Taraxacum
The digital book

Acceptance Test Plan
Version 1.2

Revision History

Date	Version	Description	Author
2011-11-03	0.1	Initial Draft	Shahid Tarar & Adil Farid
2011-11-13	0.2	Added test cases	Shahid Tarar & Adil Farid
2011-12-04	0.3	Updated test plan	Shahid Tarar
2011-12-06	0.4	Spelling check and added comments. Updated 1.1, 1.3, 1.5, 6.1, 9.x	Anne Jon Schoonhoven
2011-12-09	0.5	Updated sections 1.1, 1.3, 1.5, 3, 4, 6, 6.2	Shahid Tarar
2011-12-10	0.6	Updated the Test Use cases	Adil Farid
2011-12-15	1.0	Updated section 3, 9.2, 10.2, 11	Shahid Tarar
2012-01-11	1.1	Reviewed whole document, check spelling and grammer mistakes	Shahid Tarar
2012-01-12	1.2	Update section 9	Adil Farid

Table of Contents

1. Introduction

1.1 Purpose of this document

The purpose of this document is to determine the quality of the application and to verify the expected results from the project. Test case specifications are also included in this document to validate the implemented functions.

1.2 Intended Audience

The intended audience of the document is:
- Supervisor –responsible for monitoring the status of the project, its direction and outcomes.
- Customer – to have comprehensive report of the system quality.
- Project team – to have an overview of what will be tested of the developed product.

1.3 Scope

The document explains the implemented test case(s). It also describes the generic system testing along with particular test devised for each phase of the project and a list of test case specifications. The results of conducting the test cases are given in Test report document.

1.4 Acronyms

Acronym or abbreviation	Definitions
DB	Database
MVC	Model-View-Controller

1.5 References

This document is based on the previously developed .
- Project description
- Requirements definition
- Design description

2. Test-plan introduction

The key objective of this part is to verify all the functions implemented in whole the project and the individual parts of the system as well. Keeping in mind the user's interaction with the Taraxacum the whole system is divided into three different categories to validate all the features and functionalities.

1. Guest
2. Registered user
3. Teacher

3. Test items

We will test View and Controller using interaction testing, and model using unit testing.

4. Features to be tested

All the implemented features described in Requirement definition will be tested to make sure that they fulfill the customer's requirements.
We have divided Taraxacum into following features for testing.

- User interface – proper layout, no spelling mistakes, proper HTML controls response and front end filters for user entry.

- Logic – all the implemented functions are working properly as per the user category. The added material, stored in a proper category. The functions behaves as per required

- Backend – All the data types are correctly created to handle the data so there is no anomaly occurring while communicating with the database.

5. Features not to be tested

The following features will not be tested.
- Performance
- Security
- Usability

6. Approach

The application is tested as per our implementation iterations. In first iteration we had been testing simple, if the data is being stored and being fetched as per requirement in database. In second iteration, the application is tested with different data and checked if all the client side validtions and functions implemented are working properly. For final iteration we are also testing application in Watir framework for automatic testing. At the backend the programmers are also Unit testing the system.

6.1 Approach to configuration and installation

The system is installed locally on the machine of the tester. Because the database can be regenerate and every requested moment, we can start each test session with a predefined and fixed test setting.

6.2 Item pass/fail criteria

In use case testing, the test will be marked passed if it shows the expected results where as test will be marked failed if it doesn't match the expected results from that.

6.3 Installation and Configuration

The testing of Taraxacum web portal doesn't affect the testing procedure at all as long as the right environment (mentioned in Section 9) is used to approach the web application through internet with a working machine.

6.4 Documentation problems

This document is based on mainly Requirement description and Design description which are almost done already.

7. Suspension criteria and resumption requirements

The MVC structure is implemented where data, logic, and front end are independent parts of the system but strongly coupled with each other. If any small error occurs, the application will not crash but if there is any big error which crashes the whole system should be fixed immediately.

8. Environmental needs

8.1 Hardware

- A working machine (Desktop, Laptop)

8.2 Software

- Operating system (Windows XP, Vista, Windows 7, Unix)
- Web browser (Google Chrome or Mozilla Firefox 3.6+)
- Visual Studio 2010
- SQL Server 2008 express
- MVC3 Framework.

8.3 Other

- Internet connection

9. Test procedure

We are using a simple template for each test case management. Name, Goal, Participating actors and Precondition are self explanatory. Type means whether the test should succeed or fail. Positive type means the test should succeed, where as negative type means the test should fail. Event flow contains the steps that will eventually produce the desired results. Output definition explains the results we are expecting after following the steps defined in event flow. And remarks are additional comments or small tests we should consider while testing.

9.1   Test case specifications

9.1.1   Test Cases for General Users

Name	**Login**	**GEN 1.0-P**
Goal	Authenticate the user	
Participating actors	Registered User	
Precondition	The user has an active account	
Type	Positive	
Event flow	The Registered User is at a **PUBLIC** page. 1. The user clicks on the login button. 2. A dialog with two fields is displayed 3. The user enters his valid username and password. 4. The user details are committed. 5. The user is redirected to the page from where he clicks on the login button. The user is logged in and user's first name is listed on the page.	
Output definition	The user is successfully logged in. And a log in message is displayed to the user	
Remarks		

Name	**Login**	**GEN 1.0-N**
Goal	Authenticate the user	
Participating actors	Registered User	
Precondition	The user has an active account	
Type	Negative	
Event flow	The Registered User is at a **PUBLIC** page. 1. The user clicks on the login button. 2. A dialog with two fields is displayed 3. The user enters invalid username and password. 4. The user details are committed. 5. The system inform the user with the message "combination of username and password is not found"	
Output definition	The system generates an error message and remains at the same page	
Remarks		

Name	**Register**	**GEN 1.1-P**
Goal	Register Guest to a registered account in the system	
Participating actors	Guest	
Precondition		
Test Type	Positive	
Event flow	The Guest is at a **PUBLIC** page. 1. Guest clicks on the register button. 2. A **REGISTER** page with the required fields is displayed. (See User Information Section) 3. Guest enters his valid credentials 4. Guest clicks on the submit button 5. Guest is showed a welcome page. The Guest is now a registered user and logged in having the role with the lowest privileges possible in the system.	
Output definition	The user record is properly inserted into the database, and welcome message is displayed to the user.	
Remarks		

Name	**Register**	**GEN 1.1-N**
Goal	Register Guest to a registered account in the system	
Participating actors	Guest	
Precondition		
Test Type	Negative	
Event flow	The Guest is at a **PUBLIC** page. 6. Guest clicks on the register button. 7. A **REGISTER** page with the required fields is displayed. (See User Information Section) 8. Guest enters his invalid credentials (similar to another user account) 9. Guest clicks on the submit button 10. A proper error message is generated. And the user remained on the same page.	
Output definition	The user record not inserted into the database, and error message is displayed to the user.	
Remarks		

Name	**Register**	**GEN 1.1-N**
Goal	Register Guest to a registered account in the system	
Participating actors	Guest	
Precondition		
Test Type	Negative (incomplete entry to fields)	
Event flow	The Guest is at a **PUBLIC** page.	

	1. Guest clicks on the register button.
	2. A **REGISTER** page with the required fields is displayed.
	3. Guest enters his incomplete credentials i.e. do not fill all the required fields.
	4. Guest clicks on the submit button
	5. A proper error message is generated, prompting user to fill all the required fields. And the user remained on the same page.
Output definition	No user record is inserted into the database, and error message is displayed to the user.
Remarks	

Name	**View Resource List**	**GEN 2.0-P**
Goal	View all the resources in the system	
Participating actors	User	
Precondition		
Test Type	Positive	
Event flow	The User is at the **HOME** page. 1. The user clicks on the Browse button. 2. The user is displayed the **BROWSE** page. 3. The user selects the "Resources" Button 4. The user is displayed a **RESOURCES RESULT** page listing all the resources.	
Output definition	The list of resources is displayed to the user from the database. If the database does not have any resources then an empty page is displayed with a message "No resources available".	
Remarks		

Name	**Search Resource**	**GEN 3.0-P**
Goal	Display a list of resources based on criteria.	
Participating actors	User	
Precondition		
Test Type	Positive	
Event flow	The User is at a **RESOURCES RESULT** page. 1. The user fills in his valid criteria in the search field. 2. The user clicks on the search button. 3. The user is displayed a **RESOURCES RESULT** page listing all the resources matching his criteria.	
Output definition	The list of resources is generated from the database.	
Remarks		

Name	**Search Resource**	**GEN 3.0-N**
Goal	Display a list of resources based on criteria.	
Participating actors	User	
Precondition		
Test Type	Negative	
Event flow	The User is at a **RESOURCES RESULT** page. 1. The user fills an invalid criteria in the search field (making sure that the search criteria will generate no search results) 2. The user clicks on the search button. 3. The user is displayed a **RESOURCES RESULT** page listing no resources.	
Output definition	An empty result list is displayed to the user	
Remarks		

Name	**Search Resource**	**GEN 3.0-N**
Goal	Display a list of resources based on criteria.	
Participating actors	User	
Precondition		
Test Type	Negative (Search for empty fields)	
Event flow	The User is at a **RESOURCES RESULT** page. 1. The user leave the search field empty. 2. The user clicks on the search button.	
Output definition	No results should be generated.	
Remarks		

Name	**View Resource**	**GEN 4.0-P**
Goal	Shows the contents and description of a resource	
Participating actors	User	
Precondition		
Test Type	Positive	
Event flow	The User is at a **RESOURCES RESULT** page. 1. The user clicks on the name of a listed resource. 2. The system displays the corresponding content en description of the resource on a **RESOURCE** page. The User is at a page containing a resource link. 1. The user clicks on link of the displayed resource. 2. The system displays the corresponding content en description of the resource on a **RESOURCE** page.	

	A **RESOURCE** page contains the resources content + the meta information
Output definition	A RESOURCE page containing proper information is displayed.
Remarks	Check the method for implementation of the Resource page. If GET method is used for accessing the Resource id from the URL, then try to provide an invalid Resource ID, and see the results.

Name	**View Topic List**	**GEN 5.0-P**
Goal	View all the topics in the system	
Participating actors	User	
Precondition		
Test Type	Positive	
Event flow	The User is at the **HOME** page. 1. The user clicks on the Browse button. 2. The user is displayed the **BROWSE** page. 3. The user selects the "Topics" Button 4. The user is displayed a **TOPICS RESULT** page listing all the resources.	
Output definition	All the topics in the database are listed	
Remarks	If there are no topics in the database, a proper message is displayed.	

Name	**Search Topic**	**GEN 6.0-P**
Goal	Display a list of topics based on criteria.	
Participating actors	User	
Precondition		
Test Type	Positive	
Event flow	The User is at a **TOPIC RESULT** page. 1. The user fills in a valid criteria in the search field. 2. The user clicks on the search button. 3. The user is displayed a **TOPIC RESULT** page listing all the topics matching his criteria.	
Result Definition	A list of topics is displayed based on the search criteria.	
Remarks		

Name	**Search Topic**	**GEN 6.0-N**
Goal	Display a list of topics based on criteria.	
Participating actors	User	
Precondition		

Test Type	Negative
Event flow	The User is at a **TOPIC RESULT** page. 1. The user fills in a invalid criteria in the search field. 2. The user clicks on the search button. 3. The user is displayed a **TOPIC RESULT** page listing all the topics matching his criteria.
Result Definition	No list of topics is displayed, with a proper message.
Remarks	

Name	**View Topic**	**GEN 7.0-P**
Goal	Shows the contents and description of a topic	
Participating actors	User	
Precondition		
Test Type	Positive	
Event flow	The User is at a **TOPIC RESULT** page. 1. The user clicks on the name of a listed resource. 2. The system displays the corresponding content en description of the resource on a **TOPIC** page The User is at a page containing a topic link. 1. The user clicks on link of the displayed resource. 2. The system displays the corresponding content en description of the resource on a **TOPIC** page A **TOPIC** page contains the list of resource + the meta information	
Output definition	A **TOPIC** page is displayed with proper information.	
Remarks	Check the method for implementation of the **TOPIC** page. If GET method is used for accessing the Topic ID from the URL, then try to provide an invalid Topic ID, and see the results.	

Name	**View Course List**	**GEN 8.0-P**
Goal	View all the courses in the system	
Participating actors	User	
Precondition		
Test Type	Positive	
Event flow	The User is at the **HOME** page. 1. The user clicks on the Browse button. 2. The user is displayed the **BROWSE** page. 3. The user selects the "Courses" Button 4. The user is displayed a **COURSES RESULT** page listing all the resources.	
Output definition	A list of courses is displayed in the **COURSES RESULT** page. Or proper message if there are no courses in the database.	

Remarks	

Name	**Search Course**	**GEN 9.0-P**
Goal	Display a list of course based on criteria.	
Participating actors	User	
Precondition		
Test Type	Positive	
Event flow	The User is at a **COURSES RESULT** page. 1. The user fills valid criteria in the search field. 2. The user clicks on the search button. 3. The user is displayed a **COURSES RESULT** page listing all the resources matching his criteria.	
Output definition	A list of courses is displayed in the **COURSES RESULT** page, based on the search criteria.	

Name	**Search Course**	**GEN 9.0-N**
Goal	Display a list of course based on criteria.	
Participating actors	User	
Precondition		
Test Type	Negative	
Event flow	The User is at a **COURSES RESULT** page. 1. The user fills invalid criteria in the search field. 2. The user clicks on the search button. 3. The user is displayed a **COURSES RESULT** page, with a proper message "No Records Found".	
Output definition	An empty **COURSES RESULT** page with a proper message is displayed to the user.	

Name	**View Course**	**GEN 10.0-P**
Goal	Shows the contents and description of a course	
Participating actors	User	
Precondition		
Test Type	Positive	
Event flow	The User is at a **COURSE RESULT** page. 1. The user clicks on the name of a listed resource. 2. The system displays the corresponding content en description of the resource on a **COURSE** page. The User is at a page containing a course link. 1. The user clicks on link of the displayed resource. 2. The system displays the corresponding content en description of the resource on a **COURSE** page. A **COURSE** page contains the resources content + the meta	

	information (see course Properties, Section)
Remarks	Check the method for implementation of the **COURSE RESULT** page. If GET method is used for accessing the Course ID from the URL, then try to provide an invalid Course ID, and see the results.

Name	**Use flashcard set**	**GEN 11.0 P**
Goal	To use flashcard set and study	
Participating actors	User	
Precondition		
Type	Positive	
Event flow	The User is at a **PUBLIC** page. 6. The user clicks on "Browse". 7. The user selects "Topics" button. 8. The user is displayed **TOPIC RESULT**. 9. The user selects one topic. 10. The user selects flashcard set from the list. 11. The system displays flashcards from that flashcard set to user.	
Output definition	The user views flashcards from the flashcard set and can study them.	
Remarks		

Name	**Search for flashcard sets**	**GEN 12.0-P**
Goal	To find flashcard sets by some criteria	
Participating actors	User	
Precondition		
Event flow	The User is at a **SHARE** page. 1. The user clicks on "Flashcards" link. 2. The user fills in valid criteria in the search field (that should return at least one result). 3. The user clicks on the search button. 4. The user is displayed a **FLASHCARD RESULT** page listing all the flashcards matching his criteria. The user views flashcards from the flashcard set and can study them.	
Results	The user views flashcards from the flashcard set and can study them	
Remarks	Alternatives • Check pagination • Check sorting of the search results • Check Empty field results	

Name	**Search for flashcard sets**	**GEN 12.0-N**

Goal	To find flashcard sets by some criteria
Participating actors	User
Precondition	
Type	Negative
Event flow	The User is at a **SHARE** page. 　5. The user clicks on "Flashcards" link. 　6. The user fills in in-valid criteria in the search field (that should not return any result). 　7. The user clicks on the search button. 　8. The user is displayed a **FLASHCARD RESULT** page with a proper notification of no results.
Results	User is notified with a proper message "No results matched"'.
Remarks	Alternative • Check pagination • Check sorting of the search results • Check Empty field results

Name	**Perform Exercise**	**GEN 13.0-P**
Goal	Perform and exercise about a topic, resource or course, and get an result.	
Participating actors	User	
Precondition		
Type	Positive	
Event flow	The User is at the **HOME** page. 　1. The user clicks on "Learn" link. 　2. The learn page is shown, providing a search field. 　3. The user searches for a topic using the search field. 　4. The user selects a topic to views his courses. 　5. The user selects one of the related courses. 　6. The user selects one of the available exercises. 　7. An **EXERCISE** page is shown. 　8. The user answers questions about exercises. 　9. When the user submits, the **EXERCISE RESULT** page is shown, showing result and scores.	
Results	Exercise is processed, and results are displayed to the user. If the user is logged on, the result of the exercise is stored in the system for later usage.	
Remarks	Alternatives • Client side verification of data entry & other options. • Check entry into database. • Confirm Exercise questions are random not fixed	

Name	**Answer Question**	**GEN 13.1-P**
Goal	Answer questions presented by exercise	
Participating actors	User	
Precondition		
Type	Positive	
Event flow	The User is at an **EXCERCISE** page. 1. The user is shown a list of multiple choice questions. 2. The user selects the answers of his choice 3. The user submits the exercise. 4. The user is shown the result page.	
Results	The user answers are processed and results are shown.	
Remarks	Alternatives • Client side authentication. • Database authentication • Questions are displayed randomly.	

9.1.2   Test Cases for Registered User

Name	**Create flashcard set**	**REG 1.0-P**
Goal	Add a new flashcard set to the system.	
Participating actors	Registered user	
Precondition	The user is logged in.	
Type	Positive	
Event flow	User is at the **HOME** page. 1. User clicks on the "My Content" button. 2. User is displayed the **MY CONTENT** page. 3. User clicks the "Add new flashcard set" Button 4. User is displayed an empty **EDIT FLASHCARD SET** page containing the required fields. 5. User fills in information about the new flashcard set. 6. User clicks on submit button 7. The system shows the User the **FLASHCARD SET** page showing the newly added flashcard set.	
Results	A new flashcard set is created in the system.	
Remarks	Alternatives • Check for invalid input. • Confirm entry into database.	

Name	**Edit flashcard set**	**REG 1.1**
Goal	Edit an existing flashcard set description.	

Participating actors	Registered user, must be owner of that set
Precondition	The user is logged in.
Type	Positive

Event flow	User is at the **MY CONTENT** page.
	1. User clicks on the "Edit" button next to the chosen flashcard set.
	2. User is displayed an **EDIT FLASHCARD SET** page containing the required fields, each filled in with information.
	    a. ADD FLASHCARD TO SET (REG 2.0)
	    b. EDIT FLASHCARD FROM SET (REG 2.1)
	    c. REMOVE FLASHCARD FROM SET (REG 2.2)
	3. User updates information about the flashcard set.
	4. User clicks on submit button
	5. The system shows the User the **FLASHCARD SET** page showing the edited flashcard set.

Results	The existing flashcard set is updated in the system
Remarks	Alternative Testing • Check for invalid entry.

Name	**Remove flashcard set**	**REG 1.2**
Goal	Remove a flashcard set from the system	
Participating actors	Registered user, must be owner of that set	
Precondition	The flashcard set contains no flashcards	
Type	Positive	
Event flow	User is at the **MY CONTENT** page. 1. User clicks on the "Remove" button next to the chosen flashcard set. 2. User is displayed dialog containing a confirmation question about removing the flashcard set. 3. User clicks on confirm button 4. The system shows the User the **MY CONTENT** page.	
Results	The flashcard set is removed from the system	
Remarks	Alternative Test case • Delete Flashcard set that contain flashcards. • Check database • Using the internet browser back button, browse to the previous page from where you just remove the flashcard set. And remove the same flashcard set again.	

Name	**Share flashcard set**	**REG 1.3**
Goal	To share flashcard set with others.	

Participating actors	Registered user
Precondition	The user is logged in.
Type	Positive
Event flow	User is at the **MY CONTENT** page. 1. User clicks on the "Edit" button next to the chosen flashcard set. 2. User is displayed an **EDIT FLASHCARD SET** page containing the required fields, each filled in with information. 3. User sets property Public to true. 4. User clicks on submit button 5. The system shows the User the **FLASHCARD SET** page showing the edited flashcard set.
Results	Flashcard set is now available to every user.
Remarks	• Confirm test results carefully with different users logged in.

Name	**Un-share flashcard set**	**REG 1.3-N**
Goal	To un-share flashcard set with others.	
Participating actors	Registered user	
Precondition	The user is logged in.	
Type	Negative	
Event flow	User is at the **MY CONTENT** page. 1. User clicks on the "Edit" button next to the chosen flashcard set. 2. User is displayed an **EDIT FLASHCARD SET** page containing the required fields, each filled in with information. 3. User sets property Public to false. 4. User clicks on submit button 5. The system shows the User the **FLASHCARD SET** page showing the edited flashcard set.	
Results	Flashcard set is not available to every user.	
Remarks	• Confirm test results carefully with different users logged in.	

Name	**Add flashcard to set**	**REG 2.0**
Goal	To add the new flashcard to existing flashcard set.	
Participating actors	Registered user	
Precondition	The user is logged in.	
Type	Positive	
Event flow	User is at the **EDIT FLASHCARD SET** page. 1. User clicks on the "Add flashcard". 2. User is displayed an empty **EDIT FLASHCARD** page	

	3. User enters valid information about the flashcard.
	4. User clicks on submit button
	5. The system shows the user the **EDIT FLASHCARD SET** page showing the added Flashcard in list.
Results	The new flashcard is added to the flashcard set.
Remarks	

Name	**Add flashcard to set**	**REG 2.0**
Goal	To add the new flashcard to existing flashcard set.	
Participating actors	Registered user	
Precondition	The user is logged in.	
Type	Negative	
Event flow	User is at the **EDIT FLASHCARD SET** page. 1. User clicks on the "Add flashcard". 2. User is displayed an empty **EDIT FLASHCARD** page containing the required fields. 3. User enters invalid information about the flashcard. 4. User clicks on submit button 5. The system shows the user the **EDIT FLASHCARD SET** page showing the added Flashcard in list. The new flashcard is added to the flashcard set.	
Results	The new flashcard is added to the flashcard set.	
Remarks		

Name	**Edit flashcard**	**REG 2.1**
Goal	To edit specific flashcard from flashcard set	
Participating actors	Registered user	
Precondition		
Type	Positive	
Event flow	User is at the **EDIT FLASHCARD SET** page. 1. User clicks on the "Edit flashcard" button next to the selected flashcard. 2. User is displayed **EDIT FLASHCARD** page containing the required fields filled with data. 3. User updates information about the flashcard. 4. User clicks on submit button 5. The system shows the User the **EDIT FLASHCARD SET** page with edited Flashcard in list.	
Results	The flashcard from the flashcard set is edited.	
Remarks	Alternative Test cases • Check for invalid data entries.	

Name	**Remove flashcard from set**	**REG 2.2**
Goal	To remove specific flashcard from flashcard set	
Participating actors	Registered user	
Precondition	The user is logged in.	
Type	Positive	

Event flow	User is at the **EDIT FLASHCARD SET** page.
	  1. User clicks on the "Remove flashcard" button next to the selected flashcard.
	  2. User is displayed dialog containing a confirmation question about removing the flashcard from set.
	  3. User clicks on confirm button
	  4. The system shows the User the **EDIT FLASHCARD SET**.	

Results	The flashcard is removed from the flashcard set.	
Remarks		

9.1.3   Test Cases for Student

Name	**Create personal note about resource**	**STD 1.0**
Goal	Add new note about resource	
Participating actors	Student	
Precondition	The student is logged in.	
Type	Positive	

Event flow	Student is at the **HOME** page.
	  1. Student clicks on the "Browse" button.
	  2. Student is displayed the **BROWSE** page.
	  3. Student clicks the "Resource" Button
	  4. Student is displayed a **RESOURCES RESULT** page listing all the resources.
	  5. Student selects Resource from the grid.
	  6. Student is displayed selected **RESOURCE** page.
	  7. Student clicks on "Add note" button.
	  8. Student is displayed empty **EDIT NOTE** page containing the required fields.
	  1. Student fills in valid information about the new note.
	  2. Student clicks on submit button
	  3. The system shows the Student the **RESOURCE** page showing the newly added note.	

Results	A new note is created in the system	
Remarks	Alternatives • Check for invalid entry	

	• & client side authentication

Name	**Create personal note about resource**	**STD 1.0**
Goal	Add new note about resource	
Participating actors	Student	
Precondition	The student is logged in.	
Type	Negative	
Event flow	Student is at the **HOME** page. 1. Student clicks on the "Browse" button. 2. Student is displayed the **BROWSE** page. 3. Student clicks the "Resource" Button 4. Student is displayed a **RESOURCES RESULT** page listing all the resources. 5. Student selects Resource from the grid. 6. Student is displayed selected **RESOURCE** page. 7. Student clicks on "Add note" button. 8. Student is displayed empty **EDIT NOTE** page containing the required fields. 9. Student fills in invalid information about the new note. 10. Student clicks on submit button	
Results	The system shows the error messages to the user. No record is entered into the database.	
Remarks	Alternatives • Check for invalid entry • & client side authentication	

Name	**Edit Note about Resource**	**STD 1.1**
Goal	Edit an existing notes description.	
Participating actors	Student	
Precondition	The student is logged in.	
Type	Positive	
Event flow	Student is at the **RESOURCES RESULT** page. 1. Student clicks on one resource from the grid. 2. Student clicks on the "Edit" button next to the chosen note on **RESOURCE** page. 3. Student is displayed an **EDIT NOTE** page containing the required fields, each filled in with information. 4. Student updates information about the note. 5. Student clicks on submit button 6. The system shows the Student the **RESOURCE** page showing the edited note.	
Results	The existing note is updated in the system.	
Remarks	Alternative	

| | • Check for invalid data. |
| | • Check for client side authentication. |

Name	**Remove Note from Resource**	**STD 1.2**
Goal	Remove a note from the resource	
Participating actors	Student	
Precondition		
Type	Positive	
Event flow	Student is at the **RESOURCE** page. 1. Student clicks on the "Remove" button next to the chosen note. 2. Student is displayed dialog containing a confirmation question about removing the note. 3. Student clicks on confirm button 4. The system shows the Student the **RESOURCE** page.	
Results	The note is removed from the resource.	
Remarks	Alternative Test Case • Check database • Try to remove the entry that is already removed (using browser back button)	

Name	**Print Note from the Resource**	**STD 1.3**
Goal	To send note to the printer	
Participating actors	Student	
Precondition		
Type	Positive	
Event flow	Student is at the **RESOURCE** page. 1. Student clicks on the "Print" button next to the chosen note. 2. Student is displayed dialog containing a confirmation dialog about printing note. 3. Student clicks that it is ok to print note. 4. The system sends file to the printer. 5. The system shows the student the **RESOURCE** page.	
Results	The note is printed to the paper.	
Remarks		

Name	**Print Note from the Resource**	**STD 1.3**

Goal	To send note to the printer
Participating actors	Student
Precondition	
Type	Negative
Event flow	Student is at the **RESOURCE** page. 1. Student clicks on the "Print" button next to the chosen note. 2. Student is displayed dialog containing a confirmation dialog about printing note. 3. Student clicks that it is ok to print note. 4. The system shows a proper message "printer is not attached".
Results	No note is printed to the paper. Proper message is displayed to the user
Remarks	

Name	**Create Note about topic**	STD 2.0
Goal	Add new note about topic	
Participating actors	Student	
Precondition	The student is logged in.	
Type	Positive	
Event flow	Student is at the **HOME** page. 1. Student clicks on the "Browse" button. 2. Student is displayed the **BROWSE** page. 3. Student clicks the "Topic" Button 4. Student selects Topic from the grid. 5. Student is displayed selected **TOPIC** page. 6. Student clicks on "Add note" button. 7. Student is displayed empty **EDIT NOTE** page containing the required fields. 8. Student fills in valid information about the new note. 9. Student clicks on submit button 10. The system shows the Student the **TOPIC** page showing the newly added note.	
Results	A new note is created in the system	
Remarks	Alternative test cases • Check for valid entry. • Check for local authentication of form fields. • Check the entry into the database.	

Name	**Edit Note about topic**	STD 2.1
Goal	Edit an existing notes description.	
Participating actors	Student	

Precondition	The student is logged in.
Type	Positive

Event flow	Student is at the **TOPIC** page.
	   1. Student clicks on the "Edit" button next to the chosen note.
	   2. Student is displayed an **EDIT NOTE** page containing the required fields, each filled in with information.
	   3. Student updates information about the note.
	   4. Student clicks on submit button
	   5. The system shows the Student the **TOPIC** page showing the edited note.

Results	The existing note is updated in the system.
Remarks	• Check for invalid entry • Check entry into the database • Check local authentication of the form fields.

Name	**Remove Note from topic**	STD 2.2
Goal	Remove a note from the topic	
Participating actors	Student	
Precondition		
Type	Positive	

Event flow	Student is at the **TOPIC** page.
	   1. Student clicks on the "Remove" button next to the chosen note.
	   2. Student is displayed dialog containing a confirmation question about removing the note.
	   3. Student clicks on confirm button
	   4. The system shows the Student the **TOPIC** page.	

Results	The note is removed from the resource. And a proper message is displayed to the user.	
Remarks	• Remove the already deleted note, using browser back button. • Check database to confirm the entry removal.	

Name	**Print Note from the Topic**	STD 2.3
Goal	To send note to the printer	
Participating actors	Student	
Precondition		
Type	Positive	

	Student is at the **TOPIC** page.
Event flow	1. Student clicks on the "Print" button next to the chosen note.
	2. Student is displayed dialog containing a confirmation dialog about printing note.
	3. Student clicks that it is ok to print note.
	4. The system sends file to the printer.
	5. The system shows the student the **TOPIC** page.
Results	The note is printed to the paper.
Remarks	

Name	**Subscribe to Course**	STD 3.0
Goal	Subscribe student to specific course	
Participating actors	Student	
Precondition	The student is logged in.	
Type	Positive	
Event flow	Student is at the **HOME** page.	
	1. Student clicks on the "Browse" button.	
	2. Student is displayed the **BROWSE** page.	
	3. Student clicks the "Course" Button	
	4. The system shows the Student **COURSE RESULT** page with all courses.	
	5. Student clicks on the "Subscribe" button next to the Course from the list.	
	6. The system shows the Student the **COURSE RESULT** page showing the newly added subscription.	
Results	A new subscription is created in the system	
Remarks		

Name	**Unsubscribe from Course**	STD 3.1
Goal	Subscribe student to specific course	
Participating actors	Student	
Precondition	Student must be logged in and subscribed to the course.	
Type	Positive	
Event flow	Student is at the **MY CONTENT** page.	
	1. Student clicks on the "Remove" button next to the chosen subscription.	
	2. Student is displayed dialog containing a confirmation question about removing the subscription.	
	3. Student clicks on confirm button	
	4. The system shows the Student the **MY CONTENT** page.	
Results	The existing subscription is removed from the system.	

Remarks	• Check the subscription list for the course. • Check removal on entry from database. • Check the course link manually (copy & paste) to check student subscription.	

Name	**Signal errors in Resource**	**STD 4.0**
Goal	To signalize error to the owner of the specific resource	
Participating actors	Student	
Precondition	Student must be logged in.	
Type	Positive	
Event flow	Student is at the **RESOURCE** page. 1. Student clicks on the "Signal error" button on the RESOURCE page. 2. Student is displayed dialog where message for teacher about the error can be entered. 3. Student clicks on confirm button 4. The system shows the Student the **RESOURCE** page.	
Results	The error message is sent to teachers email.	
Remarks	• Client side Form validation. • Check entry into the database.	

9.1.4 Test Cases for Teachers

Name	**Create Topic**	**TCH 1.0-P**
Goal	Test case to add a new topic to the system.	
Participating actors	Teacher	
Precondition	The teacher is logged in.	
Test Type	Positive	
Event flow	Teacher is at the **HOME** page. 8. Teacher clicks on the "My Content" button. 9. Teacher is displayed the **MY CONTENT** page. 10. Teacher clicks the "Add topic" Button 11. Teacher is displayed an empty **EDIT TOPIC** page containing the required fields. 12. Teacher fills in information about the new topic. 13. Teacher clicks on submit button 14. The system shows the Teacher the **TOPIC** page showing the newly added topic. A new topic is created in the system	
Output definition	A message is displayed showing "Topic created successfully". And new record is inserted into the database	

Remarks	

Name	**Create Topic**	**TCH 1.0-N**
Goal	Test case to add a new topic to the system.	
Participating actors	Teacher	
Precondition	The teacher is logged in.	
Test Type	Negative	
Event flow	Teacher is at the **HOME** page. 1. Teacher clicks on the "My Content" button. 2. Teacher is displayed the **MY CONTENT** page. 3. Teacher clicks the "Add topic" Button 4. Teacher is displayed an empty **EDIT TOPIC** page containing the required fields. 5. Teacher fills in invalid information about the new topic. (Enter Topic already exist) 6. Teacher clicks on submit button A error message is displayed.	
Output definition	A message is displayed showing "Topic already exists". And no record is added to the database.	
Remarks	Alternative • Check for client side validation	

Name	**Edit Topic**	**TCH 1.1-P**
Goal	Edit an existing topics description.	
Participating actors	Teacher	
Precondition	The teacher is logged in.	
Test Type	Positive	
Event flow	Teacher is at the **MY CONTENT** page. 6. Teacher clicks on the "Edit" button next to the chosen topic. 7. Teacher is displayed an **EDIT TOPIC** page containing the required fields, each filled in with information. 8. Teacher updates information about the topic. 9. Teacher clicks on submit button 10. The system shows the Teacher the **TOPIC** page showing the edited topic. The existing topic is updated in the system	
Output definition	A success message is displayed "Successfully update the topic" And the current topic is updated in the database.	
Remarks		

Name	**Edit Topic**	**TCH 1.1-N**

Goal	Edit an existing topics description.
Participating actors	Teacher
Precondition	The teacher is logged in.
Test Type	Negative
Event flow	Teacher is at the **MY CONTENT** page. 1. Teacher clicks on the "Edit" button next to the chosen topic. 2. Teacher is displayed an **EDIT TOPIC** page containing the required fields, each filled in with information. 3. Teacher updates information about the topic. (update the topic name, that is similar to another topic) 4. Teacher clicks on submit button The system displays an error message.
Output definition	An error message is generated, and no record is updated in the database.
Remarks	

Name	**Remove Topic**	**TCH 1.2-P**
Goal	Remove a topic from the system	
Participating actors	Teacher	
Precondition	The topic contains no resource	
Test Type	Positive	
Event flow	Teacher is at the **MY CONTENT** page. 5. Teacher clicks on the "Remove" button next to the chosen topic. 6. Teacher is displayed dialog containing a confirmation question about removing the topic. 7. Teacher clicks on confirm button 8. The system shows the Teacher the **MY CONTENT** page. The topic is removed from the system	
Output definition	The Topic record is deleted from the database, and a confirmation message is displayed to the user.	
Remarks		

Name	**Create Resource**	**TCH 2.0-P**
Goal	Add a new resource to the system.	
Participating actors	Teacher	
Precondition	The teacher is logged in.	
Test Type	Positive	
Event flow	Teacher is at the **HOME** page.	

      6. Teacher clicks on the "My Content" button.
      7. Teacher is displayed the **MY CONTENT** page.
      8. Teacher clicks the "Add resource" Button
      9. Teacher is displayed an empty **EDIT RESOURCE** page containing the required fields.
      10. Teacher fills in information about the new resource.
      11. Teacher uploads a file, or adds a text, or adds a URL
      12. Teacher clicks on submit button
      13. The system shows the Teacher the **RESOURCE** page showing the newly added topic.

A new resource is created in the system. The resource is added to the default topic "uncategorized".

Output definition	The resource record is inserted into the database, and a confirmation message is displayed to the teacher.
Remarks	

Name	**Create Resource**	**TCH 2.0-N**
Goal	Add a new resource to the system.	
Participating actors	Teacher	
Precondition	The teacher is logged in.	
Test Type	Negative	
Event flow	Teacher is at the **HOME** page. 1. Teacher clicks on the "My Content" button. 2. Teacher is displayed the **MY CONTENT** page. 3. Teacher clicks the "Add resource" Button 4. Teacher is displayed an empty **EDIT RESOURCE** page containing the required fields. 5. Teacher fills in already existed information for the new resource. 6. Teacher uploads a file, or adds a text, or adds a URL 7. Teacher clicks on submit button An error message is generated by the system.	
Output definition	An error message is generated, and the resource record is not inserted into the database.	
Remarks	• Check for client side authentication.	

Name	**Edit Resource**	**TCH 2.1-P**
Goal	Edit an existing topics description.	
Participating actors	Teacher	
Precondition	The teacher is logged in.	
Test Type	Positive	
Event flow	Teacher is at the **MY CONTENT** page. 6. Teacher clicks on the "Edit" button next to the chosen resource.	

7. Teacher is displayed an **EDIT RESOURCE** page containing the required fields, each filled in with information.
8. Teacher updates valid information about the resource.
   a. Optional: Teacher uploads a file, or changes a text, or changes a URL
9. Teacher clicks on submit button
10. The system shows the Teacher the **RESOURCE** page showing the edited resource.

The existing resource is updated in the system

Output definition	The record is updated in the database, and the confirmation message is generated.
Remarks	

Name	**Edit Resource**	TCH 2.1-N
Goal	Edit an existing topics description.	
Participating actors	Teacher	
Precondition	The teacher is logged in.	
Test Type	Negative	
Event flow	Teacher is at the **MY CONTENT** page. 1. Teacher clicks on the "Edit" button next to the chosen resource. 2. Teacher is displayed an **EDIT RESOURCE** page containing the required fields, each filled in with information. 3. Teacher updates invalid information about the resource (name already exist). 4. Teacher clicks on submit button Error message is displayed.	
Output definition	An error message is displayed and no record is updated in the database.	
Remarks	• Client side authentication check.	

Name	**Remove Resource**	TCH 2.2
Goal	Remove a resource from the system	
Participating actors	Teacher	
Precondition		
Test Type	Positive	
Event flow	Teacher is at the **MY CONTENT** page. 1. Teacher clicks on the "Remove" button next to the chosen resource. 2. Teacher is displayed dialog containing a confirmation question about removing the resource. 3. Teacher clicks on confirm button	

	4. The system shows the Teacher the **MY CONTENT** page. The resource is removed from the system
Result definition	A success message is displayed. And the resource is removed from the database.
Remarks	If we try to delete item, which is not present in the database.

Name	**Create Course**	**TCH 3.0-P**
Goal	Add a new course to the system.	
Participating actors	Teacher	
Precondition	The teacher is logged in.	
Test Type	Positive	

Event flow	Teacher is at the **HOME** page.
	    4. Teacher clicks on the "My Content" button.
	    5. Teacher is displayed the **MY CONTENT** page.
	    6. Teacher clicks the "Add resource" Button
	    7. Teacher is displayed an empty **EDIT COURSE** page containing the required fields.
	    8. Teacher fills in valid information about the new course.
	    9. Teacher clicks on submit button
	    10. The system shows the Teacher the **COURSE** page showing the newly added topic.
	
	A new course is created in the system.	

Output definition	Success message is displayed. New Course is created in the database	
Remarks	• Course already exists • Invalid data in fields (empty fields) (mismatch types in fields) etc	

Name	**Create Course**	**TCH 3.0-N**
Goal	Add a new course to the system.	
Participating actors	Teacher	
Precondition	The teacher is logged in.	
Test Type	Negative	

Event flow	Teacher is at the **HOME** page.
	    1. Teacher clicks on the "My Content" button.
	    2. Teacher is displayed the **MY CONTENT** page.
	    3. Teacher clicks the "Add resource" Button
	    4. Teacher is displayed an empty **EDIT COURSE** page containing the required fields.
	    5. Teacher fills in invalid information about the new course.
	    6. Teacher clicks on submit button	

	A new course is generated.	
Output definition	Error message is displayed and there is no effect on the database.	
Remarks		

Name	**Edit Course**	**TCH 3.1-P**
Goal	Edit an existing topics description.	
Participating actors	Teacher	
Precondition	The teacher is logged in.	
Test type	Positive	
Event flow	Teacher is at the **MY CONTENT** page. 1. Teacher clicks on the "Edit" button next to the chosen course. 2. Teacher is displayed an **EDIT COURSE** page containing the required fields, each filled in with information. 3. Teacher updates information about the course. 4. Teacher clicks on submit button 5. The system shows the Teacher the **COURSE** page showing the edited resource. The existing course is updated in the system	
Output definition	The course is updated in the database, and confirmation message is displayed.	

Name	**Edit Course**	**TCH 3.1-N**
Goal	Edit an existing topics description.	
Participating actors	Teacher	
Precondition	The teacher is logged in.	
Test type	Negative	
Event flow	Teacher is at the **MY CONTENT** page. 6. Teacher clicks on the "Edit" button next to the chosen course. 7. Teacher is displayed an **EDIT COURSE** page containing the required fields, each filled in with information. 8. Teacher updates invalid information about the course (Same name to another course). 9. Teacher clicks on submit button. Error message is generated by the system.	
Output definition	An error message is displayed to the user, no records are updated in the database.	

Name	**Remove Course**	**TCH 3.2-P**
Goal	Remove a course from the system	

Participating actors	Teacher
Precondition	The course contains no resources
Test Type	Positive
Event flow	Teacher is at the **MY CONTENT** page. 1. Teacher clicks on the "Remove" button next to the chosen course. 2. Teacher is displayed dialog containing a confirmation question about removing the course. 3. Teacher clicks on confirm button 4. The system shows the Teacher the **MY CONTENT** page. The course is removed from the system
Output definition	The course is removed from the database, and confirmation message is displayed.
Remarks	

Name	**Create Exercise**	**TCH 6.0**
Goal	Create a new exercise	
Participating actors	Teacher	
Precondition	Teacher has an existing course. And teacher is logged in	
Type	Positive	
Event flow	Teacher is at the **MY CONTENT** page. 1. Teacher selects a course. 2. Teacher is shown the **COURSE** page 3. Teacher clicks on the create exercise button 4. Teacher is displayed a **CREATE NEW EXERCISE** page form, containing name and descriptions fields. 5. Teacher fills in required valid information. 6. Using AJAX, teacher can add multiple choice questions to the exercise. 7. When finished, Teacher clicks on confirm button 8. The system shows the Teacher the **COURSE** page.	
Results	The exercise is saved in the system.	
Remarks		

Name	**Create Exercise**	**TCH 6.0**
Goal	Create a new exercise	
Participating actors	Teacher	
Precondition	Teacher has an existing course. And teacher is logged in	
Type	Negative	
Event flow	Teacher is at the **MY CONTENT** page. 1. Teacher selects a course.	

	2. Teacher is shown the **COURSE** page 3. Teacher clicks on the create exercise button 4. Teacher is displayed a **CREATE NEW EXERCISE** page form, containing name and descriptions fields. 5. Teacher fills in required in valid information. 6. Teacher add multiple choice questions to the exercise. 7. When finished, Teacher clicks on confirm button 8. The system shows the Teacher a proper error message.
Results	The error message is displayed to the user. No record are inserted into the system.
Remarks	

Name	**Add Question**	**TCH 6.1**
Goal	Create a new question for a exercise	
Participating actors	Teacher	
Precondition		
Type	Positive	
Event flow	Teacher is at the **CREATE NEW EXERCISE** page. 1. Teacher clicks on add questions button 2. Teacher is shown the **add question form,** using AJAX 3. Teacher fills in question and possible answers. 4. Teacher selects good answer. 5. Teacher clicks on add question button. 6. The form is submitted using AJAX 7. The question is shown on the exercise page using AJAX	
Results	The question is saved in the system.	
Remarks	• Invalid entry into the fields	

Name	**Edit Exercise**	**TCH 7.0**
Goal	Create a new exercise	
Participating actors	Teacher	
Precondition	Teacher has an existing course.	
Type	Positive	
Event flow	Teacher is at the **MY CONTENT** page. 1. Teacher selects the exercise. 2. Teacher is shown the **EXERCISE** page, the description is a form. 3. Teacher modifies the content of the form. 4. Teacher submits the form. 5. Teacher is shown the **EXERCISE** page	

Results	The updated exercise is saved in the system.
Remarks	• Check for invalid entry into the fields.

Name	**Edit Question**	**TCH 7.1**
Goal	Modify the content of a question	
Participating actors	Teacher	
Precondition	An exercise contains questions	
Type	Positive	

Event flow	Teacher is at the **MY CONTENT** page.
	  1. Teacher selects the exercise.
	  2. Teacher is shown the **EXERCISE** page, each question is a form.
	  3. Teacher modifies the content of the form.
	  4. Teacher submits the form.
	  5. Teacher is shown the **EXERCISE** page	

Results	The updated question is saved in the system.	
Remarks	• Check for invalid entry.	

Name	**Remove Question**	TCH 7.2
Goal	Remove a question from an exercise.	
Participating actors	Teacher	
Precondition	An exercise contains questions	
Type	Positive	
Event flow	Teacher is at the **MY CONTENT** page. 1. Teacher selects the exercise. 2. Teacher is shown the **EXERCISE** page 3. Teacher clicks on the remove button next to a question. 4. A dialog appears, questioning if the question really needs to be removed. 5. Teacher answers yes. 6. Teacher is shown the **EXERCISE** page, now without the question.	
Results	The question is removed from the system.	
Remarks	• Check for already deleted question, using the back button of the browser.	

Name	**Remove Exercise**	TCH 8.0
Goal	Remove an exercise.	
Participating actors	Teacher	
Precondition	An exercise exists.	
Type	Positive	
Event flow	Teacher is at the **MY CONTENT** page. 1. Teacher clicks on the remove button, next to an exercise. 2. A dialog appears, questioning if the exercise really needs to be removed. 3. Teacher answers yes. 4. Teacher is shown the **MY CONTENT** page, now without the exercise.	
Results	The exercise is removed from the system, along with it's questions.	
Remarks	• Check for already deleted exercise, using the back button of the browser.	

9.2   Test plan

Test team execute the test cases continuously during the projects based on the development iterations. We carry out the test plan in three different ways.

9.2.1 Watir Framework:

Almost 80% of the test cases are written in watir testing framework to be executed automatically. The remaining 20% of the test cases are based on the data that is dynamically generated from the database. It was hard and time consuming to implement. The code for the automated testing is provided with the project code. For each requirement there are two automated test cases (one positive and one negative) as listed above.

9.2.2 Unit Testing

Unit test are written to assure the working of the data layer and model layer. All unit tests are within the TaraxacumTests subproject. For each service there is a Test class, which contains one Test Method for each service method offered by the corresponding service.

To run the tests it is enough to click on the "run tests" button within Visual Studio. To test just a smaller part of the project, test methods can be sorted into test lists.

9.2.3 Manual Testing

20% of the test cases that were not implemented through the automated test case are tested manually. Moreover test for functionalities like validations, browsing, links & other functionality that are supporting the main requirements are conducted manually.

10. Responsibilities

10.1 Developers

The development team is expected to fix all the discovered bugs during the testing and development phases of the project. Project management can decide to reject bugs if they are interfering with the project planning.

11. Risks and contingencies

The major risk is to test all the implemented functionalities at a good time margin so the developers have enough time to fix the bugs and errors before it gets late. But in our case we have/had the whole testing team which is continuously testing updated and new implemented requirements. In case of any bug/error, it is directly reported to development team to fix it immediately. This is continuous process in all the iterations through out Taraxacum development lifecycle.

12. Approvals

Name	Title	Date yyyy-mm-dd	Signature