SpiraTeam® | Quick Start Guide

Inflectra Corporation

Date: January 28, 2018

Contents

Introduction

The Spira™ family of applications from Inflectra® are a powerful set of tools that help you manage your software lifecycle.

SpiraTeam® is our integrated Application Lifecycle Management (ALM) system that manages your project's requirements, releases, test cases, issues and tasks in one unified environment.

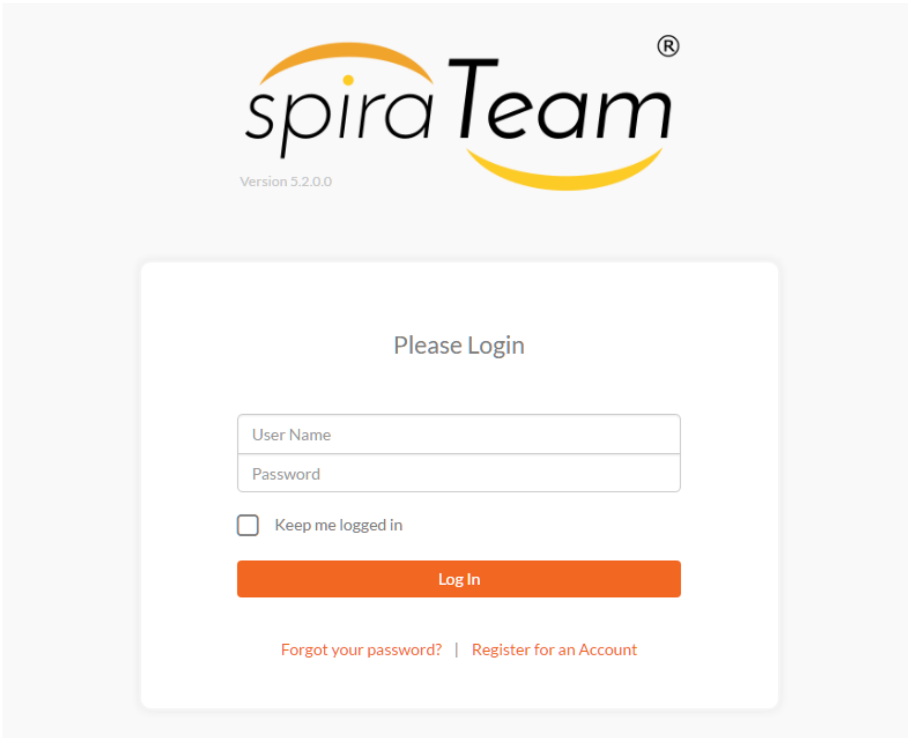
This guide provides a quick step-by-step tutorial for creating a sample SpiraTeam project, adding some requirements and releases, building the test plan, assigning the backlog items to the various sprints, executing the tests and logging defects, and finally triaging and resolving the bugs/issues raised.

For further information on using SpiraTeam, please refer to the more comprehensive *SpiraTeam User Manual*.

For information on setting up a new SpiraTeam instance, creating projects and users and other administration tasks, please refer to the *SpiraTeam Administration Guide*.

1. Logging In and Selecting a Project

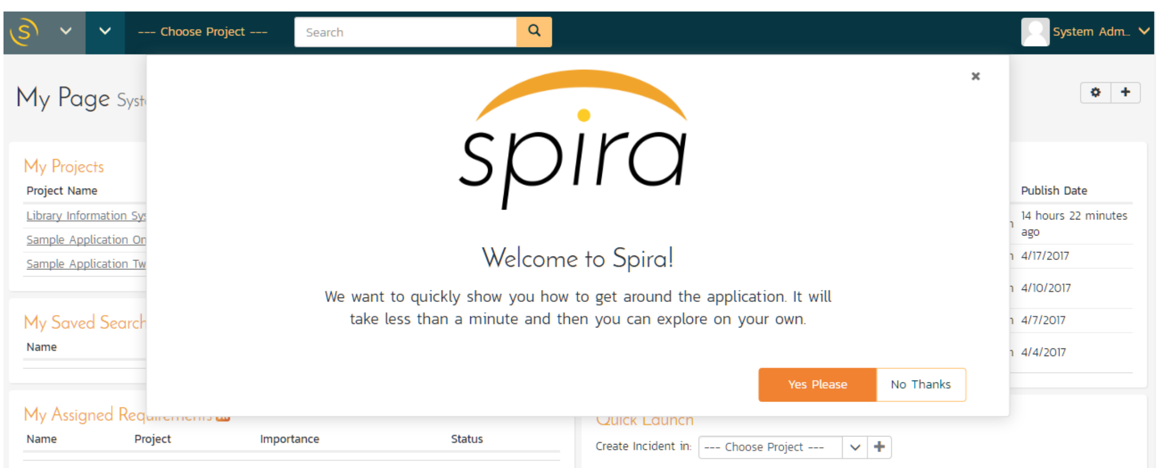
Once you have either downloaded and installed a trial version of SpiraTeam or signed up for a hosted trial, you should be able to see the following login screen in your web browser:



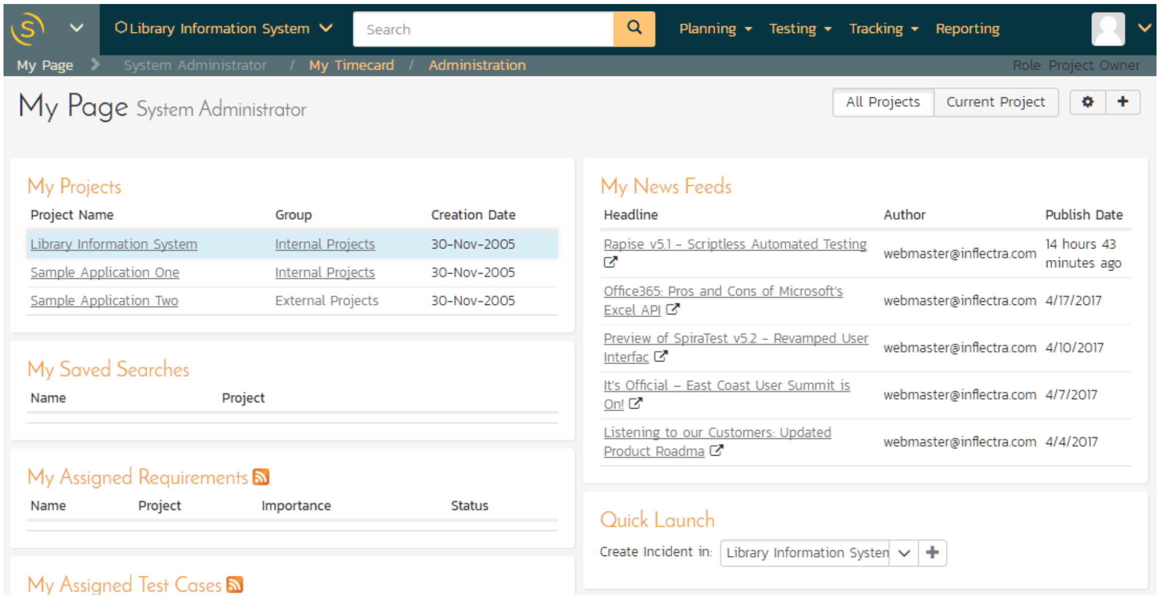
Enter the following default login and password to start using the system:

- Login: administrator
- Password: PleaseChange

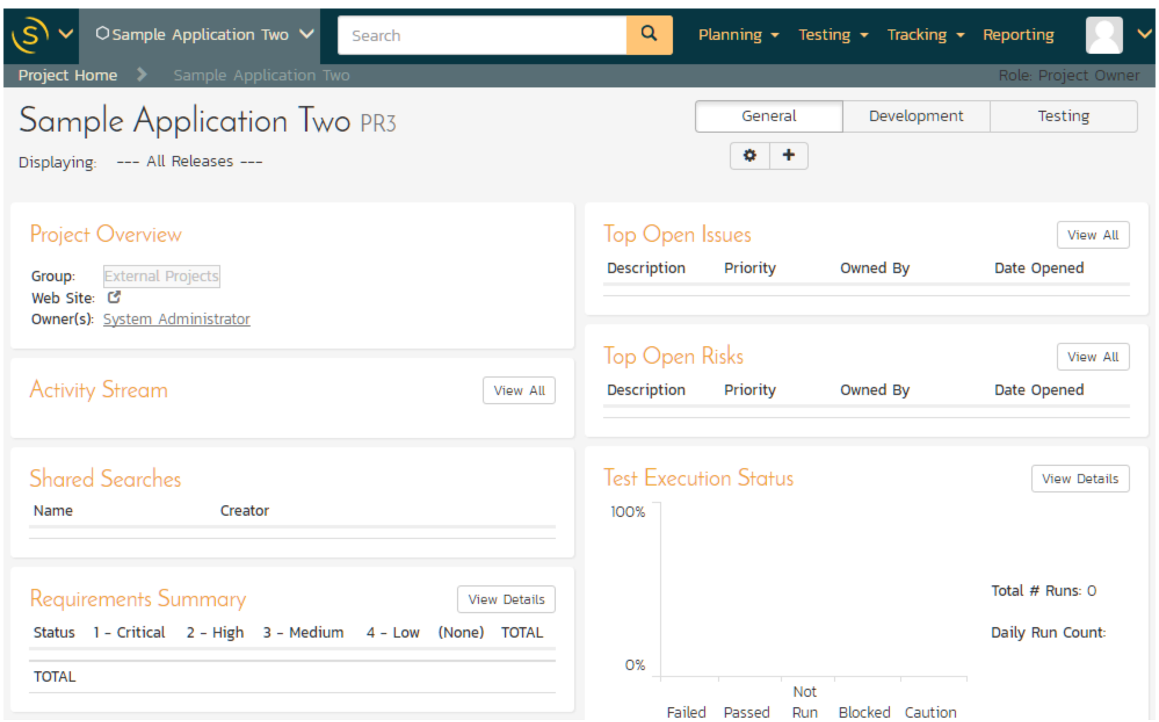
Once you have logged-in, you should be taken to the “My Page” for your user. Please note, that the very first time you log in you will be asked if you want to take a quick orientation tour of the application (which will look similar to the screenshot below).



Initially this page will be fairly empty, with only the list of sample projects populated. This is normal.



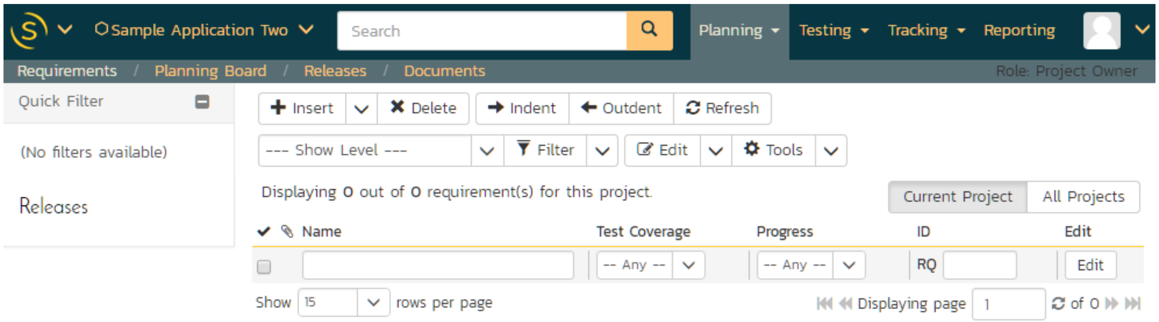
For this tutorial we want to start out with an empty project that has no data in it, so click the hyperlink under "My Projects" for "Sample Application Two". That will bring up the homepage for the empty project:



The project home page will display various widgets containing key project metrics; however they will be initially empty. The next step will be to start entering your project's data.

2. Define the Requirements

Click on the Planning > Requirements menu option to display the project's requirements list page:



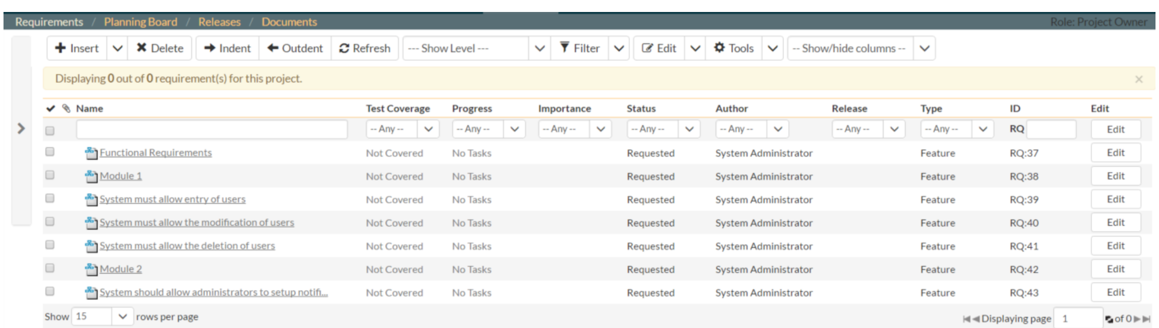
The terminology in SpiraTeam is designed to be methodology agnostic, however the following table illustrates the nomenclature used in SpiraTeam and how it relates to some common agile methodologies:

SpiraTeam	**Extreme Programming**	**Scrum**	**AgileUP / DSDM**
Summary Requirement	Epic	Epic	Feature Group
Requirement	User Story	Backlog Item	Requirement
Task	Task	Task	Task
Release	Release	Release	Release
Iteration	Iteration	Sprint	Iteration

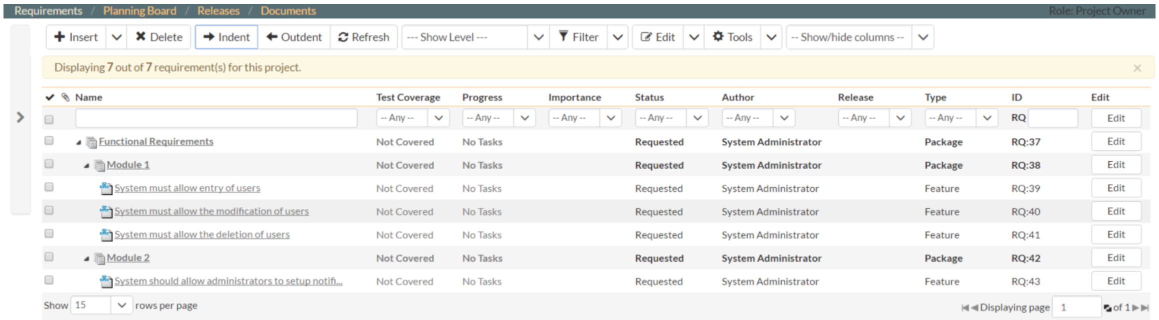
Initially the requirements list will be empty. Click on the “Insert” button in the main toolbar to create the following requirements in turn:

- Functional Requirements
- Module 1
- System must allow entry of users
- System must allow the modification of users
- System must allow the deletion of users
- Module 2
- System should allow administrators to setup notifications

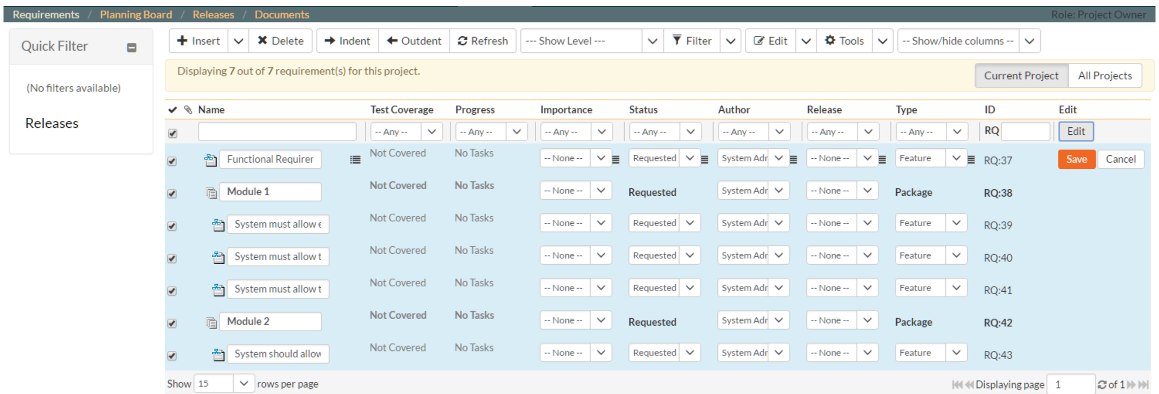
You should now have a simple flat requirements list, as illustrated below:



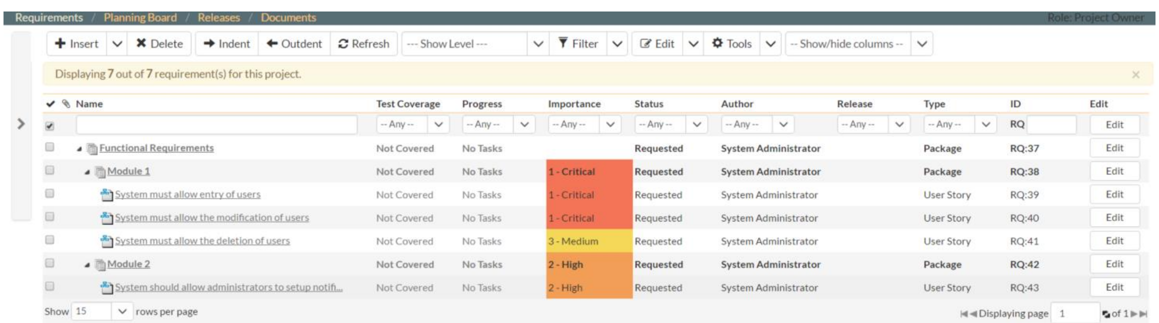
Now to correctly indent the requirements, select the checkboxes of all the requirements under “Functional Requirements” and click ‘Indent’. Then select the three requirements under “Module 1” and click ‘Indent’ again. Finally, right-click on the last requirement and choose ‘Indent’ from the popup context menu. You should now have a list that looks like:



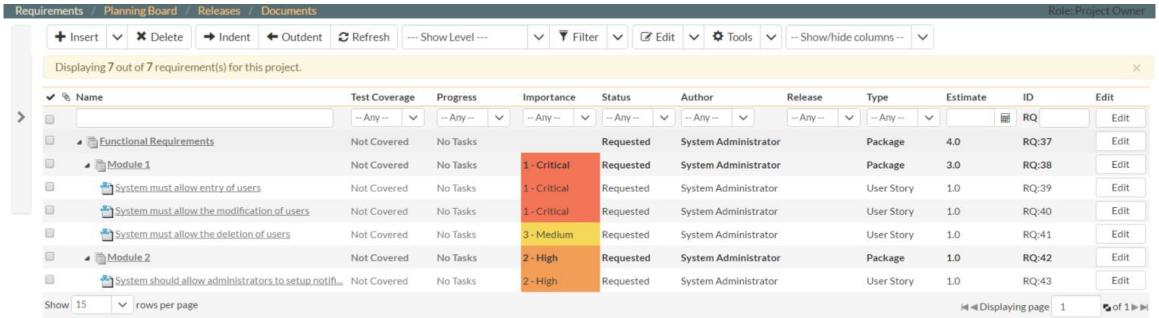
We finally want to set a type and priority value for the different requirements, to do that we can simply click the "select all" checkbox at the top of the list and then click on the top 'Edit' button in the right-hand column of that same row. That will make all the requirement rows editable:



Now, set the type of all your requirements to 'User Story' and choose some different values for the 'Importance' field and click Update. Now you will have a prioritized list of user story requirements:



Now that we have prioritized the requirements, the developers or business analysts responsible for estimating the complexity of each requirement need to enter estimates. To do that we first need to display the estimate column in the grid. To do that, click on the "Show/Hide Columns" dropdown list and show the Estimate column. This will display the requirements' estimates in the grid:

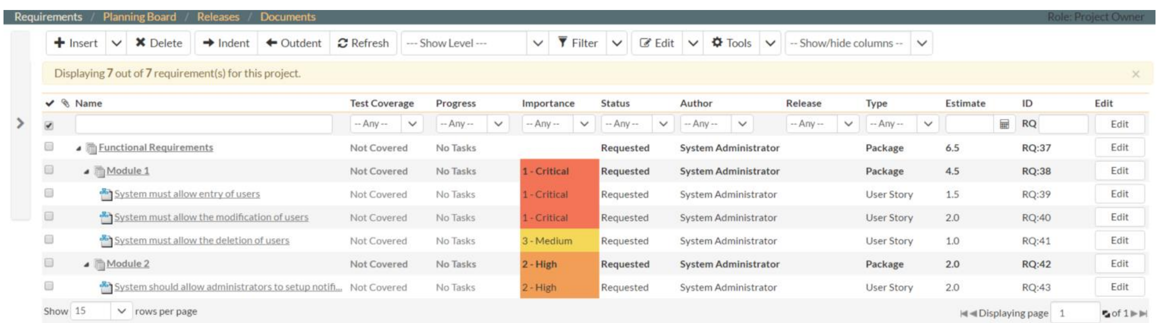


By default, all the requirements will have been assigned a default estimate of 1.0 points. The points are not a defined number of hours, but are instead an indication of the size of the requirement. The estimates should reflect how big each of the requirements are relative to each other.

Now, we can simply click the "select all" checkbox at the top of the list and then click on the top 'Edit' button in the right-hand column. The requirements will be displayed in editable mode again. Enter the following initial estimates for the requirements:

Requirement	Estimate
System must allow entry of users	1.5 points
System must allow the modification of users	2.0 points
System must allow the deletion of users	1.0 points
System should allow administrators to setup notifications	2.0 points

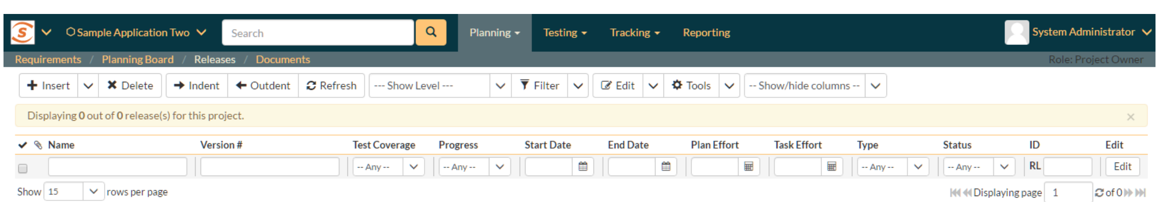
You should now have the requirements' estimates displayed in the system:



Now that we have a list of prioritized, estimated requirements, the next task is to enter in our planned list of releases and iterations for the project.

3. Create the Release and Iteration Plan

Click on the Planning > Releases menu option to display the project's release list page:



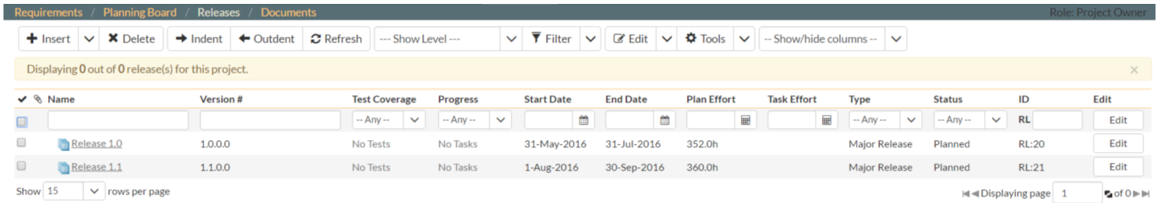
Initially the release list will be empty. Click on the "Insert" button in the main toolbar to enter the following releases in turn:

- Release 1.0 – version number 1.0.0.0

- Start Date: Today's Date
- End Date: Today's Date + 2 months

- Release 1.1 – version number 1.1.0.0
  - Start Date: Today's Date + 2 months
  - End Date: Today's Date + 4 months

You should now have a list of releases that looks like the following:

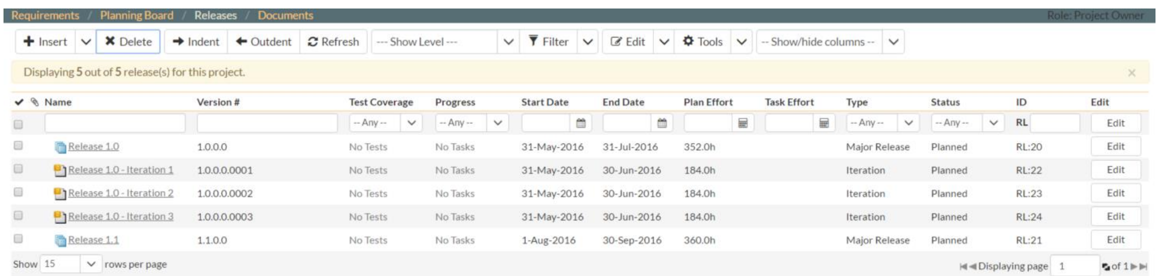

Requirements / Planning Board / Releases / Documents

Displaying 0 out of 0 release(s) for this project.

Name	Version #	Test Coverage	Progress	Start Date	End Date	Plan Effort	Task Effort	Type	Status	ID	Edit
Release 1.0	1.0.0.0	No Tests	No Tasks	31-May-2016	31-Jul-2016	352.0h		Major Release	Planned	RL:20	Edit
Release 1.1	1.1.0.0	No Tests	No Tasks	1-Aug-2016	30-Sep-2016	360.0h		Major Release	Planned	RL:21	Edit

Now that we have the releases defined, we need to add one additional level of detail to each release – the list of *iterations* or *sprints* that will take place in each release. We shall enter in some sample iterations for the first release only.

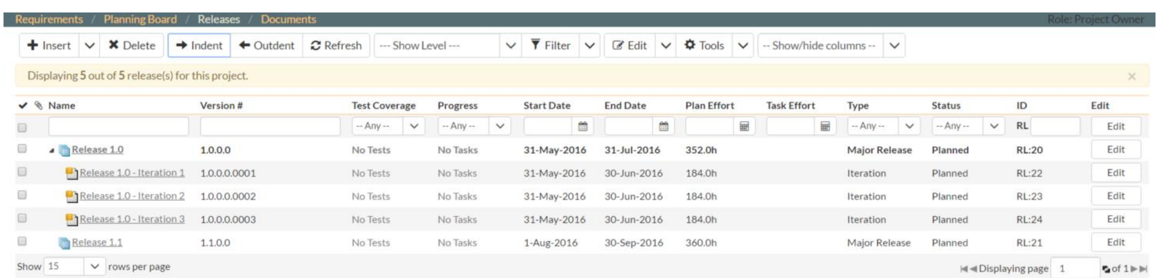
Select the checkbox for the second release (Release 1.1) and choose the menu option Insert > New Iteration. A new iteration will be inserted between Release 1.0 and Release 1.1. Choose a name for the iteration, specify the date-range for the iteration, and click 'Update'. Now repeat this process twice and you should now have three iterations added to the list. We recommend making each iteration last 2-weeks and have each one scheduled in series:


Requirements / Planning Board / Releases / Documents

Displaying 5 out of 5 release(s) for this project.

Name	Version #	Test Coverage	Progress	Start Date	End Date	Plan Effort	Task Effort	Type	Status	ID	Edit
Release 1.0	1.0.0.0	No Tests	No Tasks	31-May-2016	31-Jul-2016	352.0h		Major Release	Planned	RL:20	Edit
Release 1.0 - Iteration 1	1.0.0.0.0001	No Tests	No Tasks	31-May-2016	30-Jun-2016	184.0h		Iteration	Planned	RL:22	Edit
Release 1.0 - Iteration 2	1.0.0.0.0002	No Tests	No Tasks	31-May-2016	30-Jun-2016	184.0h		Iteration	Planned	RL:23	Edit
Release 1.0 - Iteration 3	1.0.0.0.0003	No Tests	No Tasks	31-May-2016	30-Jun-2016	184.0h		Iteration	Planned	RL:24	Edit
Release 1.1	1.1.0.0	No Tests	No Tasks	1-Aug-2016	30-Sep-2016	360.0h		Major Release	Planned	RL:21	Edit

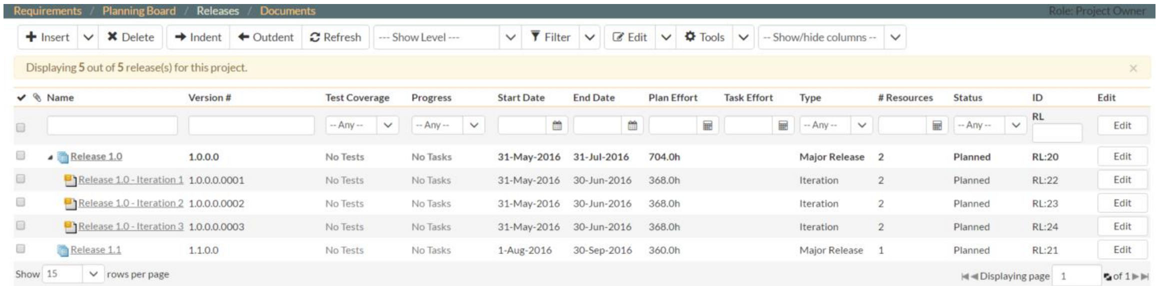
Finally, to indent the three iterations under the parent release, select the checkboxes of the three iterations and click the 'Indent' button. You should now have a release hierarchy that looks like:


Requirements / Planning Board / Releases / Documents

Displaying 5 out of 5 release(s) for this project.

Name	Version #	Test Coverage	Progress	Start Date	End Date	Plan Effort	Task Effort	Type	Status	ID	Edit
Release 1.0	1.0.0.0	No Tests	No Tasks	31-May-2016	31-Jul-2016	352.0h		Major Release	Planned	RL:20	Edit
Release 1.0 - Iteration 1	1.0.0.0.0001	No Tests	No Tasks	31-May-2016	30-Jun-2016	184.0h		Iteration	Planned	RL:22	Edit
Release 1.0 - Iteration 2	1.0.0.0.0002	No Tests	No Tasks	31-May-2016	30-Jun-2016	184.0h		Iteration	Planned	RL:23	Edit
Release 1.0 - Iteration 3	1.0.0.0.0003	No Tests	No Tasks	31-May-2016	30-Jun-2016	184.0h		Iteration	Planned	RL:24	Edit
Release 1.1	1.1.0.0	No Tests	No Tasks	1-Aug-2016	30-Sep-2016	360.0h		Major Release	Planned	RL:21	Edit

Finally, we need to specify the number of resources that will be assigned to each iteration and release. To do that, click on the "Show/Hide Columns" dropdown list and Show the "# Resources" column. To make room you can hide the "Iteration?" column. Once the column is visible, select the checkboxes for "Release 1.0" and click the Bulk Edit button again. Now you should be able to adjust the # resources for the release and its iterations to 2. Then click Update:

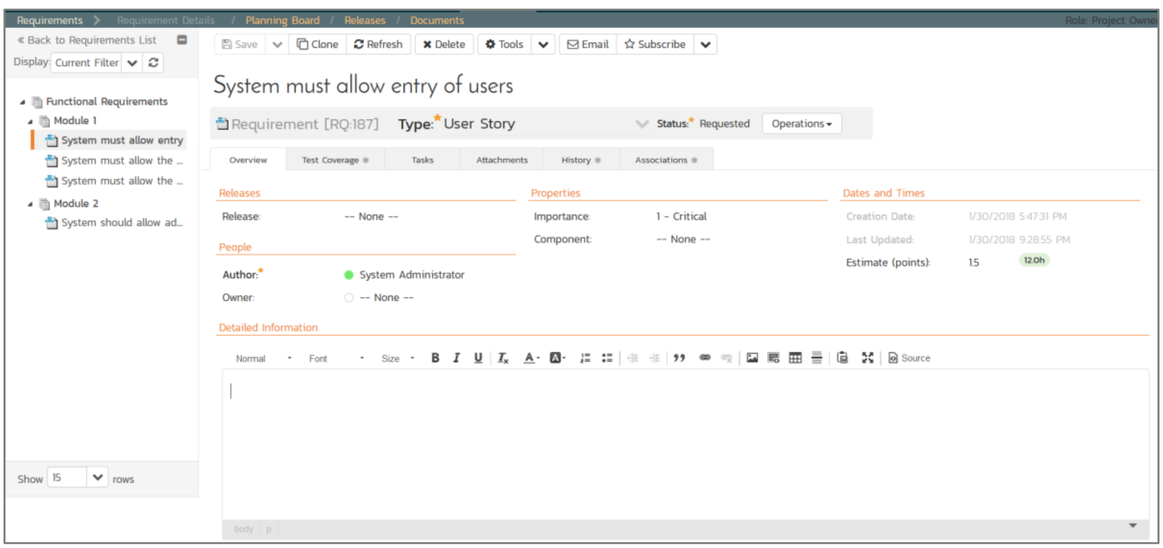


The screen now shows that the Plan Effort for each of the iterations has been doubled. This is the amount of time that you have available to allocate.

4. Adding Requirement Tasks

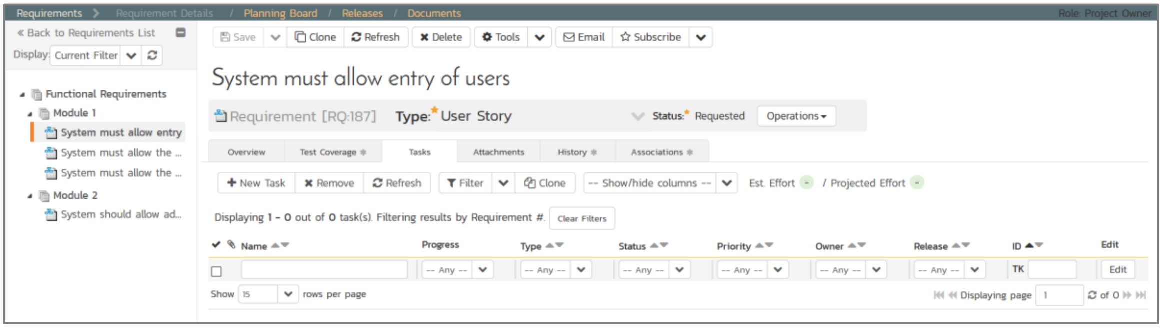
Now that we have defined the macro-schedule for Release 1.0, the next stage is to have the developers take each of the requirements defined so far and define the various tasks that will need to be carried out to develop the functionality outlined in the requirement. Each task will have its own estimate associated with it. In addition, you can optionally specify date-ranges and priorities to each of the individual tasks.

To start adding tasks, click on Planning > Requirements to display the requirements list. Now, click on the hyperlink for the first requirement ("System must allow entry of users") and the requirements details page will be displayed:



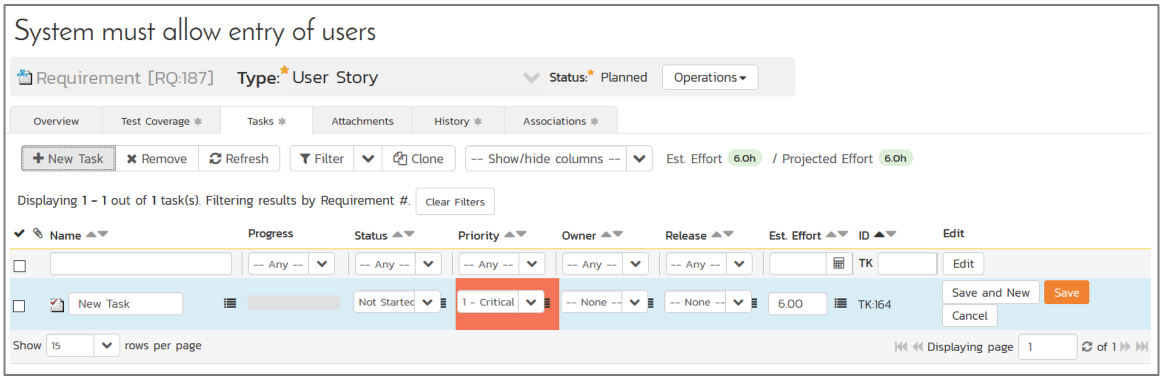
Notice that the system displays an initial resource estimate of 12.00 hours for this requirement. This is simply based on an initial project setting of 8 hours per story point. Once you start adding tasks and getting metrics based on the actual team velocity (how many story points they can accomplish in a given time frame), the system can update that conversion metric.

Now click on the "Tasks" tab to display the list of tasks defined for this requirement:

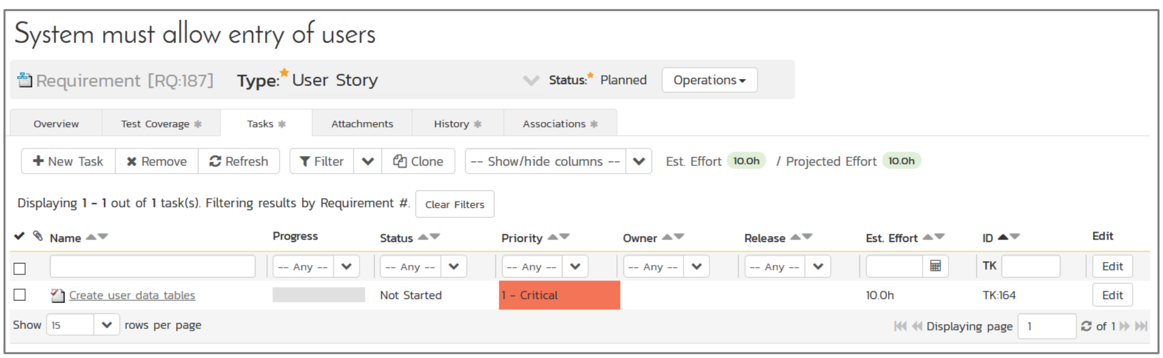


Initially the list of tasks will be empty. Before entering the tasks, first click on the "Show/Hide Columns" dropdown list and show the "Show Est. Effort" column. This will display the estimated effort for each Task.

Now, click on the "New Task" button to add a new task associated with this requirement:



Enter the name of the task "Create user data tables" and the estimated effort 10.0h and click Update. The new task should now have been added:

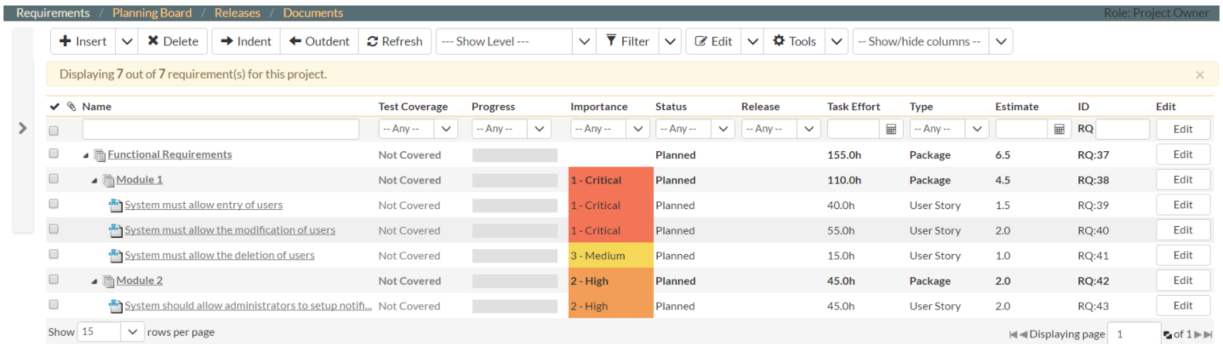


Now we need to add the remaining tasks; the following table outlines all the tasks that need to be added under each of the requirements (we have included the one already created for completeness):

Requirement / Task	Est. Estimate
System must allow entry of users	
Create user data tables	10.0h
Develop user business object	10.0h
Build user creation screens	20.0h
System must allow the modification of users	
Extend user business object to handle updates	5.0h
Add user list page	15.0h
Add user details page	20.0h

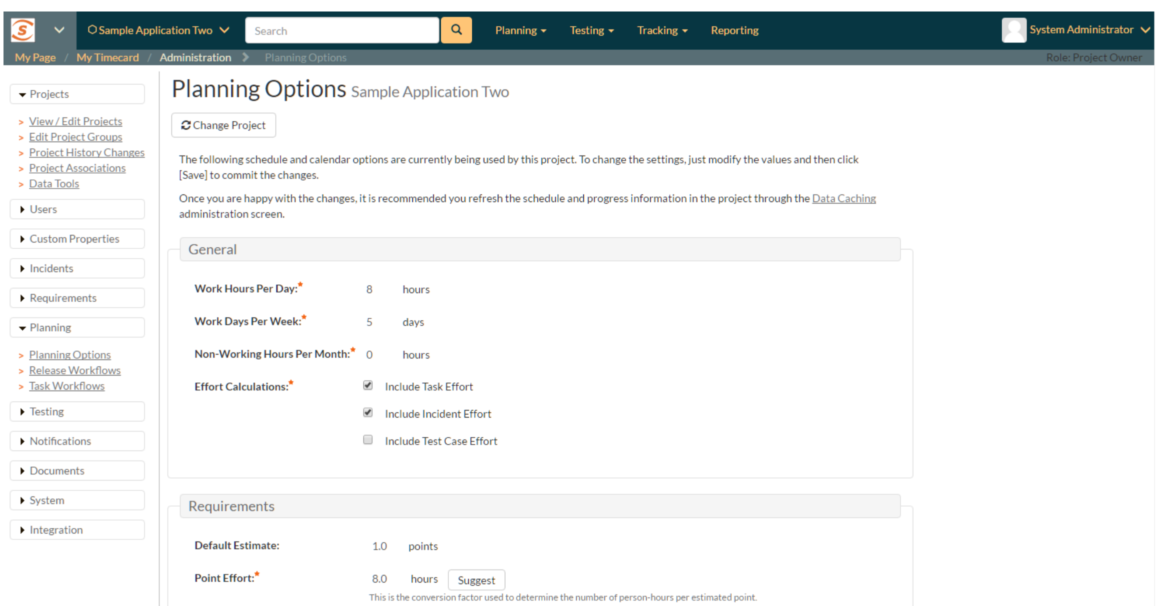
Add user permissions page	15.0h
System must allow the deletion of users	
Extend user business object to handle deletes	5.0h
Update user list page to add delete functionality	10.0h
System should allow administrators to setup notifications	
Create user administration home page	15.0h
Add user settings for notifications to database	10.0h
Create user notifications administration page	20.0h

You should now have the following requirements list page:

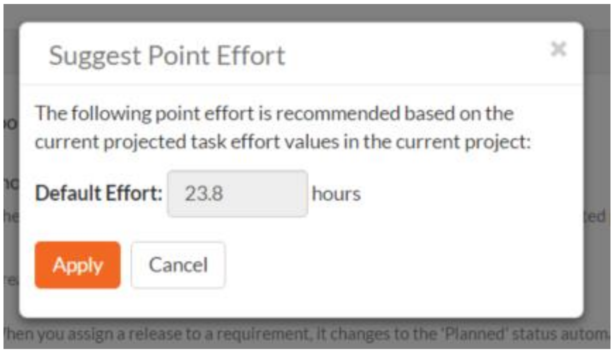


Note that we've hidden the "Author" field and shown the "Task Effort" field to show the detailed task effort aggregated up to the parent requirements.

Now when we add new requirements, it would be nice if the system would use this updated information when determining the estimated number of hours to deliver requirements with a specified number of story points. To update the metric the system uses, go to the dropdown menu to the right of the "S" logo and select Administration > Planning> Planning Options:



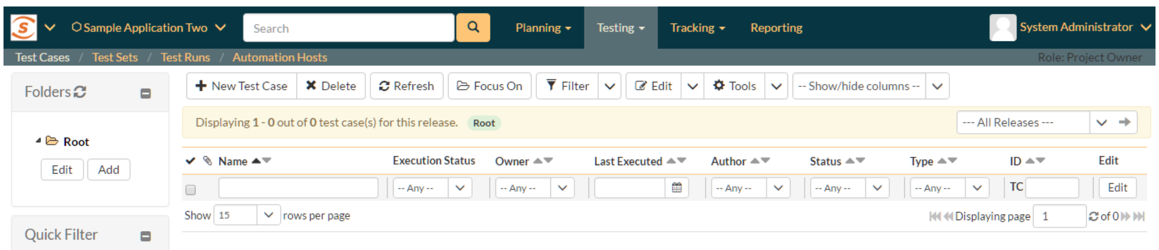
As you can see, the system lists 8.0 hours as the current number of hours required to deliver a single story point of functionality. Now that we have some actual tasks in the project, click on the "Suggest" button to have the system provide its suggestion of the new metric:



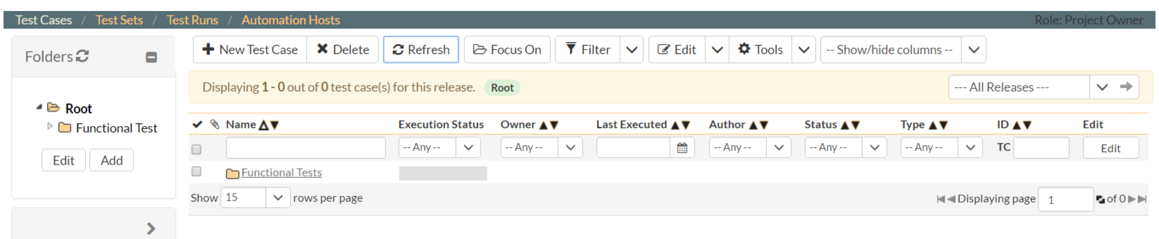
Click the ‘Apply’ button to update the planning metric, and then click the main ‘Save button to confirm the change.

5. Adding the Test Cases

Click on the Testing > Test Cases menu option to display the project’s test case list page:



Initially the test case list will be empty and the only folder visible in the folder tree on the left hand side will be “root”. Click on the “Add” button underneath the folder tree on the left, enter the new folder name “Functional Tests” and click the ‘Save button. You now should have a new folder in the tree view, and after clicking the “Refresh” button above the list on the right, you will see a single folder:



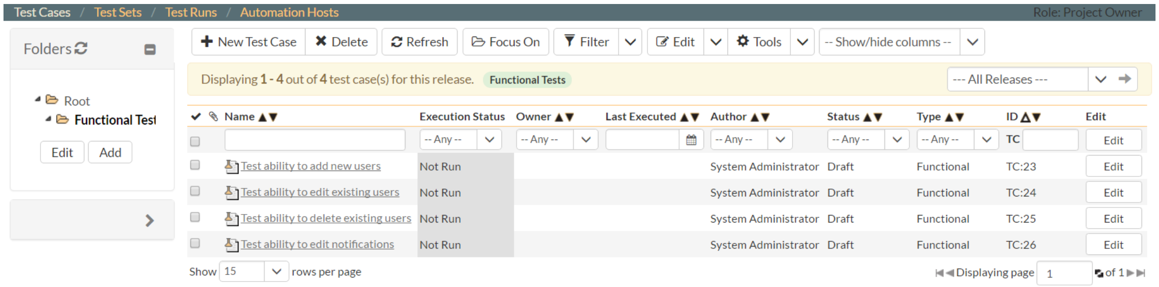
Now click on this folder from the folder tree on the left, and click “**New Test Case**” from the main toolbar. Enter the name of the new test case:

- Test ability to add new users

and click “Save”. Now repeat this for the following other test cases:

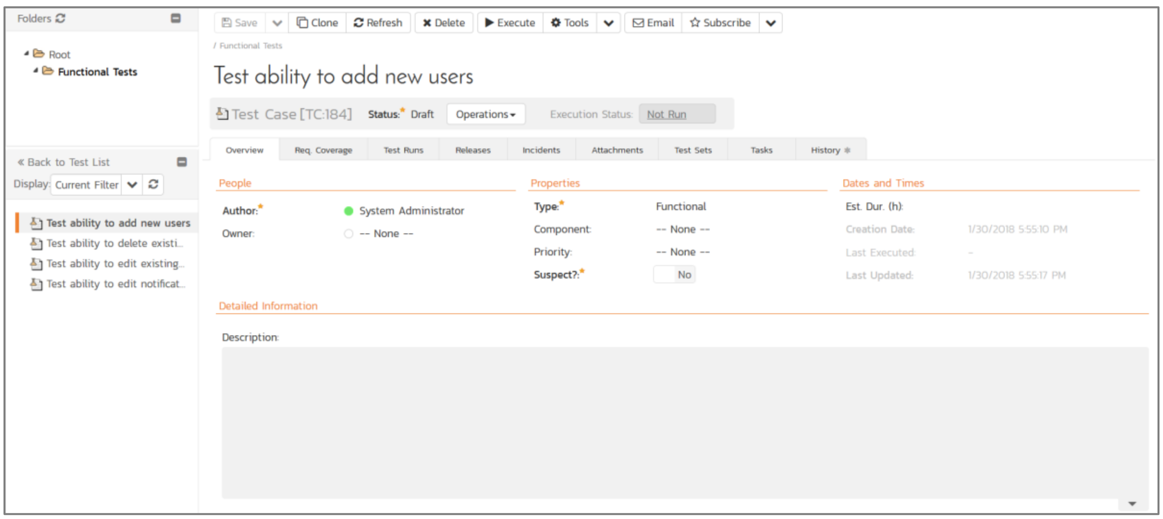
- Test ability to edit existing users
- Test ability to delete existing users
- Test ability to edit notifications

You should now have the following test case list:

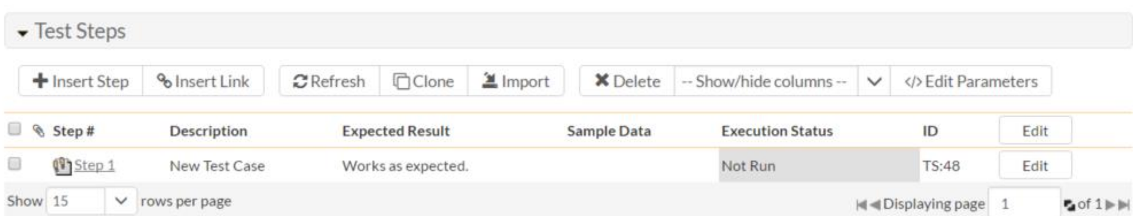


The next step will be to enter the detailed test steps and assign the test cases to the appropriate requirements that we previously entered.

Click on the hyperlink for the first test case “Test ability to add new users”. This will bring up the test case details page:



In the Description box, enter a brief overview of the test case (e.g. “this test case verifies that you can add new users to the system and that all of the fields get saved correctly.”). In the “Test Steps” section, you will see a single test step is already automatically created for you with some suggested text:

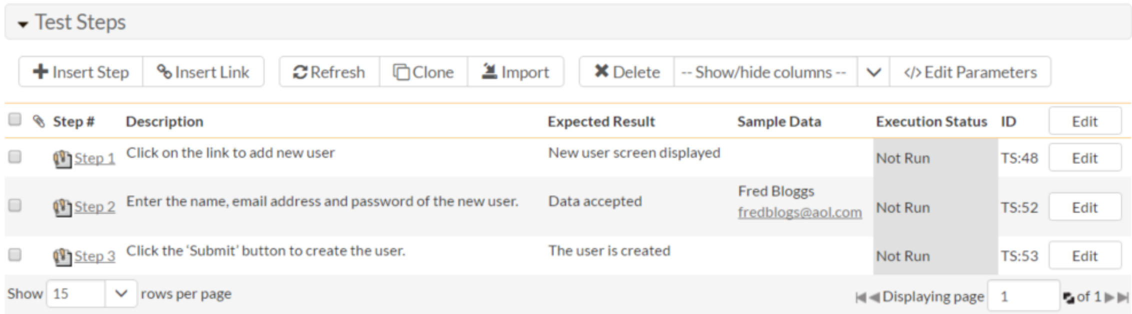


Enter the following three test steps (click on the ‘Insert Step’ link twice to add two more new test steps):

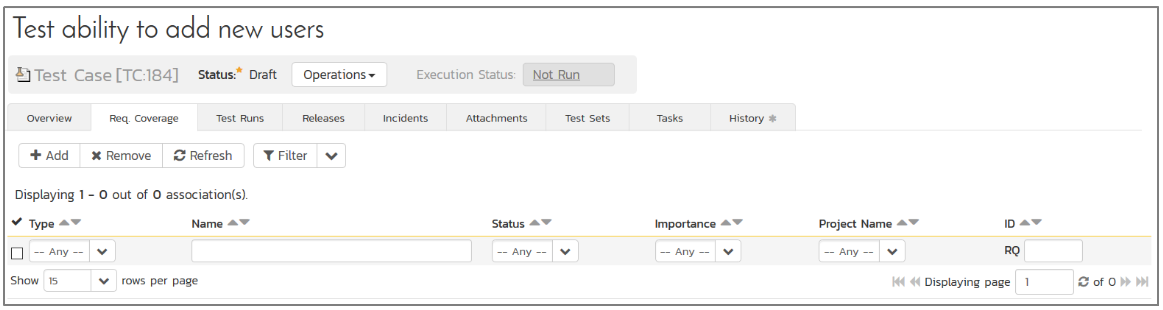
Test Step Description	Expected Result	Sample Data
Click on the link to add new user	New user screen displayed	
Enter the name, email address and password of the new user.	Data accepted	Fred Bloggs fredblogs@aol.com
Click the ‘Submit’ button to create the user.	The user is created	

After the first two, click the ‘Insert’ button to commit the change and add another row. After the third and final test step, click the ‘Save button to commit the change.

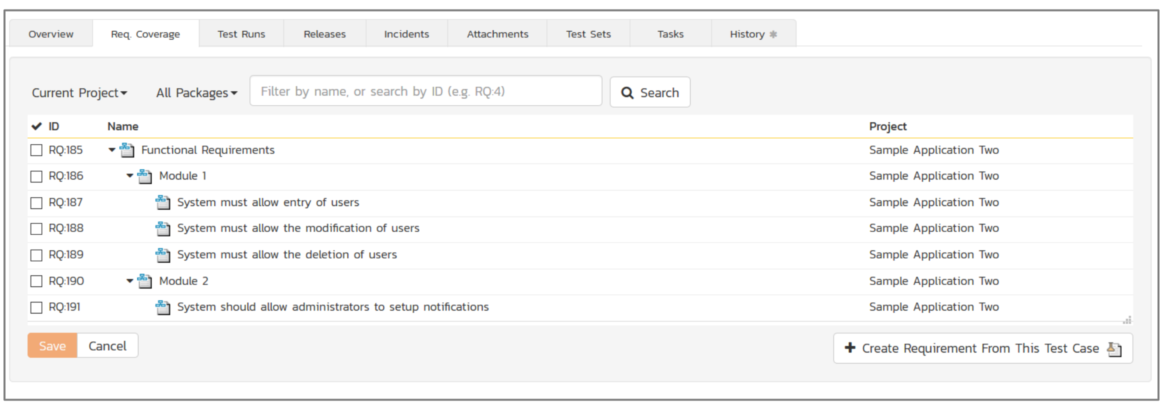
You should now have the following test steps in the test case:



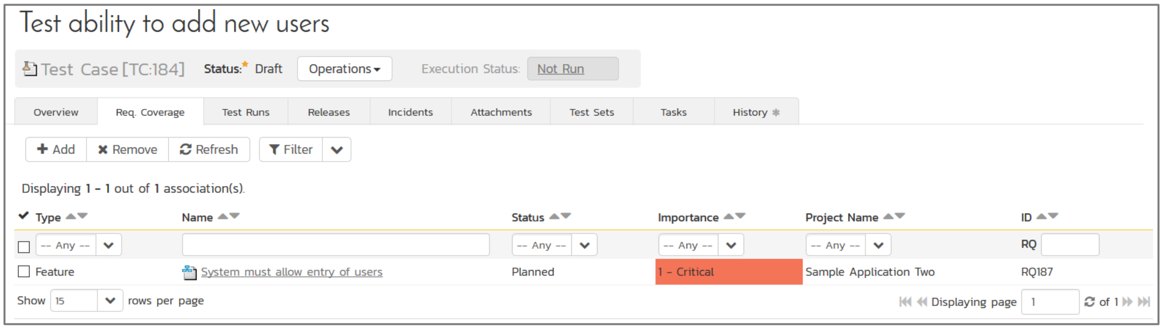
Now we need to link this test case to the requirement(s) that it validates. To do this, click on the “Req. Coverage” tab:



Click on the ‘+ Add’ button to display the association adding panel:



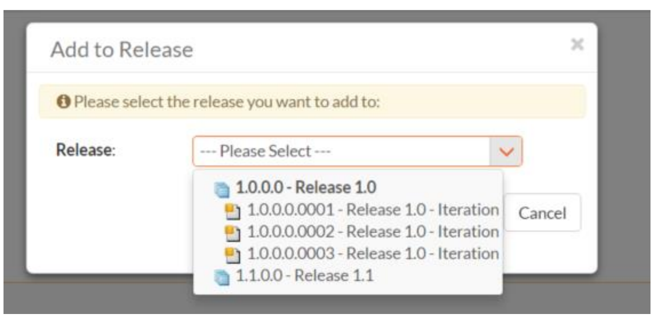
Choose the “System must allow the entry of users” requirement. Now click on the ‘Save’ button to add the test case to this requirement:



Now repeat the process for the other test cases, adding a couple of test steps for each one and linking the test cases to the requirements according to this table:

Test Case	Requirement
Test ability to add new users	System must allow entry of users
Test ability to edit existing users	System must allow the modification of users
Test ability to delete existing users	System must allow the deletion of users
Test ability to edit notifications	System should allow administrators to setup notifications

Now that we have our test cases entered and test coverage established, we need to specify which releases and iterations they can be tested in. To do that, first navigate to the project's test case list page again. Next, you need to select all of the contents of the "Functional Tests" folder. To do so you can put a checkbox in the box to the left of the folder name. Alternatively, you can click on the folder to show its contents. To quickly select all the test cases shown in the folder, you can check the checkbox in the header row of the table. Once the right selection has been made, click on Tools > Add to Release:



Now select "Release 1.0" and click "Add". You will now have added all the tests to the overarching Release. To add the tests to the different iterations, simply select the checkboxes of the different tests and then choose the Tools > Add to Release and add the individual tests to the appropriate iterations:

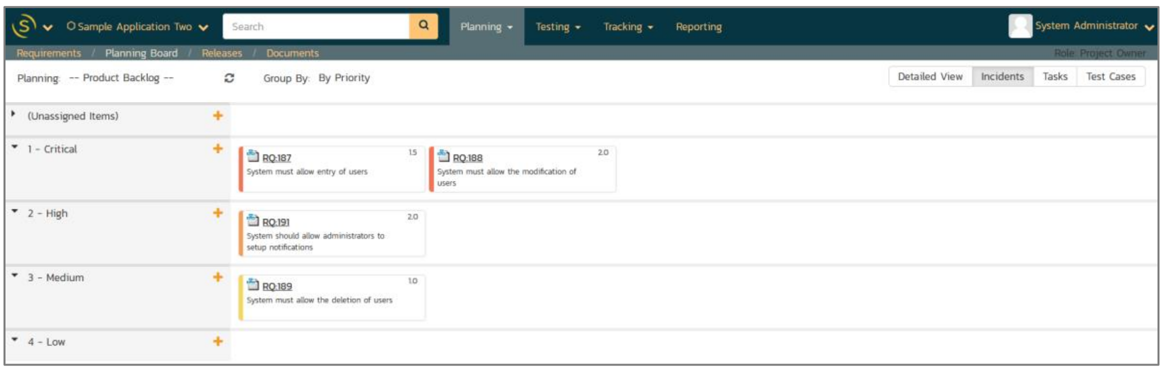
Test Case	Iteration(s)
Test ability to add new users	Release 1.0 - Iteration 1 Release 1.0 - Iteration 2 Release 1.0 - Iteration 3
Test ability to edit existing users	Release 1.0 - Iteration 1 Release 1.0 - Iteration 2 Release 1.0 - Iteration 3

Test ability to delete existing users	Release 1.0 - Iteration 2 Release 1.0 - Iteration 3
Test ability to edit notifications	Release 1.0 - Iteration 3

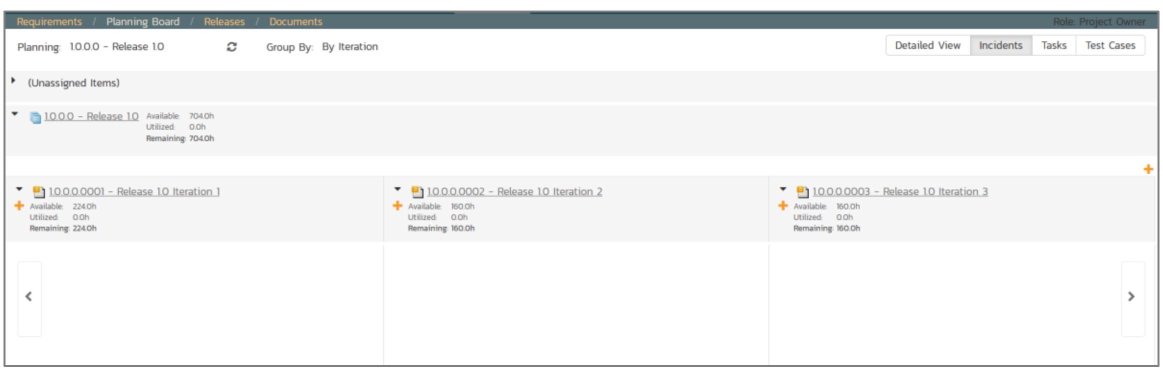
As illustrated, you typically want to include previous functionality in each of the successive iterations to ensure regression coverage. So, each iteration includes all the new test cases for the additional requirements, plus the existing test cases for the pre-existing functionality.

6. Planning the Iteration

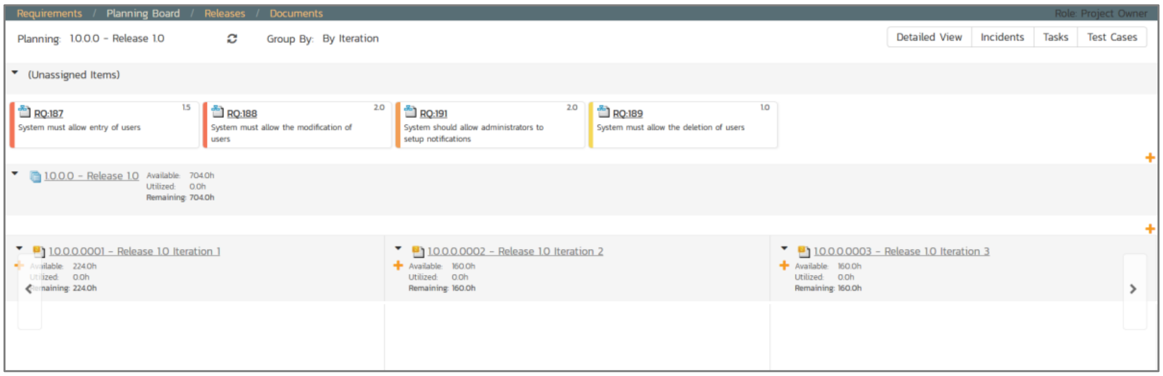
Now that the developers have elaborated each of the requirements to include the necessary tasks, the next step is to scope out the plan for each iteration / sprint. Click on the Planning > Planning Board option on the main menu to display the product backlog planning board. Make sure the “Group By” dropdown on the left is set to “By Priority”:



Now to view the iteration plan for a specific release, select ‘Release 1.0’ and ‘Group by Priority’ from the drop-down menus at the top-left of the screen. That will display the iteration plan for the selected release:

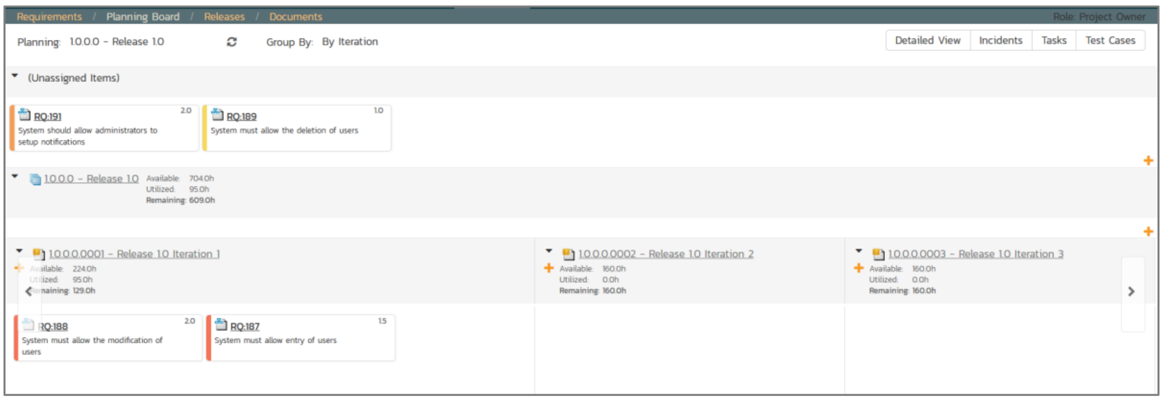


Expand the ‘(Unassigned Items)’ entry to display the requirements that are in the product backlog:

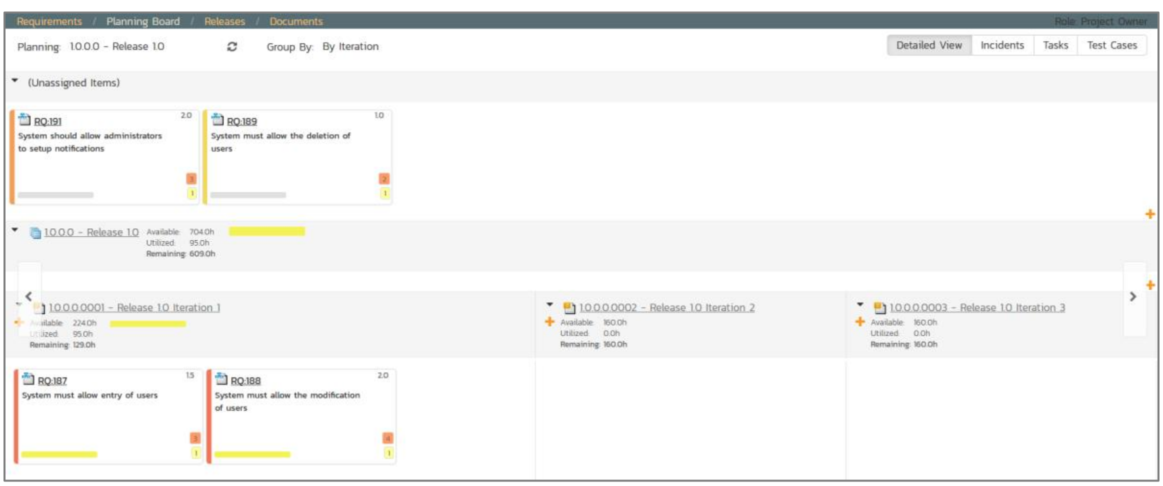


Each backlog item (requirement or incident) is represented by a virtual "story card" in the iteration. The left-hand side of each item displays the priority color and the progress bar at the bottom of each item depicts the progress of the item. You can flip through each iteration to see the work planned by clicking the left/right arrow buttons at the sides of the screen.

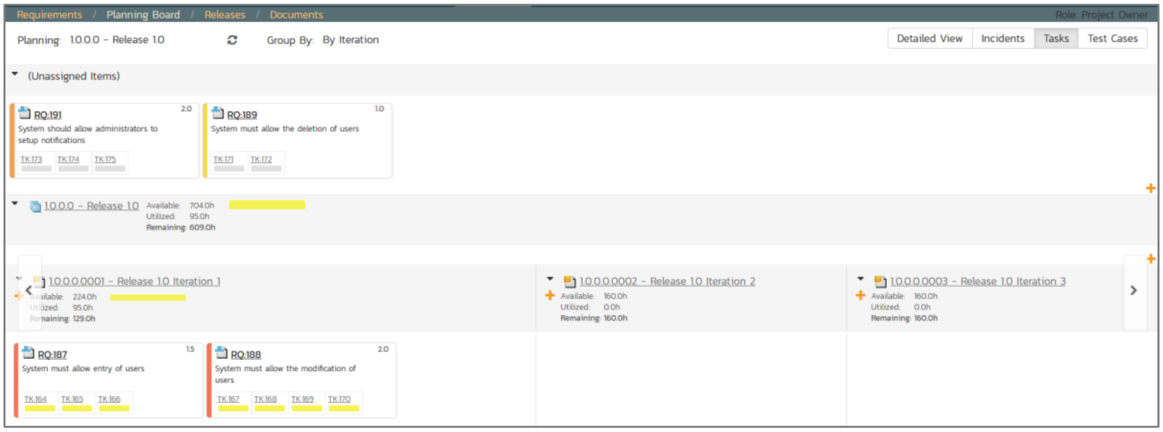
Now drag the two highest priority requirements (the ones with Importance = 1-Critical) to the first iteration:



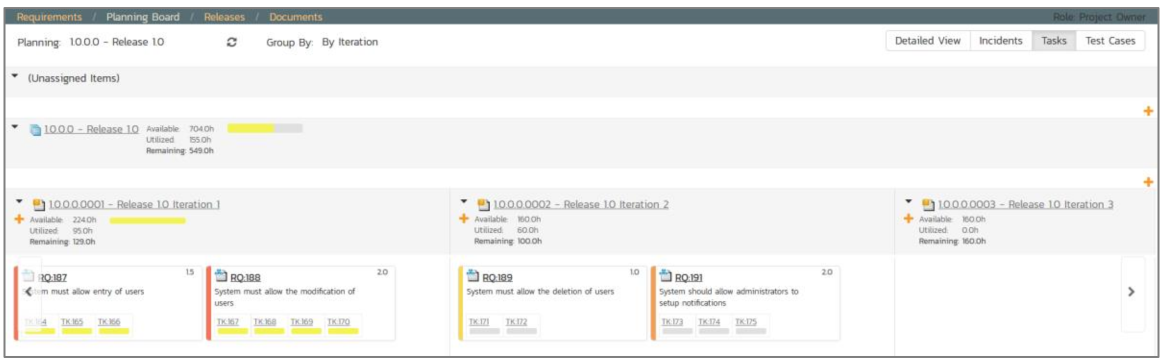
If you would like to see more information about each requirement, simply enable the 'Detailed View' option:



To see the individual tasks associated with each requirement, you can also select the 'Tasks' option:



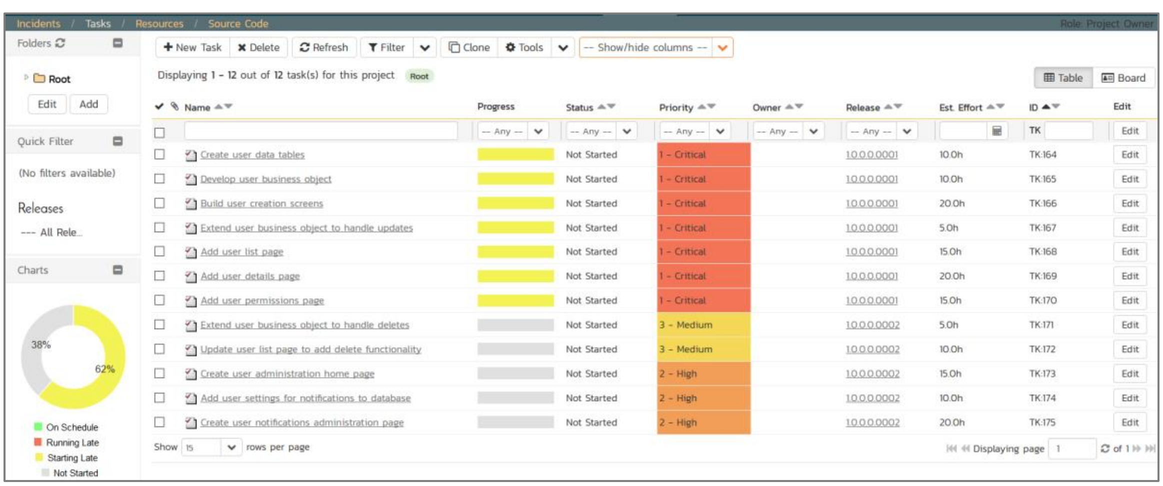
You can now determine how much time has been scheduled in the first iteration and how much time is remaining. Although we do have spare time available in Iteration 1 we will leave room left for fixing incidents, so next, drag and drop the remaining two requirements to the second iteration:



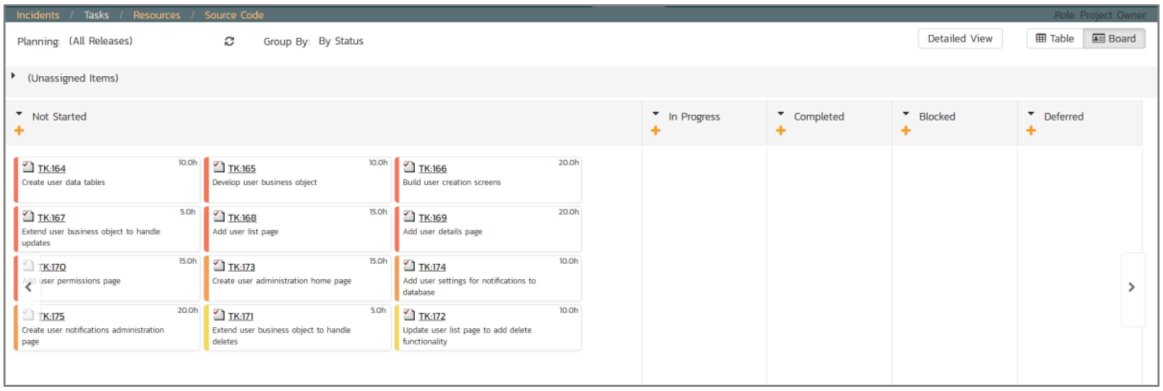
7. Assigning the Requirements & Tasks

Now that we have planned which requirements (user stories) and tasks are planned for each iteration, we can assign the individual tasks to the appropriate developer(s). The process you follow will depend on your methodology (e.g. in Scrum the developers simply pick the tasks, whereas in Extreme Programming the project manager will usually do the assignment).

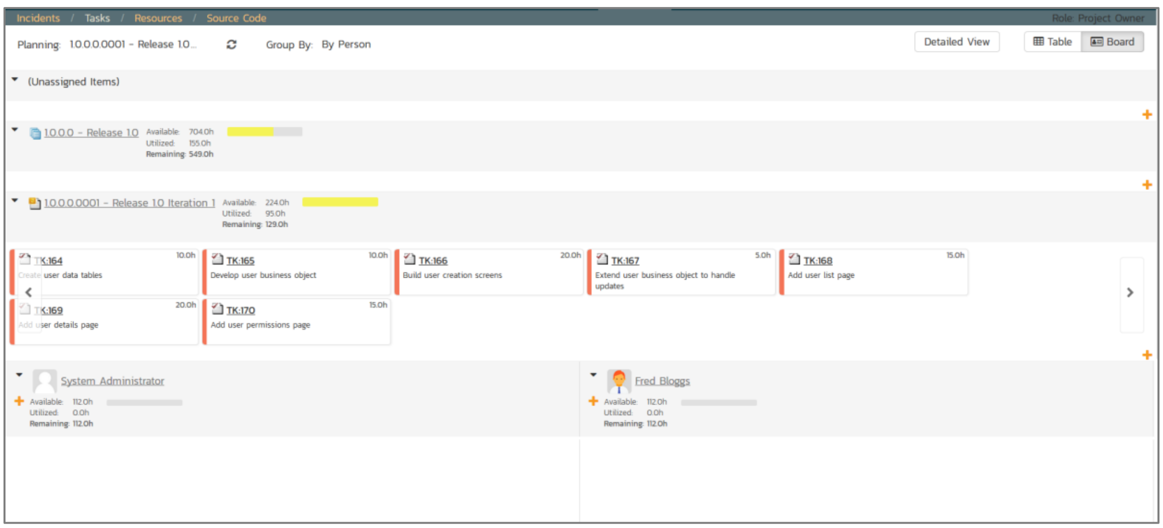
To assign the tasks, click on Tracking > Tasks to display the main tasks list page:



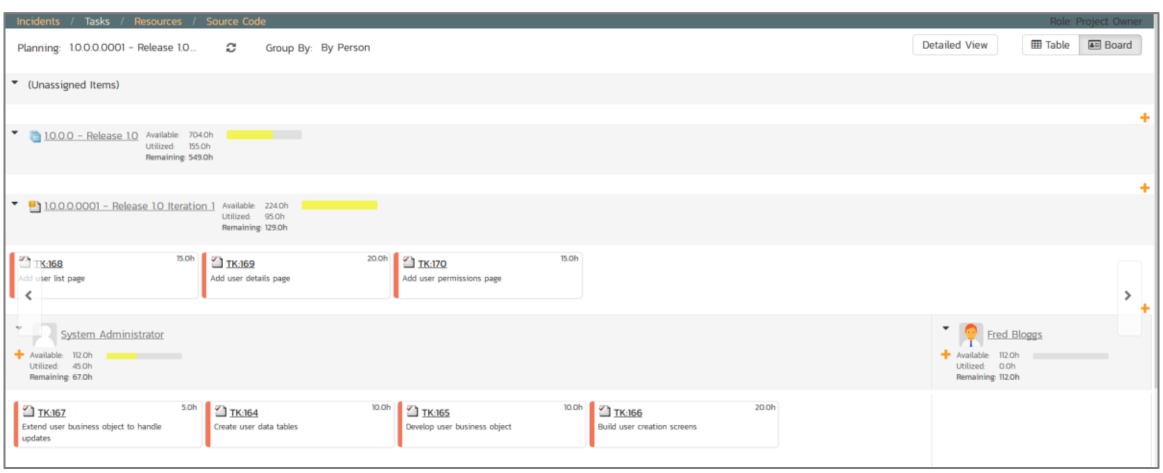
Click on the ‘Board’ option on the top-right of the screen to change to the Kanban board view:



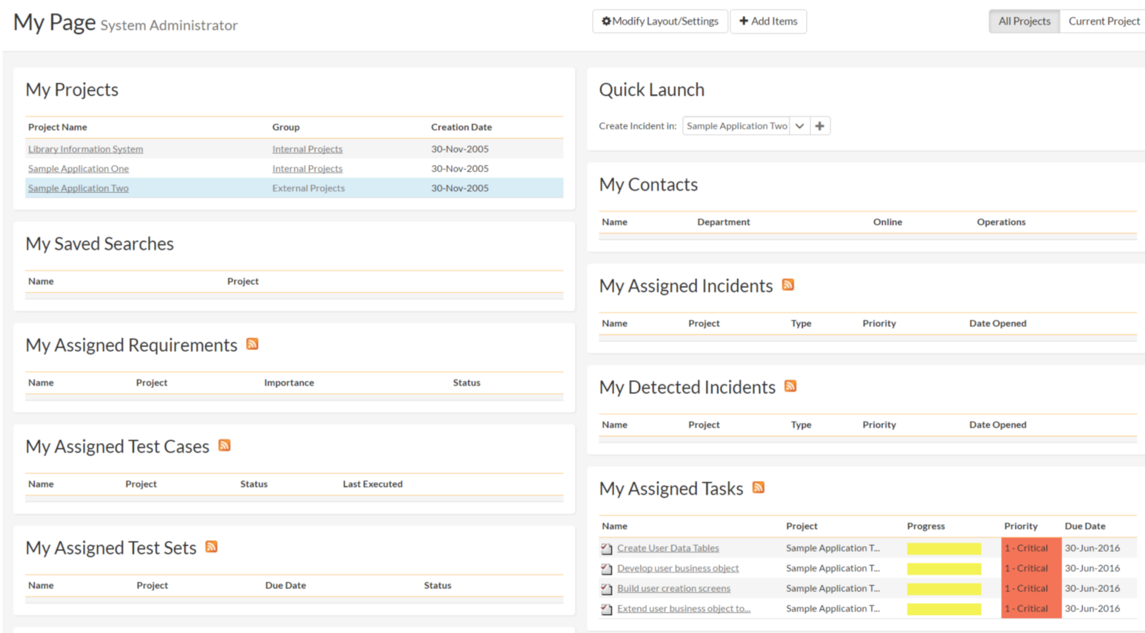
You can see at a glance which tasks are in each status (in this case, they are all marked as 'not started'). To see the distribution of tasks by person for specific iteration, change the release selection to 'Release 1.0 Iteration 1', and the 'Group By' dropdown to 'By Person':



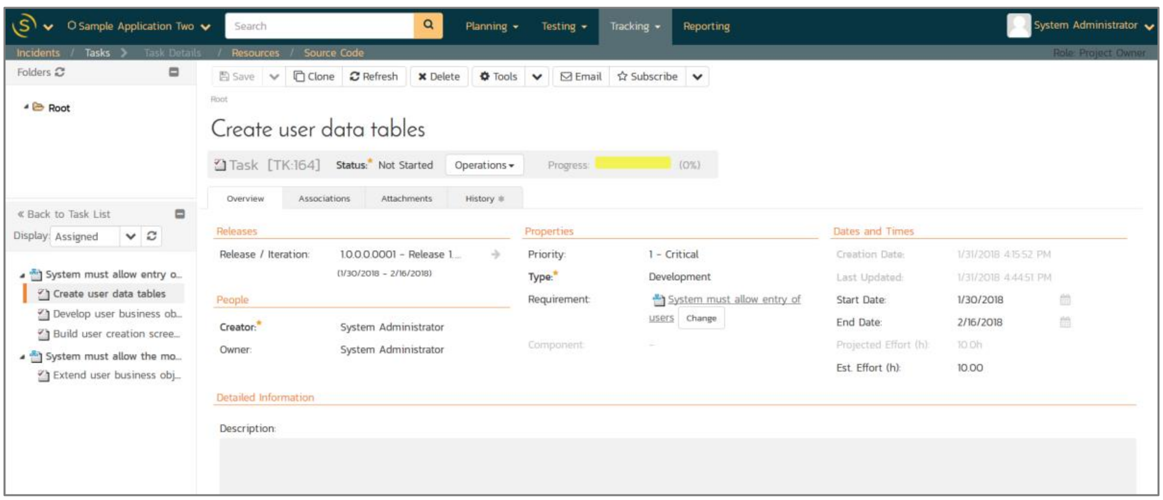
For our sample project we have two project members listed (included ourselves). As an example, select the first four tasks (which are all priority = 1 – Critical) and drag them to your user's section:



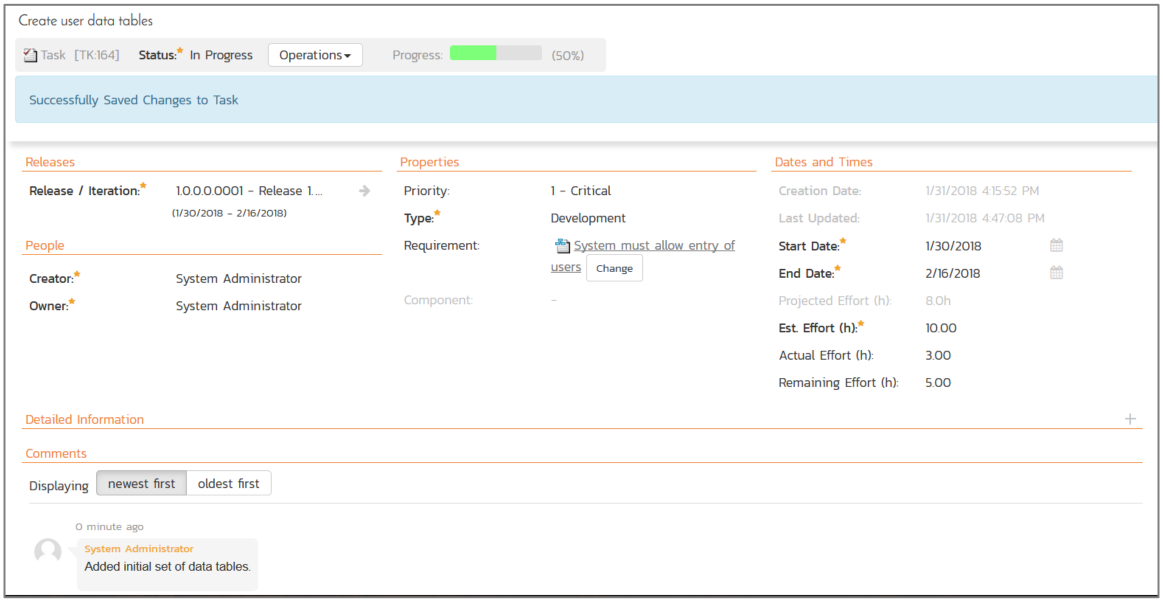
Now you can clearly see the four tasks that have been assigned to your user. To simulate how this would appear to a developer, click on the main SpiraTeam icon (in the top-left) to display your user's "My Page" dashboard:



This page lists all the development tasks that have been assigned to your user. Click on the first task "Create user data tables" to display the task details page:



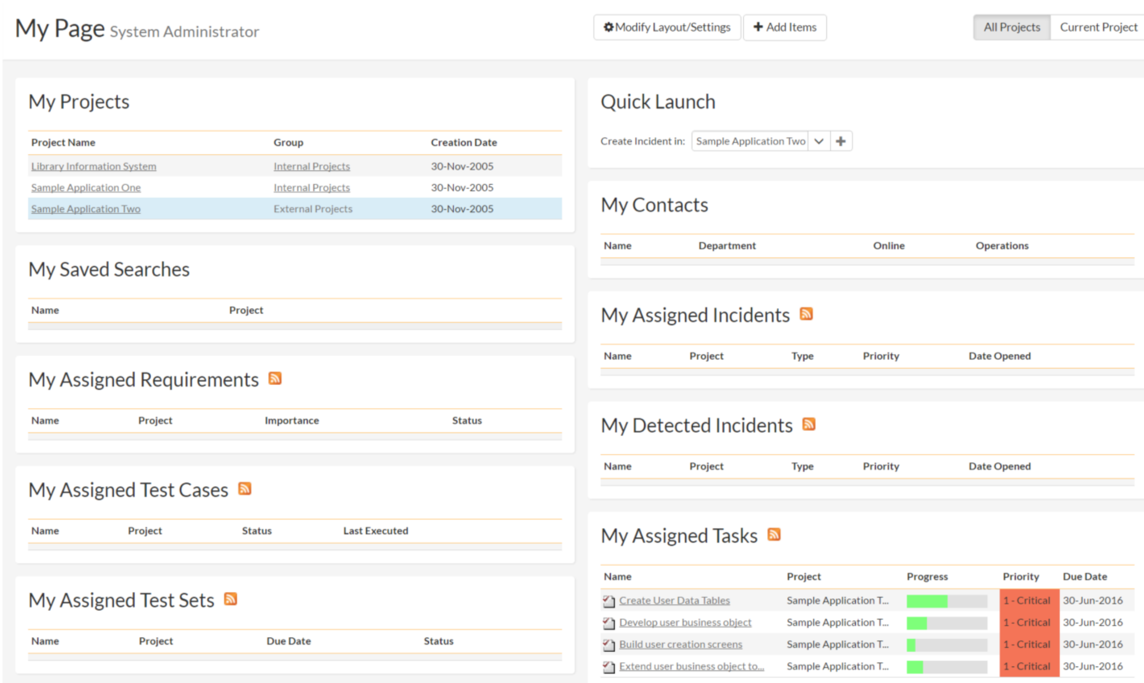
This task has been estimated at 10.0 hours and is currently not started. The next step is to start working on the assigned task and report back progress. As an example, click the workflow operation "Start Task", then enter an actual effort of 3.0 hours, a remaining effort of 5.0 hours and add a comment "Added initial set of data tables". Upon clicking 'Save', the progress indicator will reflect the changes and the new comment will have been added to the list:



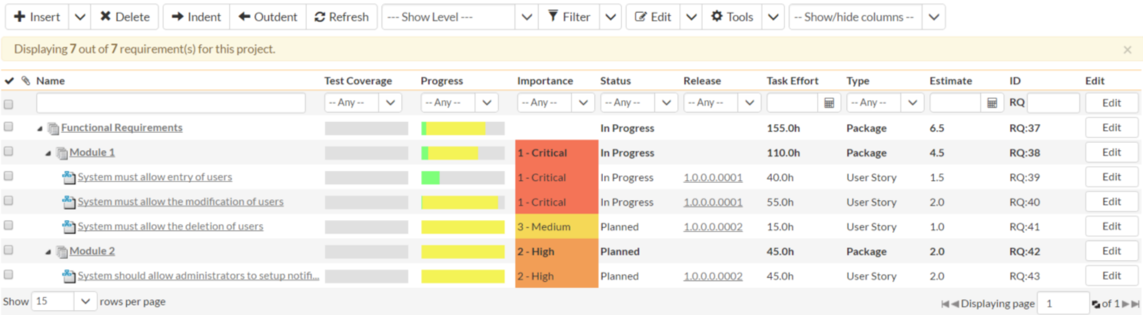
Now click on the other three assigned tasks, start them, and specify the following:

Requirement / Task	Est. Estimate	Actual Effort	Remaining Effort
Create user data tables	10.0h	3.0h	5.0h
Develop user business object	10.0h	2.0h	7.5h
Build user creation screens	20.0h	3.0h	18.0h
Extend user business object to handle updates	5.0h	0.5h	4.0h

After updating the assigned tasks, the ‘My Page’ dashboard will now reflect the changes:



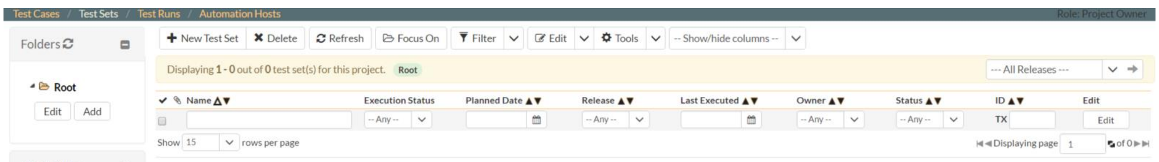
Similarly, for the project manager, clicking on Planning > Requirements to display the requirements list will show the task progress as it impacts the various requirements:



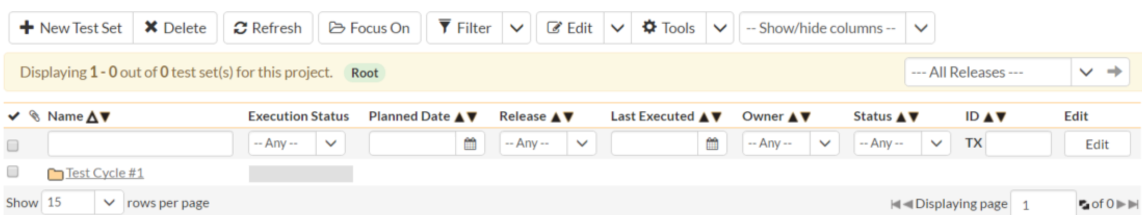
8. Scheduling the Testing Activities

Now that we have created our test plan for each release and iteration we now want to schedule the test cases for execution by our testers. As an example, we'll create a single test set (also known as test suites) that contains a list of test cases to be executed by a specific tester.

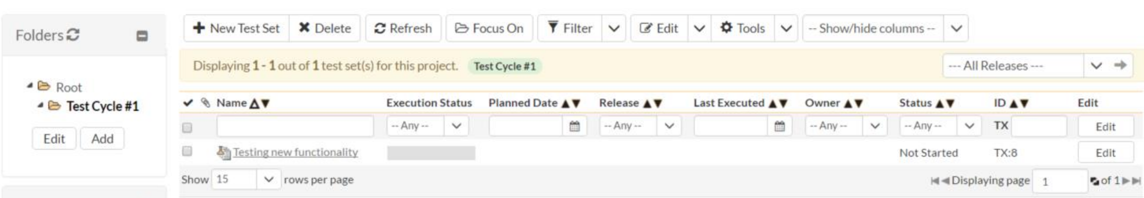
Click on the Testing > Test Sets menu option to display the project's test set list page:



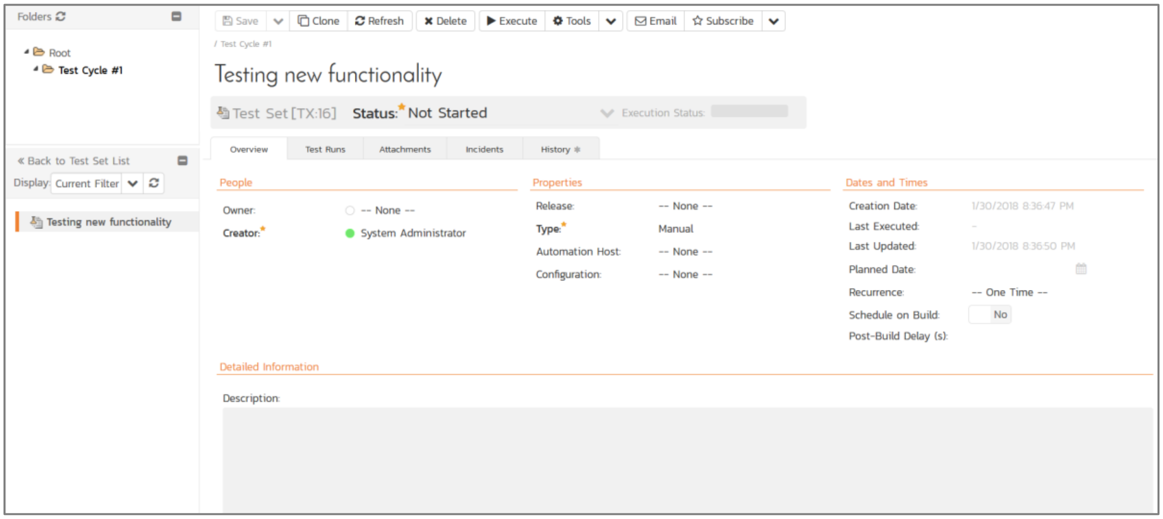
Initially the test set list will be empty and the folder tree on the left will only show "root". Click on the "Add" button beneath the folder tree, enter the new folder name "Test Cycle #1" and click the 'Add' button. Select that folder from the folder tree. After clicking 'Refresh' on the main toolbar, you now should have a single folder in the list on the right hand side:



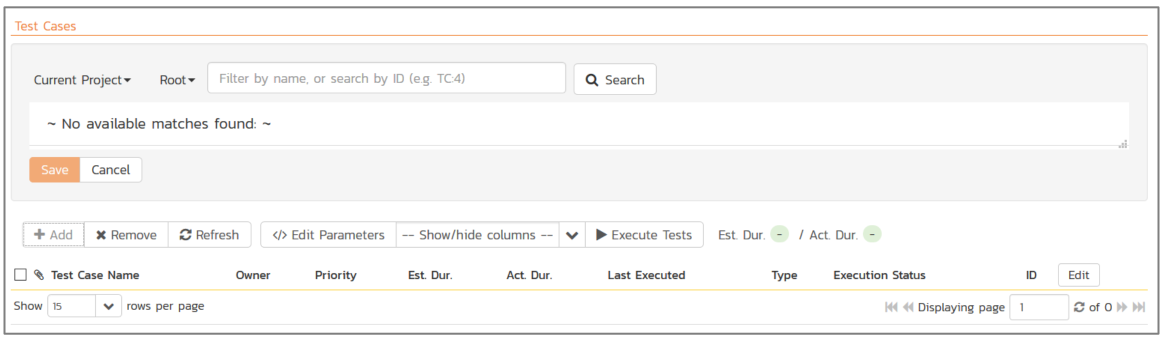
Now click on the folder and then click on the "New Test Set" from the main toolbar. Enter the name of the new test set "Testing new functionality" and click Save. You should now have the following test set list:



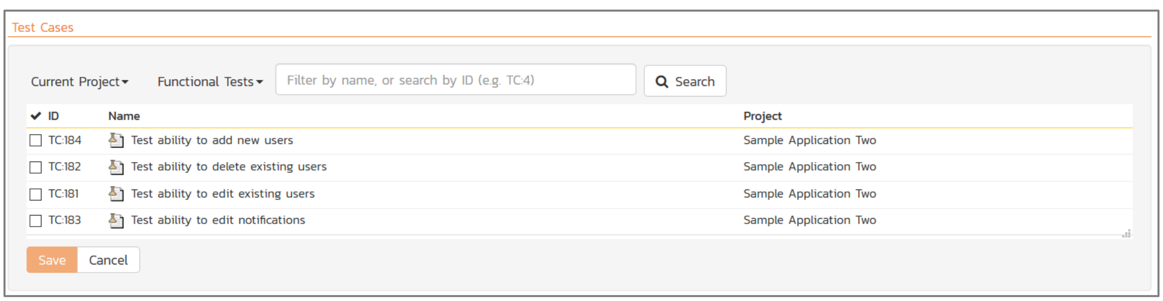
Click on the hyperlink for the test set to bring up the test set details page:



The first task is to add the appropriate test cases to this set. To do this, click on the “Add” button in the “Test Cases” section half way down the page to bring up the following dialog box:



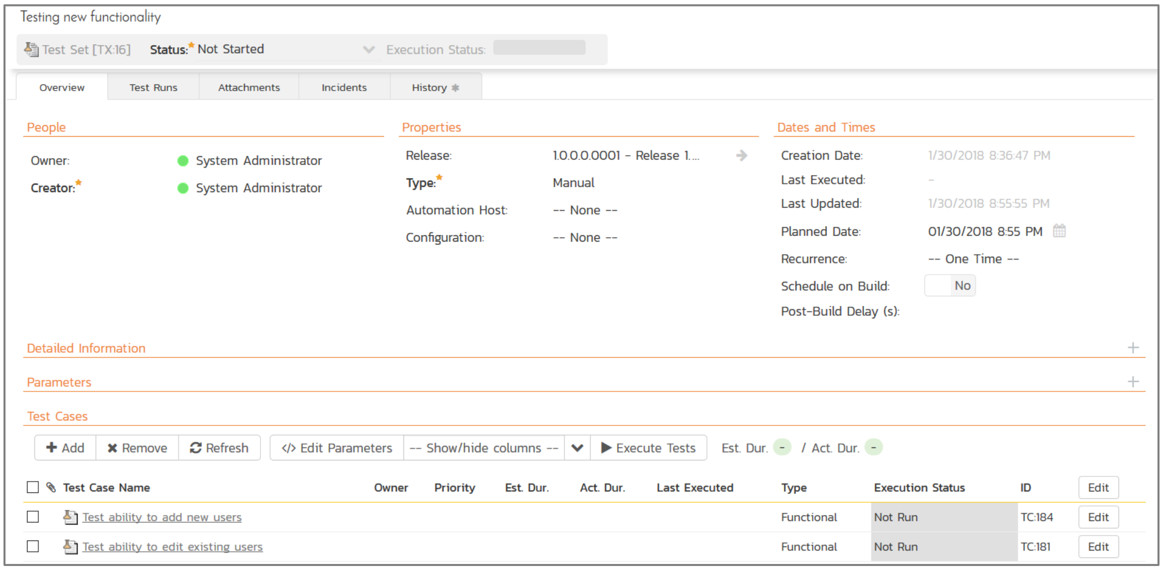
Choose the ‘Functional Tests’ folder and the test cases in that folder will be displayed:



Select the following test cases and click the ‘Add’ button:

- Test ability to add new users
- Test ability to edit existing users

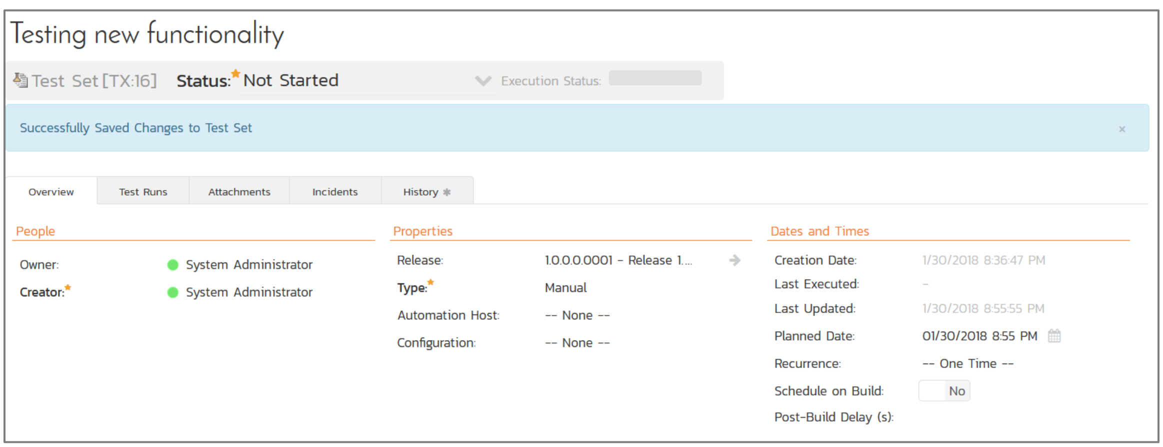
You should now have the following displayed:



The next step is to assign this test set to a specific release/iteration and to a particular tester. To do that, choose the following values for the following fields and click 'Save':

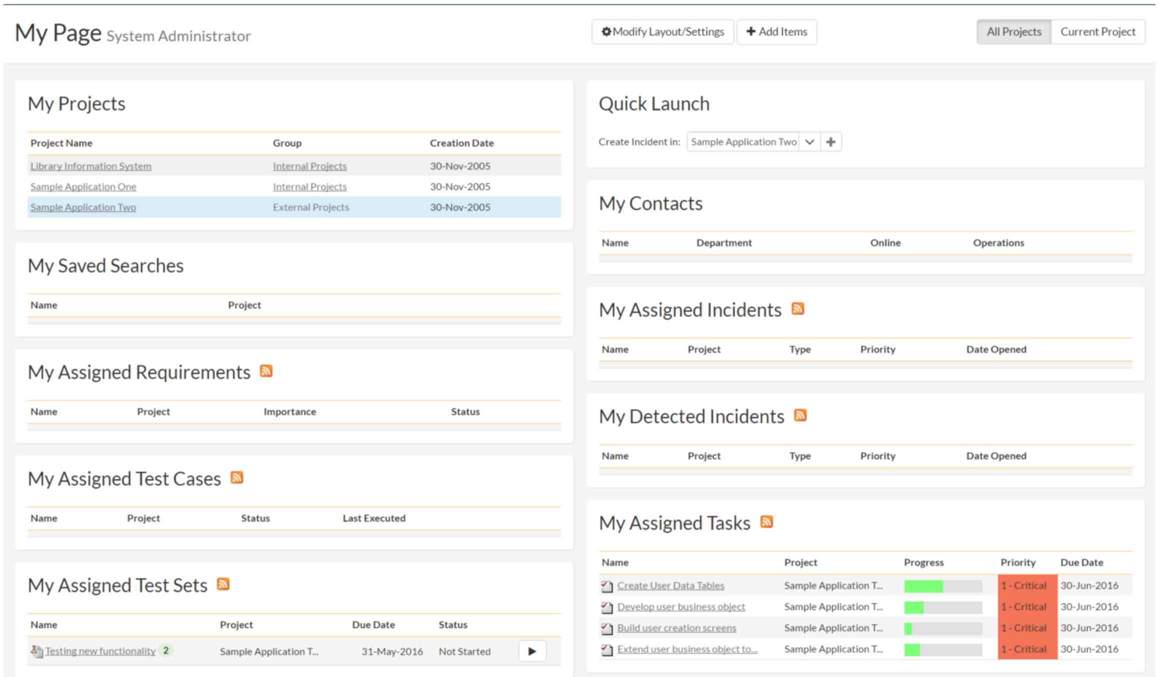
- Owner = System Administrator (your user)
- Release = Release 1.0 - Iteration 1
- Planned Date = (Today's Date).

You have now scheduled this test set to be executed by your user by the end of today against the first iteration of release 1.0:

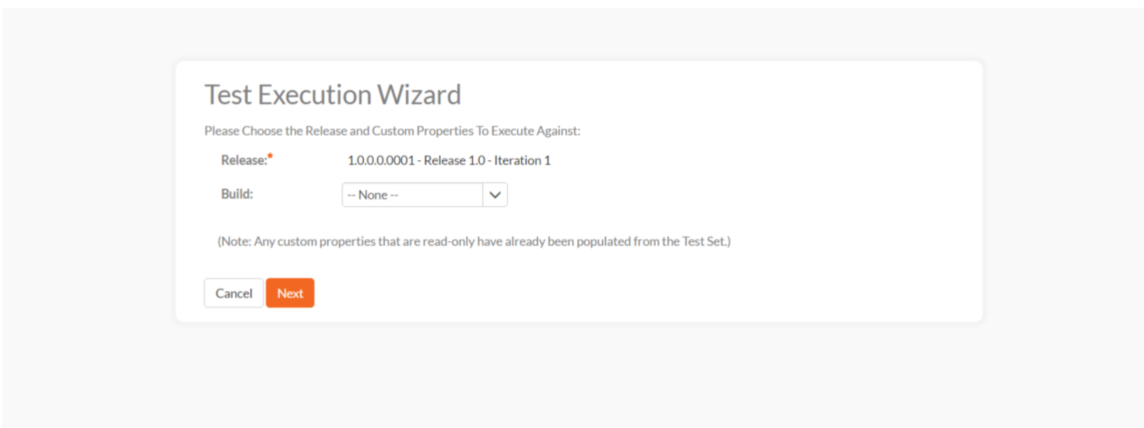


9. Running Tests and Logging Incidents

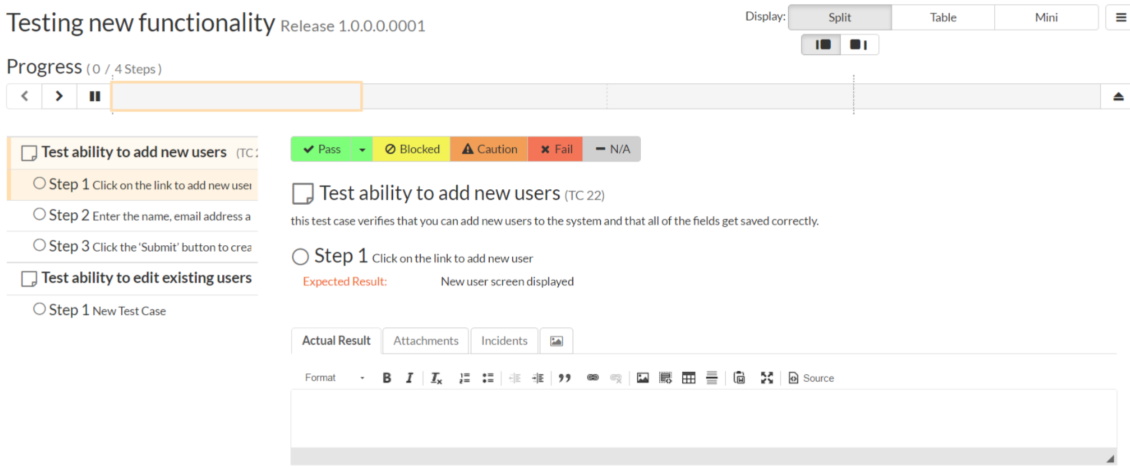
Now that you have scheduled the test set, if you go to the 'My Page' by clicking on the orange SpiraTeam application logo in the top-left, you'll see your newly assigned test set down on the left:



Click on the ‘Execute’ button (with the play icon) to the right of this new test set. That will start the test execution wizard:



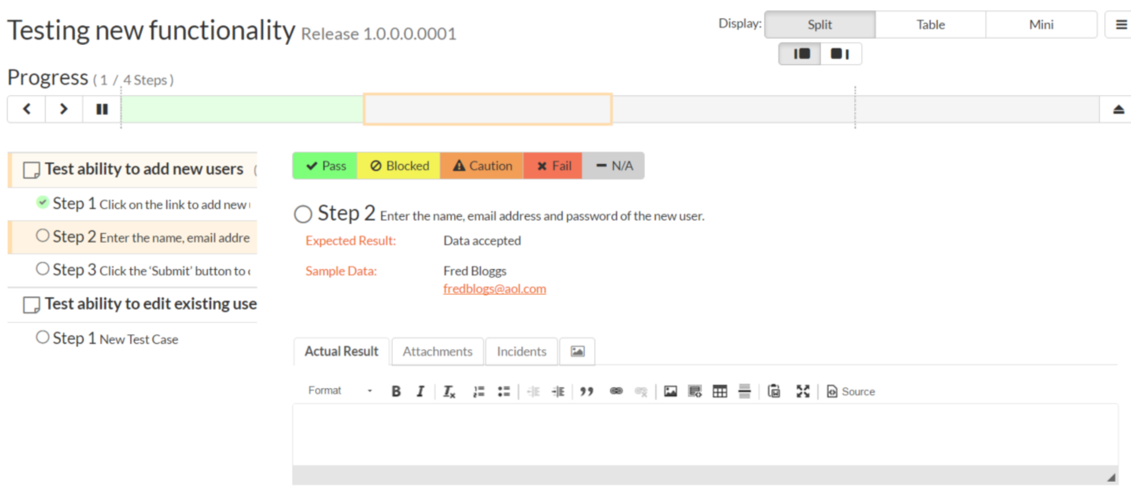
On the first screen, the release dropdown list will have been automatically pre-selected to the release/iteration specified in the test set, so just click the ‘Next’ button to move to the first test step in the first test case:



Note that when you first visit this page, you will be shown a quick guided tour of how the page works.

As a tester, you can progress through each of the test steps in each test case in the test set in turn. For each test step you can enter **Pass**, **Fail**, **Blocked, Caution, or Not Applicable**. If you enter any status other than Pass you need to enter a value for the 'Actual Result'. For a pass status, the Actual Result is optional.

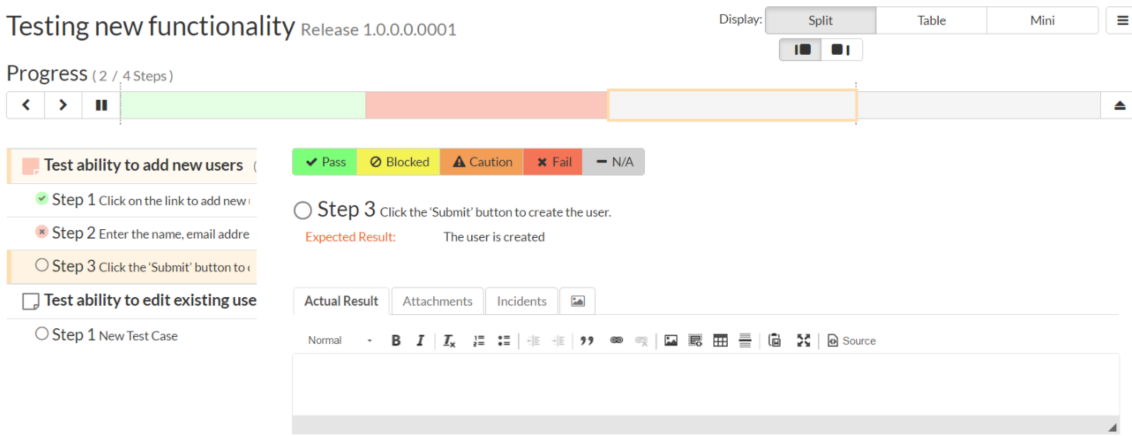
First, click on the Pass button to pass the first test step. As soon as you do, the test will automatically progress to the second test step:



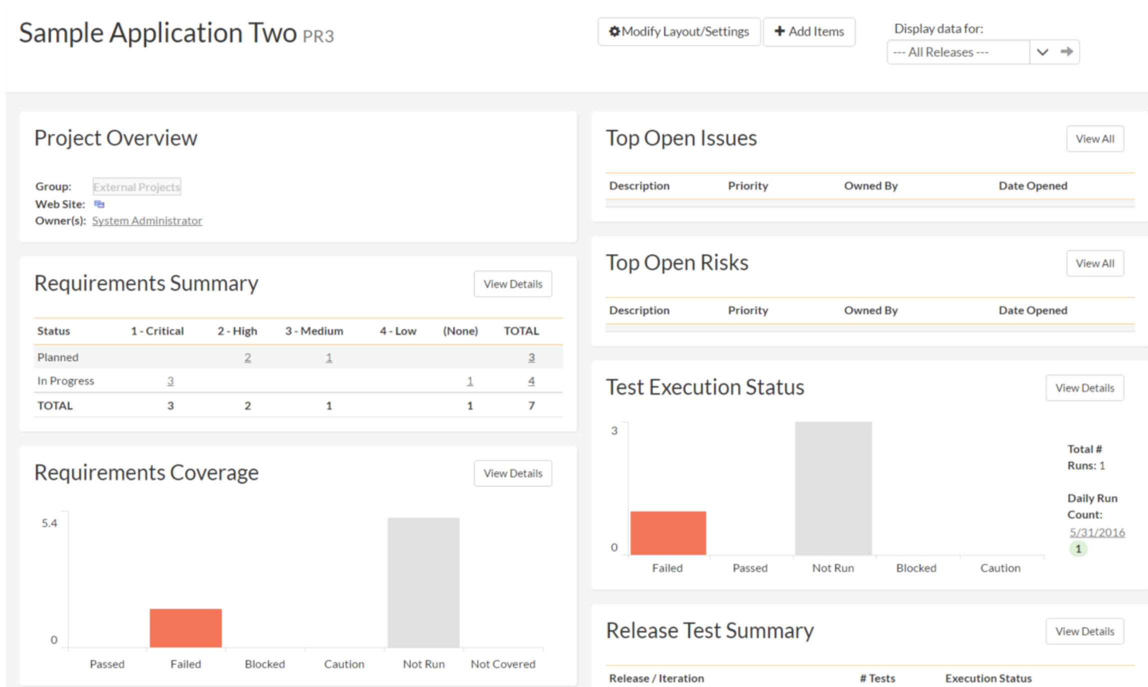
Now for the second test step, enter in the actual result field "Unable to enter the sample data as the fields were disabled". Before clicking the Fail button we also want to enter in the following fields in the Incident form (accessed by clicking the "Incidents" tab):

- Name = Error displaying user fields
- Type = Bug
- Priority = 2 – High

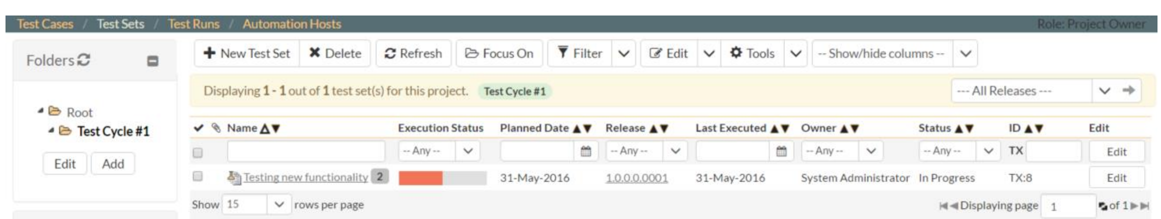
Now click the Fail button and you will have recorded a test failure and a new incident/defect:



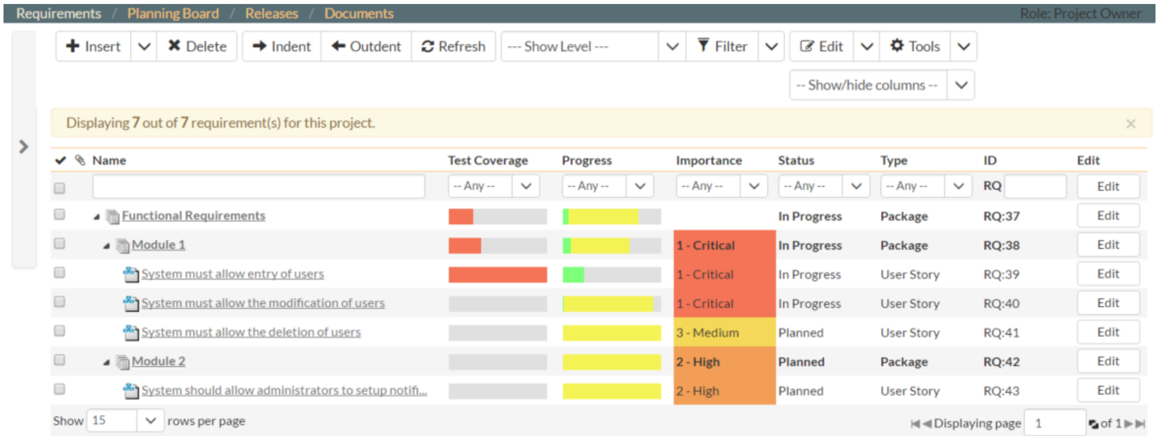
Now that we have logged the test failure and the new incident/defect, if you click on the main "Sample Application Two" option on the top-navigation bar, you'll be taken to the project homepage with the requirements and test case metrics now visible:



If you go to the Testing > Test Sets page, you also see the status of our test set:



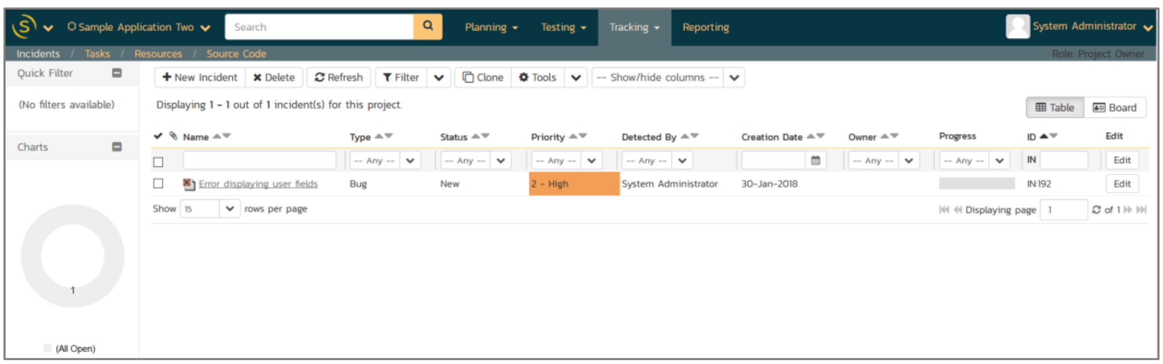
If you go to the Planning > Requirements page, you'll see the different requirements' test coverage and the status of the tests associated with each requirement:



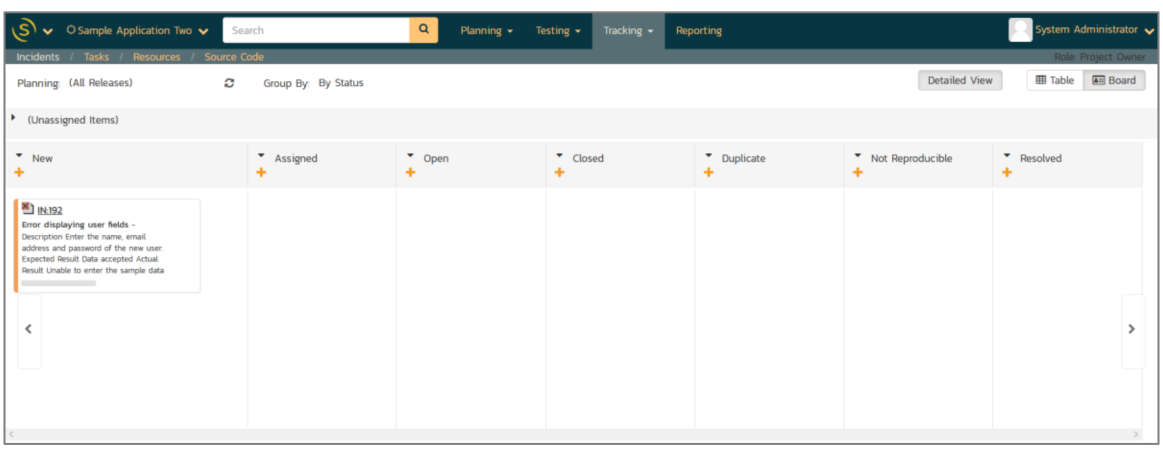
The final step in the process is to triage the logged defect and assign it to a developer to be fixed.

10. Triaging Issues and Defects

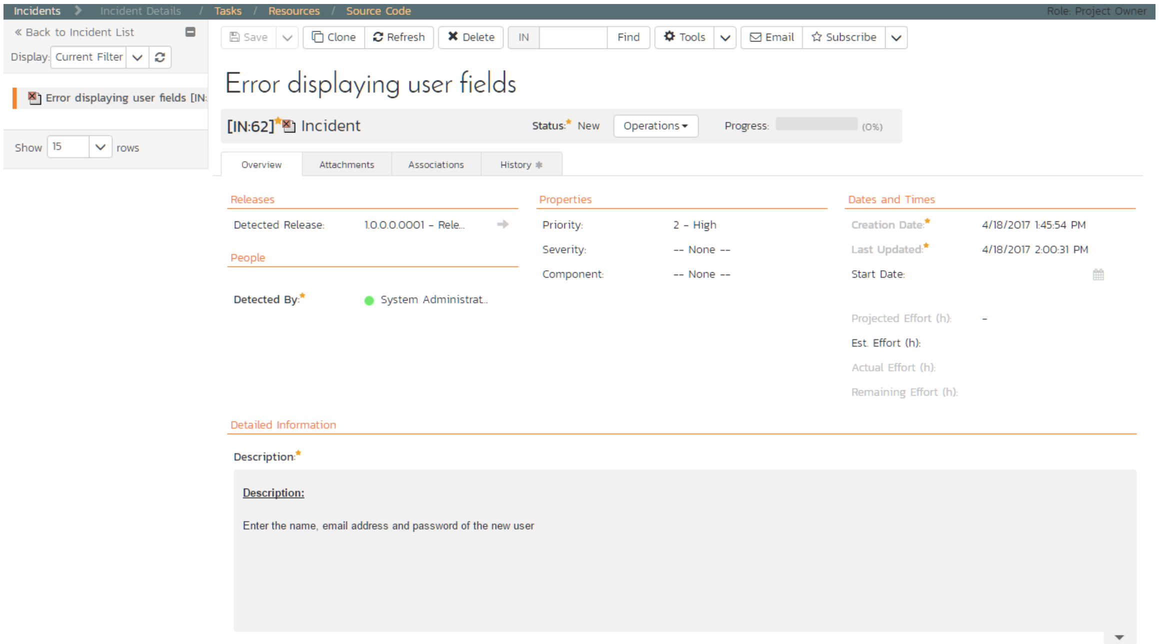
Now that a new incident has been logged, the next step in the process is to review the incident and assign it to a developer to be fixed. First, click on the Tracking > Incidents menu item. This will display the incident list page for the project:



You can also view the same list of incidents in a Kanban board view:



In either view, click on the hyperlink for the new incident “Error displaying user fields”. This will display the incident details page:

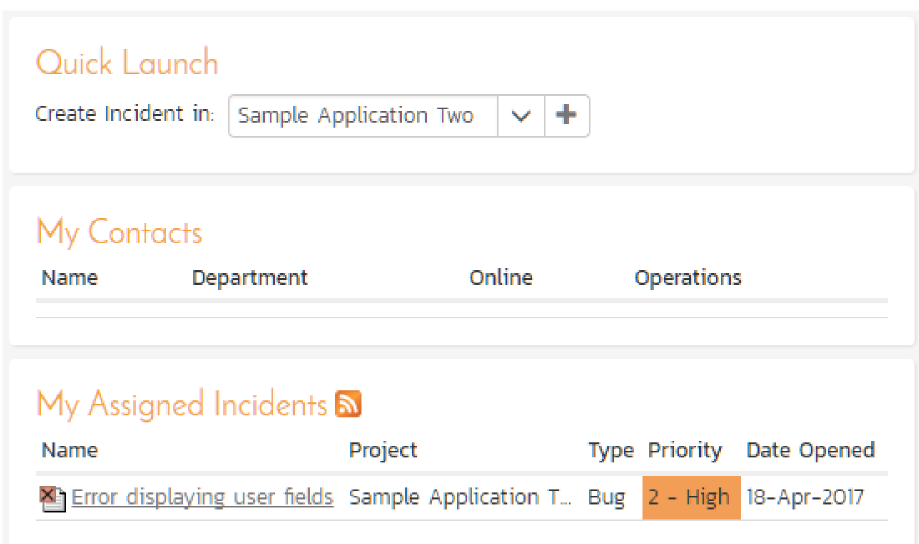


Click on the “Assign Incident” option in the “Operations” dropdown underneath the large incident name near the top of the page. This will switch the status of the incident from New > Assigned. You now need to enter the following fields:

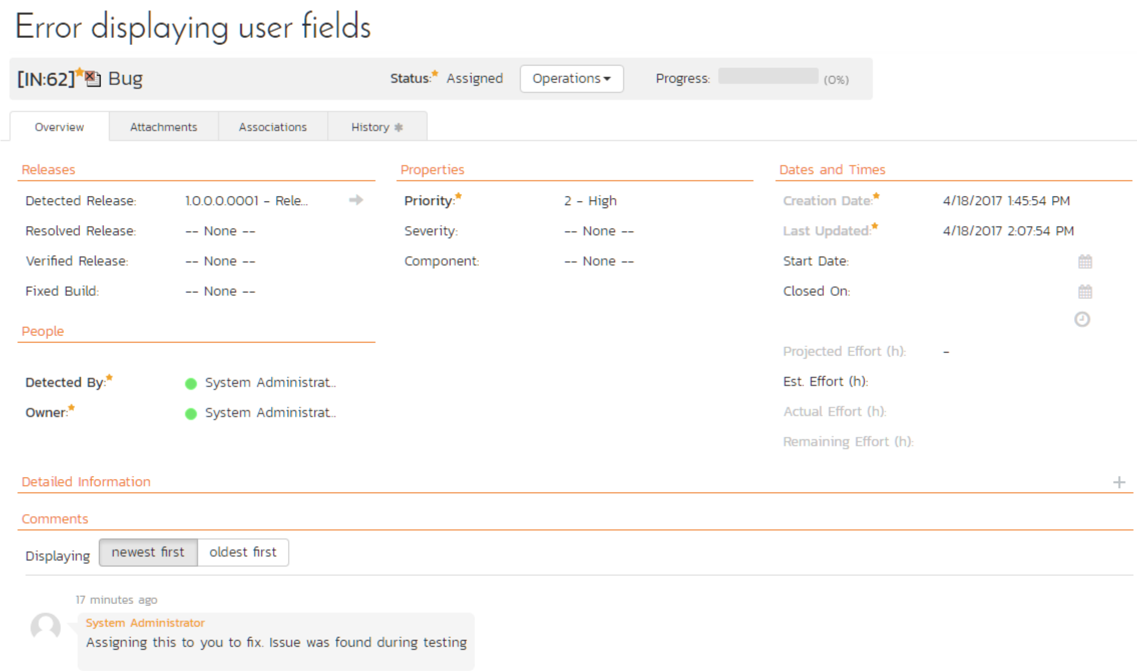
- Owner = System Administrator (your user)

Also add a new comment in the “Comments” section – “Assigning this to you to fix. Issue was found during testing.” Now click the Save button in the toolbar and the incident will be assigned to your user for fixing.

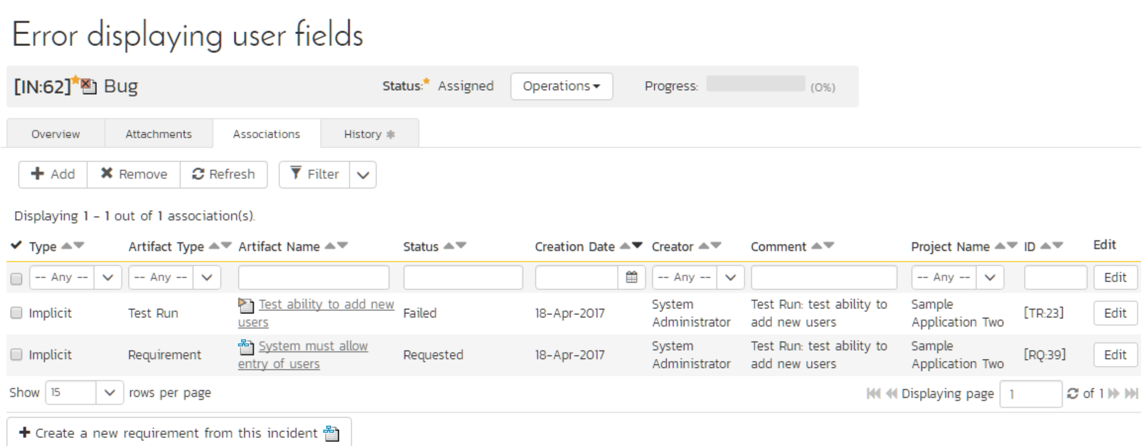
To simulate this process, go back to the “My Page” by clicking on the orange Spira icon in the top-left of the screen:



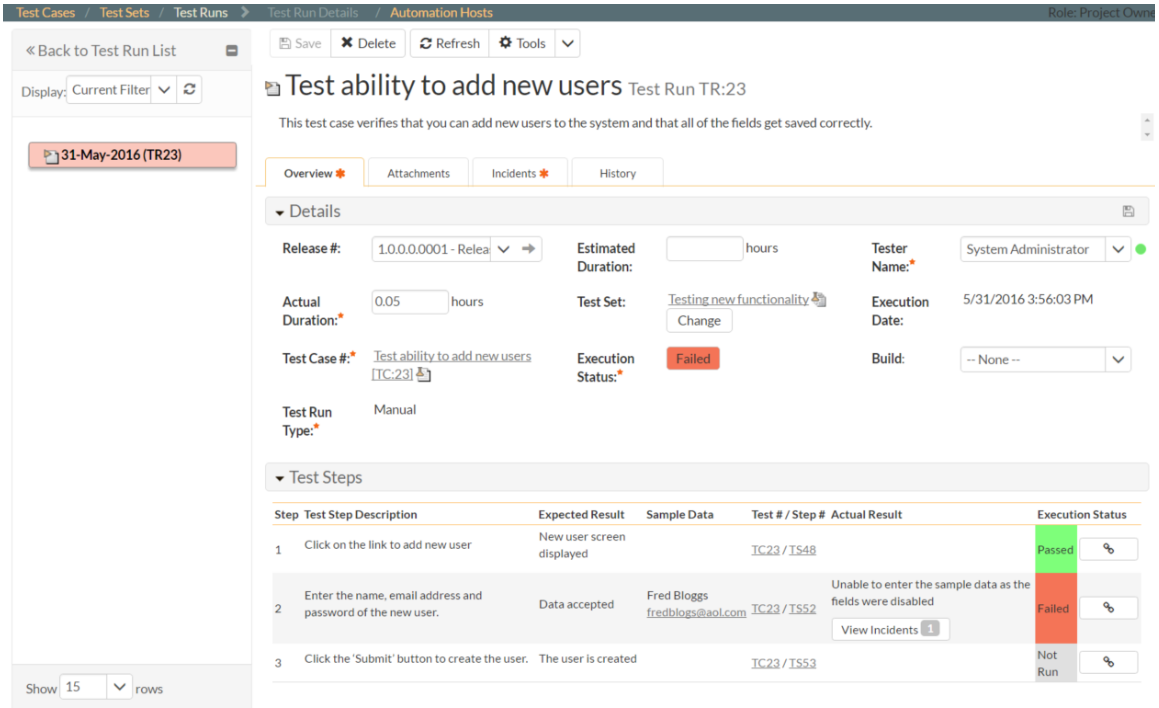
You can now see that you’ve been assigned an incident under the “My Assigned Incidents” widget (on the right-hand side). Now click on the hyperlink for the incident to bring up the incident details page:



You can see that the status has been changed to "Assigned" and the comment from the project manager is clearly visible. To help you reproduce the issue to be fixed, you can also click on the "Associations" tab to display the test run and requirements associated with this incident:



If you click on the test run hyperlink "Test ability to add new users", you will see the detailed information about the test execution that resulted in the bug being logged:



This allows the developer to retrace the steps taken by the tester and attempt to reproduce the issue. Assuming that we have been able to reproduce and fix the issue, we will now resolve the incident.

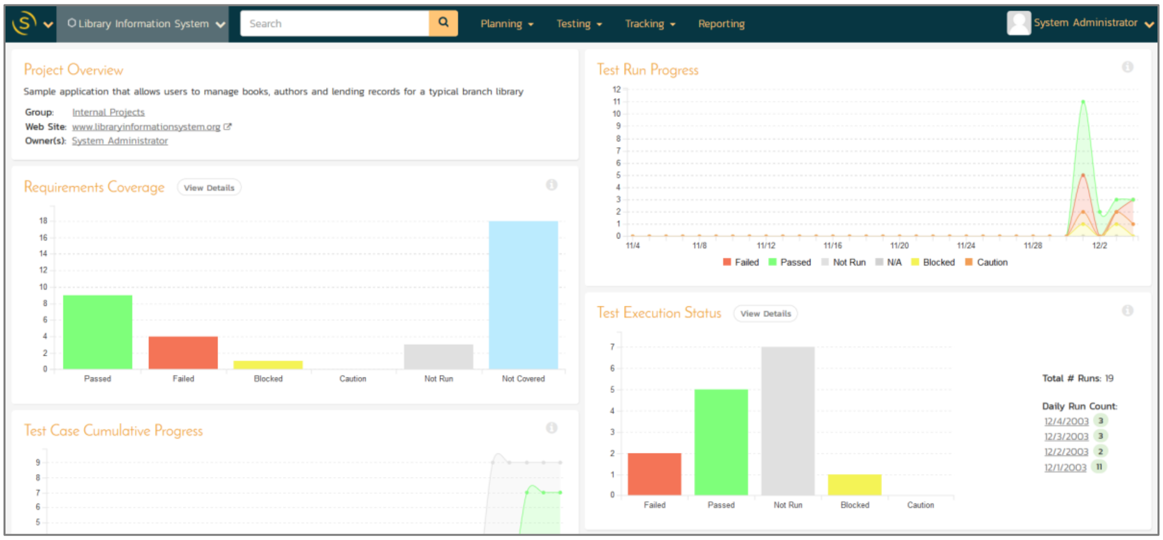
Back on the incident details screen, click on the workflow operation link "Resolve Incident", fill in the following fields and click Save:

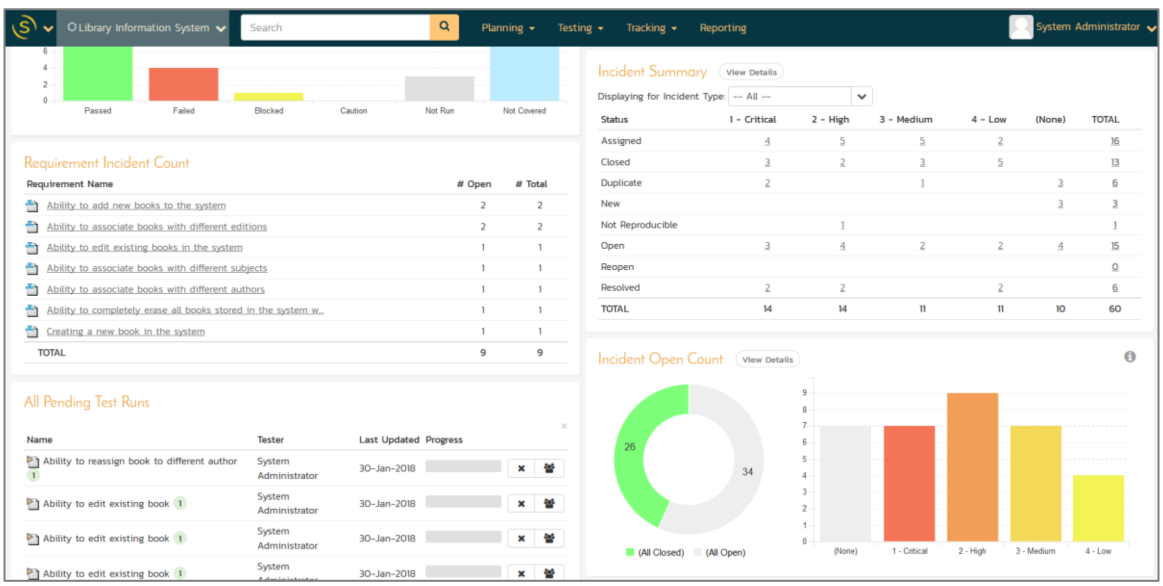
- Resolved Release = Release 1.0 - Iteration 2
- New Comment = "Fixed the incident."

The incident will now change from Assigned > Resolved and an email will be sent to the tester letting them know that they need to retest the test case and close the incident.

11. Reviewing Your Project

You can check on the overall status of the project by clicking the project name from the navigation bar. This will take you to the project home page. Below is what this home page looks like for a more complete project than we have been working through in this quick start guide. Note how you can change between several views (the buttons on the right) to show different information based on your role or current needs, or only show data for a particular release (see the dropdown beneath the project name on the left).





Congratulations! You have now completed the software development and testing lifecycle using SpiraTeam. For more information about any of the features, please refer to the SpiraTeam User Manual.

Legal Notices

This publication is provided as is without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

This publication could include technical inaccuracies or typographical errors. Changes are periodically added to the information contained herein; these changes will be incorporated in new editions of the publication. Inflectra® Corporation may make improvements and/or changes in the product(s) and/or program(s) and/or service(s) described in this publication at any time.

The sections in this guide that discuss internet web security are provided as suggestions and guidelines. Internet security is constantly evolving field, and our suggestions are no substitute for an up-to-date understanding of the vulnerabilities inherent in deploying internet or web applications, and Inflectra® cannot be held liable for any losses due to breaches of security, compromise of data or other cyber-attacks that may result from following our recommendations.

The section of the manual that describes modifying the Windows System Registry ("Registry") should only be attempted by experienced Windows administrators who are familiar with its organization and contents. Inflectra® cannot be held liable for any losses due to damage to the system registry made by inexperienced personnel.

SpiraTest®, SpiraPlan®, SpiraTeam® and Inflectra® are registered trademarks of Inflectra Corporation in the United States of America and other countries. Microsoft®, Windows®, Explorer® and Microsoft Project® are registered trademarks of Microsoft Corporation. All other trademarks and product names are property of their respective holders.

Please send comments and questions to:

Technical Publications

Inflectra Corporation

8121 Georgia Ave

Suite 504

Silver Spring, MD 20910

U.S.A.

support@inflectra.com