to successfully delete a selected expense from the list displayed via the web client. |
|----|------|----------------------|-------------------------------------------------------------------------------------------------------|
| 18 | Pass | testModifyExpense | The user is able to successfully modify the description, amount, and type of expense via the web client |
| 19 | Pass | testModifyTeam | The user is able to successfully modify the team description, location, name, and whether it is managed via the web client |
| 20 | Pass | testResetPassword | The user is able to reset their password by providing their username. User receives an email with temporary password. |
| 21 | Pass | testGetTeamTasks | The user is able to see the tasks that are assigned to the selected team from the team page. |
| 22 | Pass | testEditTeamTasks | The user is able to edit the selected team task from the UI. |
| 23 | Pass | testMarkAsComplete | The user is able to mark a task as complete from the UI. |
| 24 | Pass | testDeleteTeamTask | The user is able to delete a team task from the UI. |
| 25 | Pass | testApproveUser | A manager is able to approve a request from the alert dropdown in the header of the client. |
| 26 | Pass | testRejectUser | A manger is able to reject a request from the alert dropdown in the header of the client. |
| 27 | Pass | testUploadReceipt | The user is able to upload an image as a receipt from the Expenses grid. |
| 28 | Pass | testViewReceipt | The user is able to view an uploaded image from the Expenses grid. |
| 29 | Pass | testRequestJoinTeam | A user can request to join a managed/private team from the JoinTeam popup. |
| 30 | Pass | testAddEvent | User is able to create a new event on the Agenda page. |
| 31 | Pass | testEditEvent | User is able to edit the selected event on the Agenda page. |

| 32 | Pass | testDeleteEvent | User is able to delete the selected event on the Agenda page. |
|----|------|-----------------|---------------------------------------------------------------|
| 33 | Pass | testGetDirections | Client produces the google map web page with the event address when the user clicks the "Get Directions" icon on the Agenda page. |

## Test Case Name: User Login - PASS

| Step Number | Step Description | Test Data | Expected Result |
|---|---|---|---|
| 1. | User access application | | Default page of application is shown |
| 2. | User enters authentication | User's credentials | Login form is completely filled out |
| 3. | User submits authentication | User's credentials | User accesses the system |

## Test Case Name: Create User - PASS

| Step Number | Step Description | Test Data | Expected Result |
|---|---|---|---|
| 1. | User access application | | Default page of application is shown |
| 2. | User navigates to registration | | Registration part of application is shown |
| 3. | User enters information | User's information | Registration form is completely filled out |
| 4. | User submits registration information | User's information | User is entered into the system |

## Test Case Name: Get User - PASS

| Step Number | Step Description | Test Data | Expected Result |
|---|---|---|---|
| 1. | User access application | | Default page of application is shown |
| 2. | User enters authentication | User's credentials | Login form is completely filled out |
| 3. | User submits authentication | User's credentials | User accesses the system |
| 4. | Client retrieves user's information | User's loginId | User data retrieved by client |

## Test Case Name: Create Managed Team - PASS

| Step Number | Step Description | Test Data | Expected Result |
|---|---|---|---|
| 1. | Manager accesses application | | Default page of application is shown |
| 2. | Manager navigates to team registration | | Team registration is shown |
| 3. | Manager enters team information | Team's information | Team registration form is completely filled out |
| 4. | Manager submits team information | Team's information | Team is successfully added to the system. |

## Test Case Name: Create Unmanaged Team - PASS

| Step Number | Step Description | Test Data | Expected Result |
| --- | --- | --- | --- |
| 1. | User logs into system | | Default page of application is shown |
| 2. | User navigates to team registration | | Team registration is shown |
| 3. | User enters team information. User specifically selects public team. | Team's information | Team registration form is completely filled out |
| 4. | User submits team information. | Team's information | Team is successfully added to the system. |

## Test Case Name: Handle Access Removal - PASS

| Step Number | Step Description | Test Data | Expected Result |
| --- | --- | --- | --- |
| 1. | User views team information | Select team | View information |
| 2. | Manager removes user | Remove user | User is not shown on team member list |
| 3. | User is rejected from access | Navigate to additional information within team information | User does not receive information, receives a no access notification and is returned to an empty screen |

## Test Case Name: Joining Public Team- PASS

| Step Number | Step Description | Test Data | Expected Result |
| --- | --- | --- | --- |
| 1. | User joins public team | User Name, Team Name | Request is filed, and team is added to user's teams |
| 2. | User views team information | User Name, Team Name | Team information is displayed to the user after they are authenticated and authorized access |

## Test Case Name: Joining Private Team- PASS

| Step Number | Step Description | Test Data | Expected Result |
| --- | --- | --- | --- |
| 1. | User joins private team | User Name, Team Name | Request is filed, but membership is not granted until a team manager approves membership. |
| 2a. | Manager approves membership | Manager User Name, Team Name, Other User Name | Manager approves membership and team member is notified their membership is granted. |
| 2a.1 | User receives notification of team approval | User Name, Team Name | Information is displayed to the user that their membership is granted. |

| 2b. | Manager rejects membership | Manager User Name, Team Name, Other User Name | Manager approves membership and team member is notified their membership is denied. |
| 2b.1 | User receives notification of team member rejection | User Name, Team Name | Information is displayed to the user that their membership was denied. |

## Test Case Name: Expense Retrieval - PASS

| Step Number | Step Description | Test Data | Expected Result |
|---|---|---|---|
| 1. | User views expenses | User Name, Team Name | User is presented data only related to himself for the current team he is viewing. |
| 2. | User edits expenses | User Name, Team Name, Expense ID | User updates expense data tied to current expense for himself in the current group he is viewing. |