

AcceptLink

Treinamento

Visão Geral

- -Introdução
 - *Desenvolvimento de Aplicações Web
 - *TDD
 - *ATDD
- -AcceptLink Objetivos
 - *Abordagem em ATDD
 - *Modelagem de páginas
 - *Especificação de Testes de Aceitação
 - *Gerar Testes para Awtaf
- -Como Usar
 - *Tutorial

Introdução – Desenvolvimento Web

- Aplicações web tornaram-se parte essencial de muitas estratégias empresariais;
- Ambiente altamente competitivo, sujeito a constantes mudanças de regras de negócio;
- Algumas atividades, como os testes, podem não ser tratadas corretamente;

Introdução – Processo Ágil

- Entregas incrementais, constantes feedbacks de clientes e requisitos de usuário altamente personalizados caracterizam o processo de engenharia *web*;
- Processo de desenvolvimento ágil de software mostra-se eficaz para o desenvolvimento desse tipo de aplicação;
- O Processo ágil se baseia em curtos ciclos de desenvolvimento iterativo e incremental dos quais o cliente deve participar ativamente do desenvolvimento.

Introdução – TDD e ATDD

- Um dos fundamentos de metodologias ágeis é a prática do TDD (*Test-Driven Development*);
- Pode não atender aos requisitos de usuário adequadamente, visto que testes unitários são escritos pelos desenvolvedores;
- Assim, neste contexto, surge a estratégia de Desenvolvimento Dirigido à Testes de Aceitação (ATDD - *Acceptance Test-Driven Development*), no qual testes de aceitação desenvolvidos e validados pelo cliente passam a orientar o processo de desenvolvimento de software.

AcceptLink - Objetivos

- Implementar uma abordagem ATDD de alto nível em aplicações Web.
- A aplicação Web é descrita como um conjunto de páginas.
- Cada página possui componentes como como um título, campos de texto, formulários, *links*, botões, entre outros.
- As páginas se relacionam com as outras através de *links*.

AcceptLink - Objetivos

- Implementar uma abordagem ATDD de alto nível em aplicações Web.
- A aplicação Web é descrita como um conjunto de páginas.
- Cada página possui componentes como como um título, campos de texto, formulários, *links*, botões, entre outros.
- As páginas se relacionam com as outras através de *links*.

AcceptLink - Objetivos

- A aplicação *web* será modelada de modo que cada página web, seus componentes internos e relações (*links*) sejam modelados pelo usuário.
- A aplicação sob teste será descrita automaticamente por um grafo no qual, cada nó representa uma página, enquanto os *links* são representados pelas arestas.
- A partir do grafo serão gerados caminhos de teste, estes caminhos darão origem a casos de teste. Isto será feito antes da implementação da aplicação.

Tutorial AcceptLink

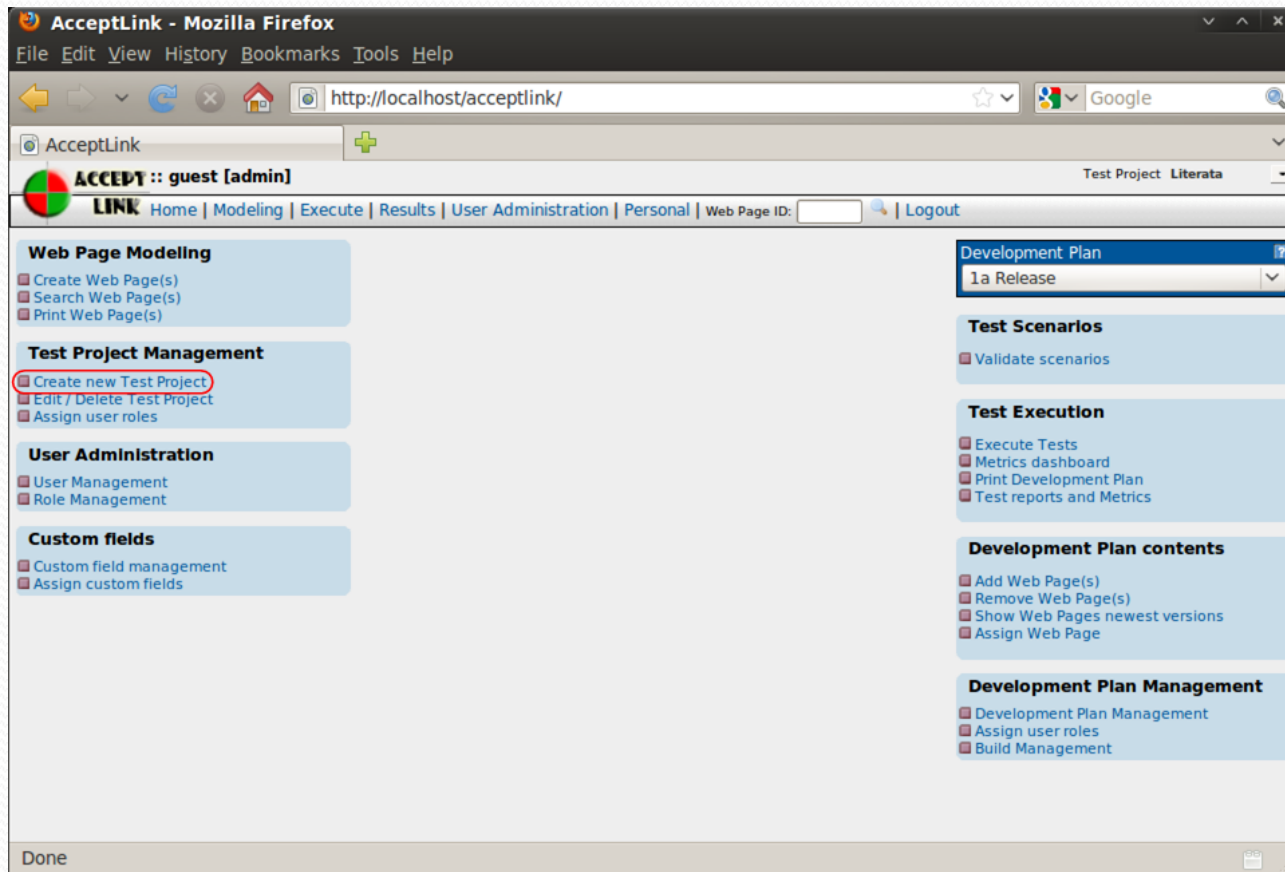
- Criar um Projeto de Testes
- Modelar as Estórias de Uso
- Modelar Páginas Web e seus componentes
- Criando um plano de desenvolvimento
- Preparar e validar Cenários de Teste

Criar um Projeto de Testes

- A primeira coisa a se pensar é criar um projeto de testes que irá conter a aplicação web e todos os seus componentes como modelos de páginas, Estórias de Usuário, cenários de testes, especificação de testes geral (GTS), e por aí em diante...

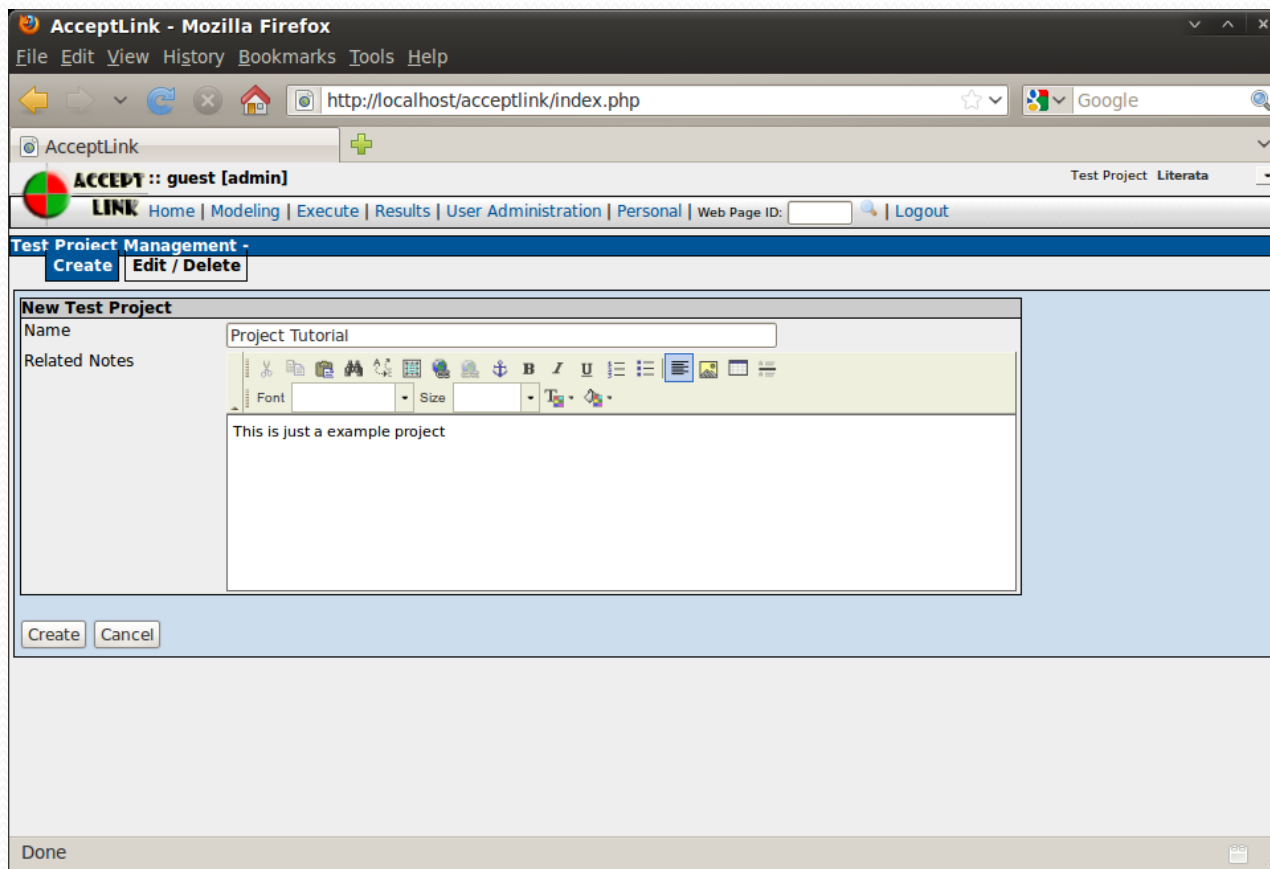
Criar um Projeto de Testes

- 1.1 Após realizar o login na aplicação, clique em “**Create new Test Project**”, opção marcada pelo círculo vermelho na figura abaixo:



Criar um Projeto de Testes

1.2 Preencha os campos **name of the project** (obrigatório) e **description** (opcional). Para finalizar esse passo, você deve clicar no botão **Create** a figura abaixo apresenta essa execução.



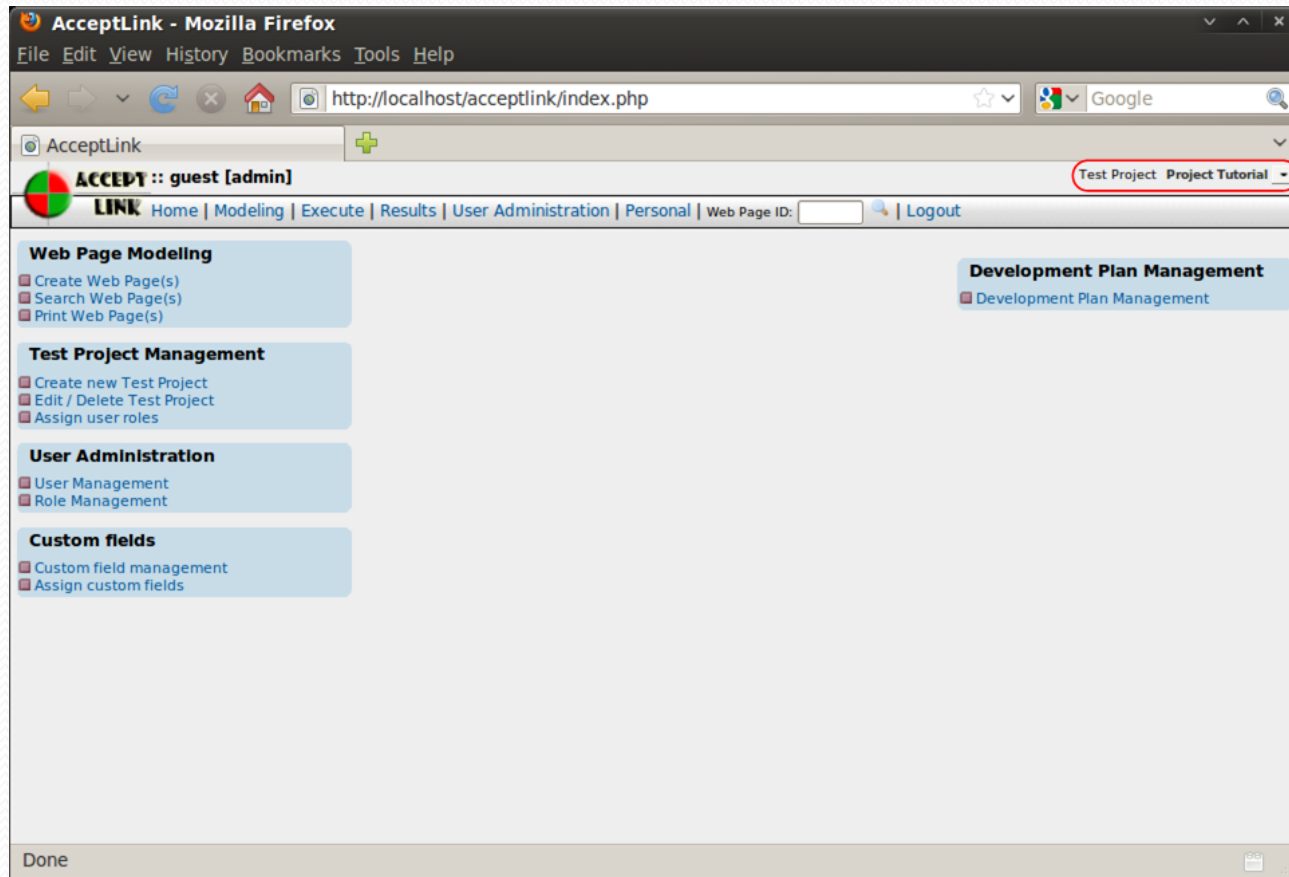
The screenshot shows a web browser window titled "AcceptLink - Mozilla Firefox" with the address bar displaying "http://localhost/acceptlink/index.php". The browser's address bar and search bar are visible. The application's header includes the "ACCEPT :: guest [admin]" status and a navigation menu with links: "LINK", "Home", "Modeling", "Execute", "Results", "User Administration", "Personal", "Web Page ID:", and "Logout". Below the header, a "Test Project Management" section contains "Create" and "Edit / Delete" buttons. The main content area is titled "New Test Project" and features a form with the following elements:

- A "Name" field containing the text "Project Tutorial".
- A "Related Notes" section with a rich text editor toolbar (including icons for bold, italic, underline, list, link, etc.) and a text area containing the text "This is just a example project".
- At the bottom of the form, there are "Create" and "Cancel" buttons.

The status bar at the bottom of the browser window displays the word "Done".

Criar um Projeto de Testes

1.3 Para começar a modelar páginas web, escolha o projeto para começar a trabalhar. Escolha o projeto clicando no combo box no canto superior direito da tela (circulado em vermelho na figura abaixo):

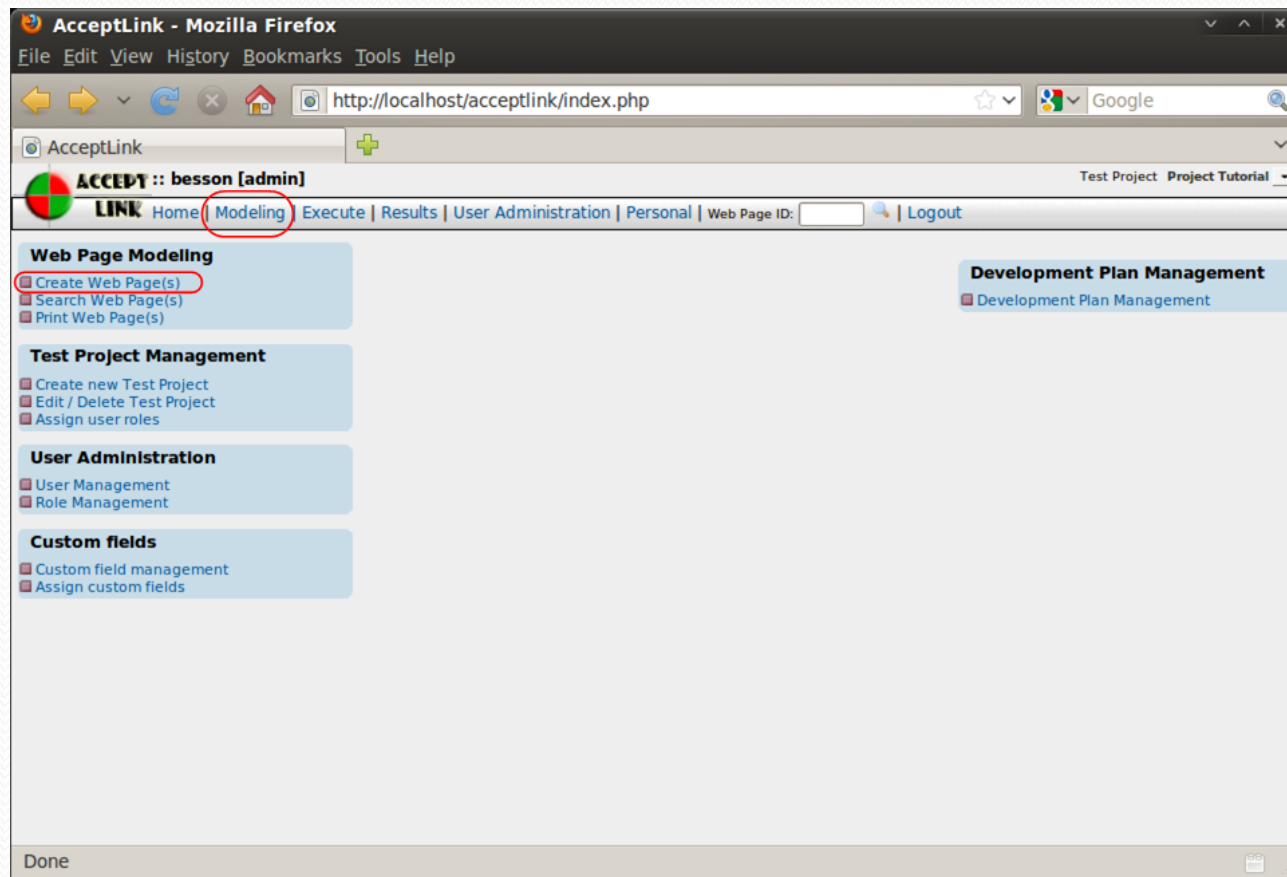


Modelar Estórias de Usuário

- Em um processo interativo de desenvolvimento de software, aplicações web são desenvolvidas incrementalmente; no nosso caso, esses incrementos são representados por meio de estórias de usuário.
- Para ilustrar esse processo, vamos considerar o exemplo de uma livraria virtual que provê entre outras funções a função de busca de livros. Para cada palavra-chave buscada, a aplicação deve consultar o banco de dados.
- Se a busca for bem sucedida, uma página web contendo os livros cujos títulos correspondam com as palavras-chave são apresentados; do contrário, uma página web com um aviso é mostrada, visto que o resultado da busca é vazio.

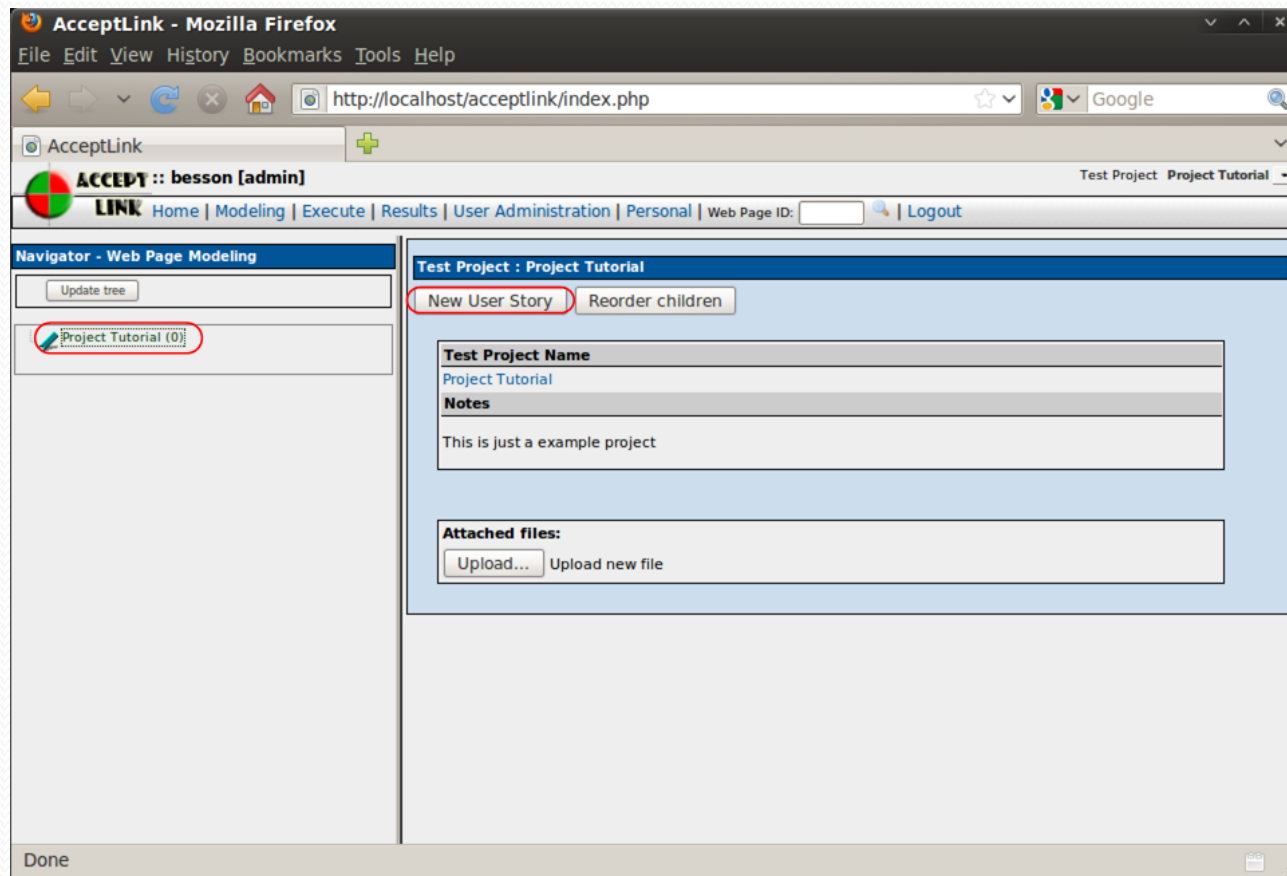
Modelar Estórias de Usuário

2.1 Criar uma Estória de Usuário clicando em **Modeling** ou **Create Web Page(s)**.



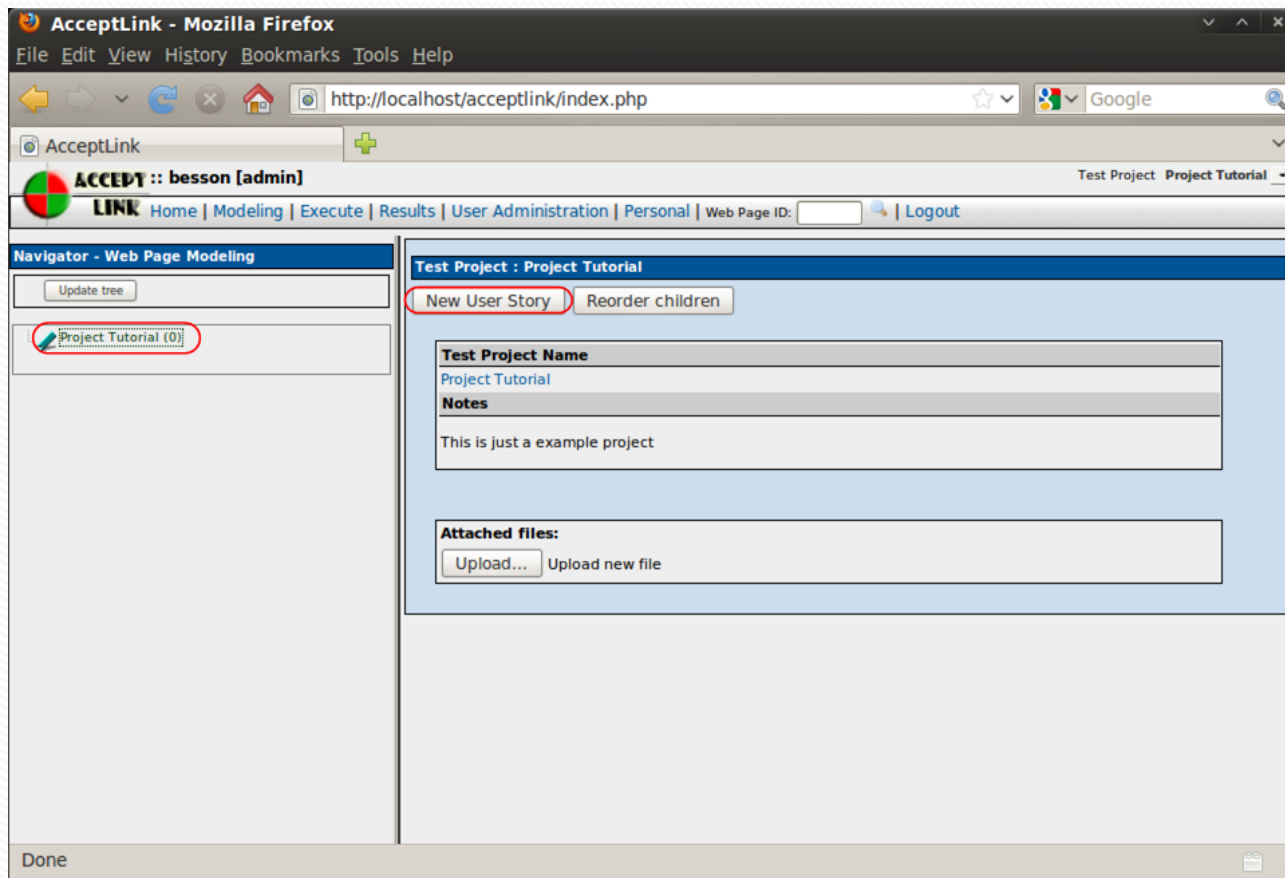
Modelar Estórias de Usuário

2.2 Selecione o projeto – em nosso caso, **Project Tutorial** – no lado esquerdo da tela. Depois de selecionar **Project Tutorial** e clicar no nome do projeto (veja a caixa circulada em vermelho no lado esquerdo da tela) a seguinte tela será apresentada.



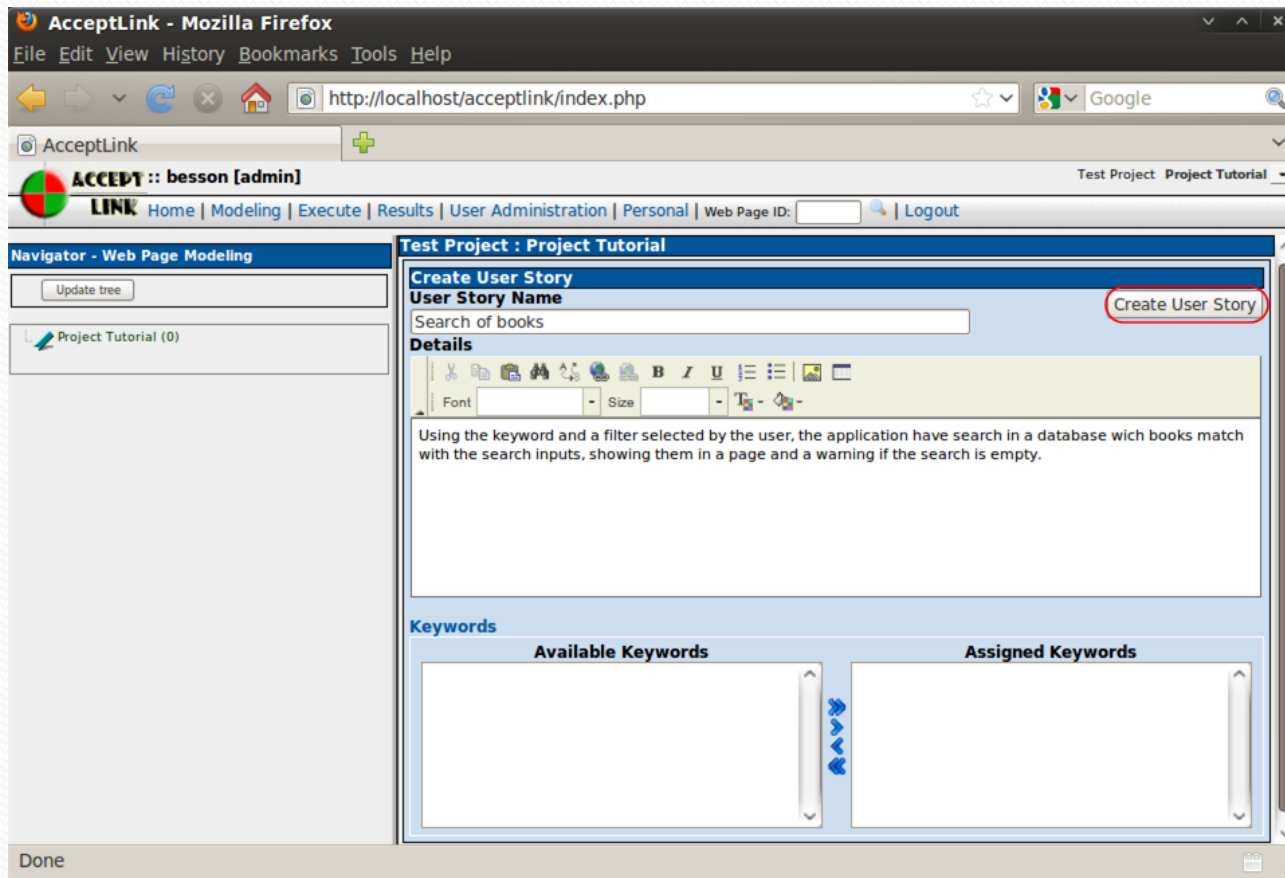
Modelar Estórias de Usuário

- A modelagem de Estórias de Usuário começa depois que clicamos no botão **New User Story** (veja a caixa circulada em vermelho no centro da figura abaixo)



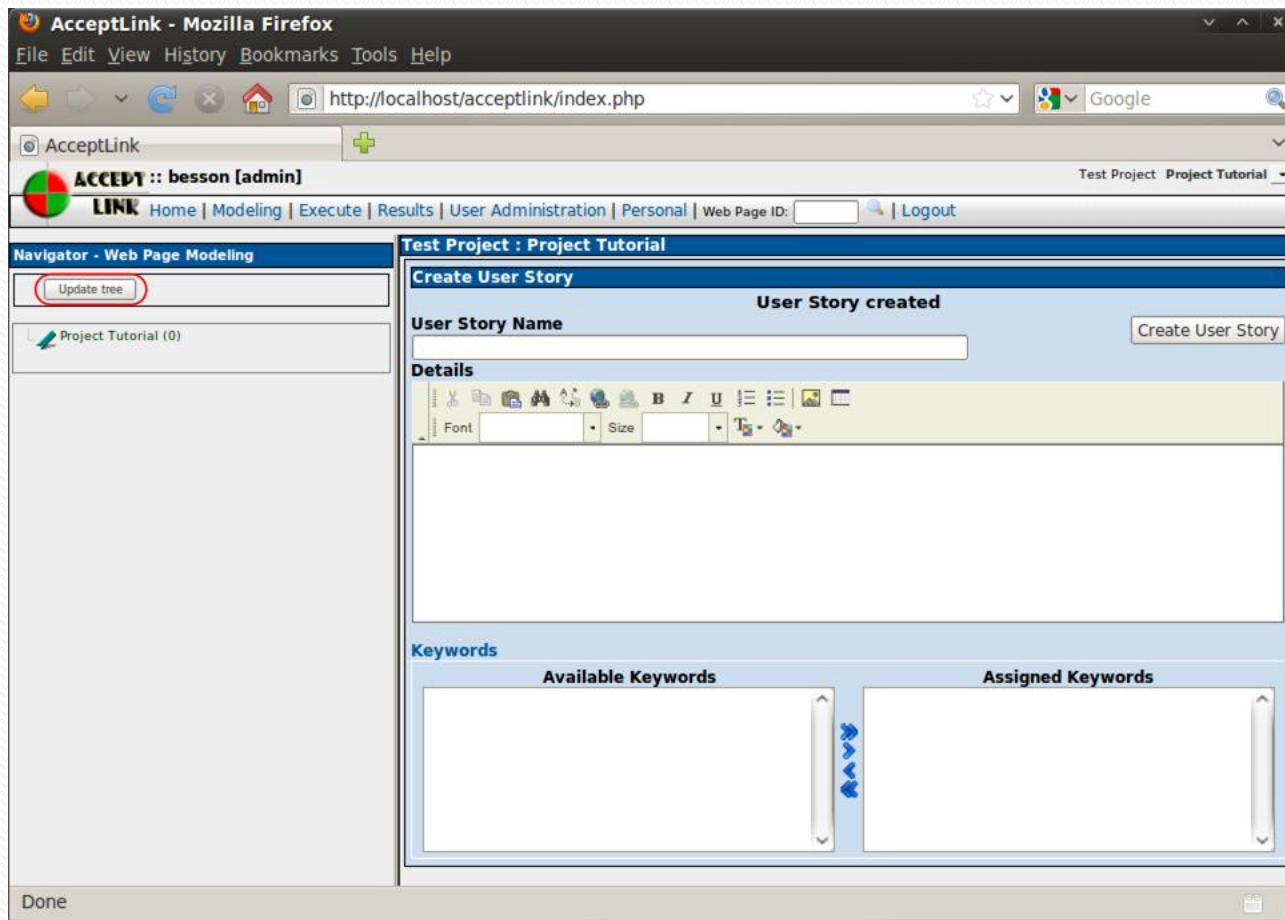
Modelar Estórias de Usuário

- Depois disso a seguinte tela será apresentada.

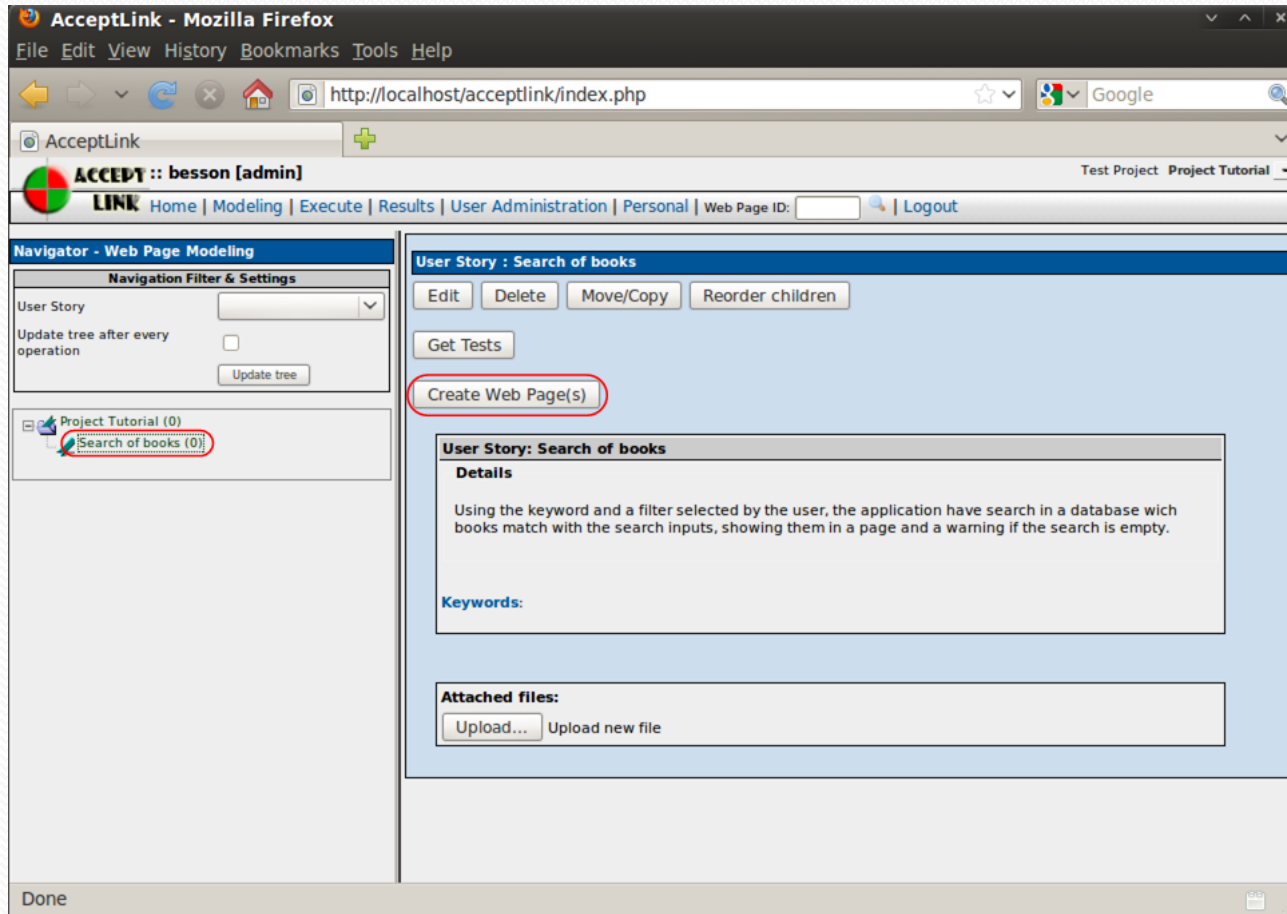


Modelar Estórias de Usuário

- 2.3 Para visualizar a estória de usuário criada clique na opção **Update**, e clique no nome da estória de usuário, como apresentado nas próximas duas figuras.



Modelar Estórias de Usuário

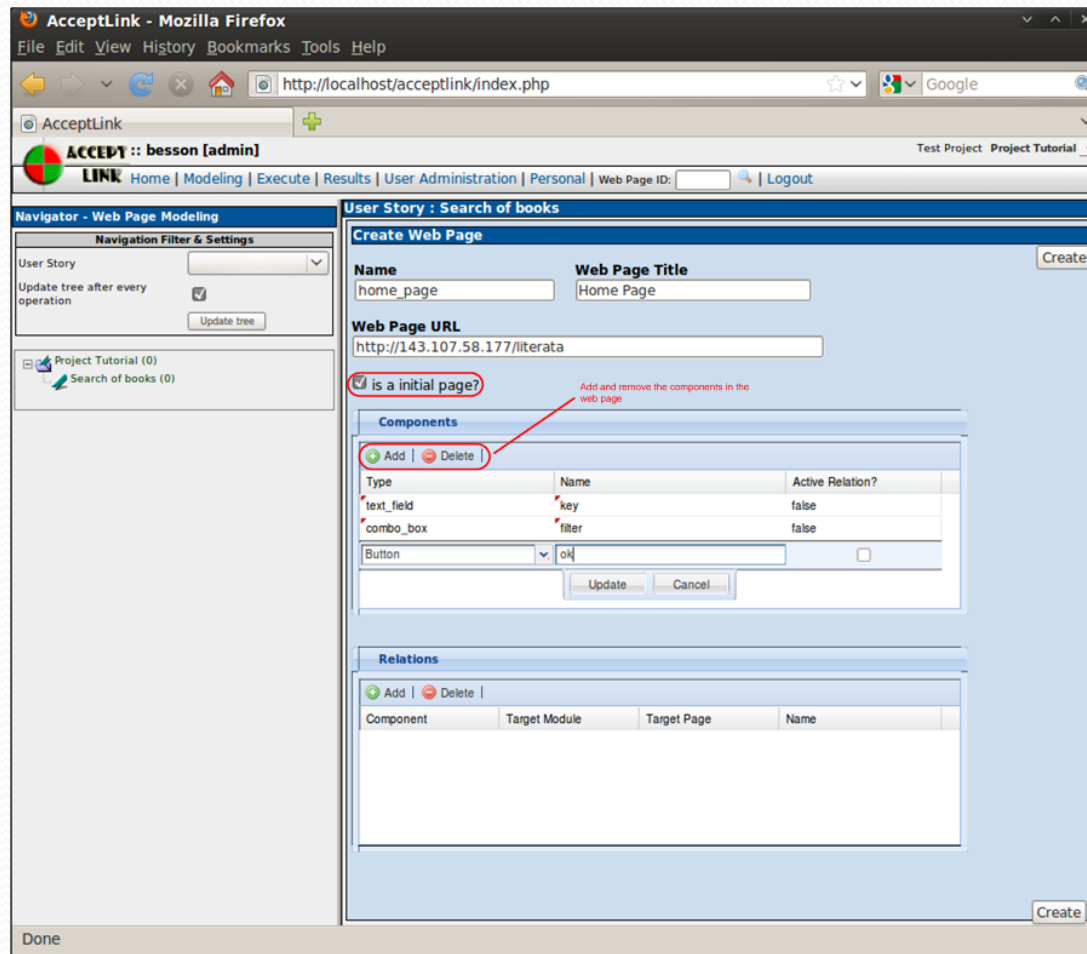


Modelar Páginas Web e seus componentes

- Agora as páginas pertencentes às Estórias de Usuário com todos seus componentes são modeladas.
- Seguindo nosso exemplo de "Search of books", nós vamos criar três páginas web: uma home page com campos para a busca, uma página web chamada quando a busca trouxer resultados, e finalmente uma página web com o aviso "No books were found" que é chamada quando nenhum resultado é retornado.

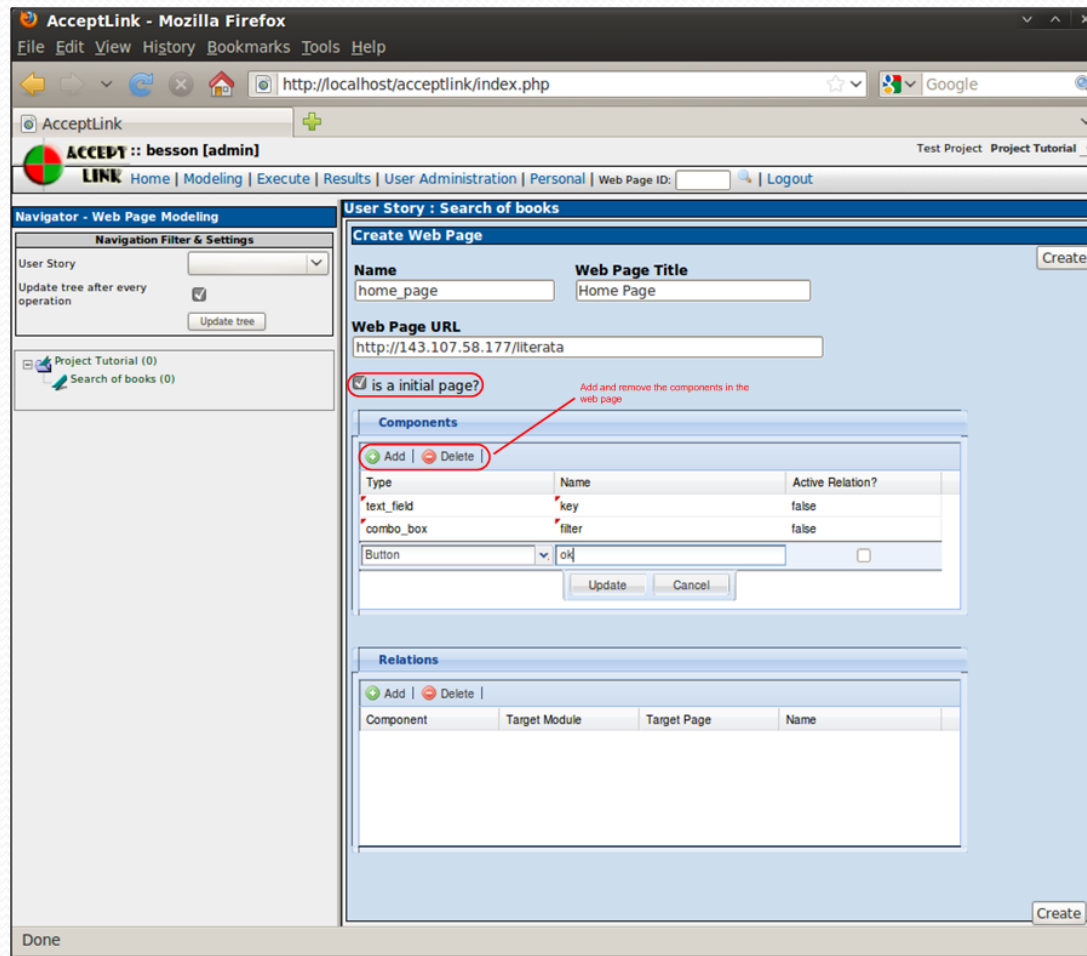
Modelar Páginas Web e seus componentes

- 3.1 Clique em **Create Web Page**. A figura abaixo mostra como criar a **home-page**. Como essa página será chamada quando a funcionalidade "Search of books" é requisitada, **home_page** é definida como a **initial Page**.



Modelar Páginas Web e seus componentes

- Note que essa página possui três componentes: um textfield que recebe a palavra-chave buscada, um combo Box que será definido como filtro (author, title, approach) e um botão que irá submeter a busca.



Modelar Páginas Web e seus componentes

- 3.2 As outras duas páginas planejadas no início são modeladas. Primeiramente nós vamos criar a página com resultados e depois a página com a mensagem de aviso.
- Ambas as páginas são muito simples já que elas contém apenas um componente body text. O uso deste componente significa que nós esperamos algo importante no corpo da página tal como uma sequência de palavras, uma frase, e outros.
- Em nosso caso nós esperamos os títulos dos livros ou uma página de aviso. Os valores esperados para todos os componentes são definidos na próxima seção desse tutorial. Por agora, nós apenas definiremos os componentes das páginas como apresentados nas próximas figuras:

Modelar Páginas Web e seus componentes

AcceptLink - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/acceptlink/index.php
AcceptLink AcceptLink Get Started
ACCEPT :: besson [admin] Test Project Project Tutorial
LINK Home | Modeling | Execute | Results | User Administration | Personal | Web Page ID: | Logout

Navigator - Web Page Modeling

Navigation Filter & Settings

User Story:
Update tree after every operation: ☒

Project Tutorial (2)
Search of books (2)
640: home_page
642: page_results

Create Web Page

Name **Web Page Title**

Web Page URL

☐ is a initial page?

Components

Type	Name	Active Relation?
body_text	results	false

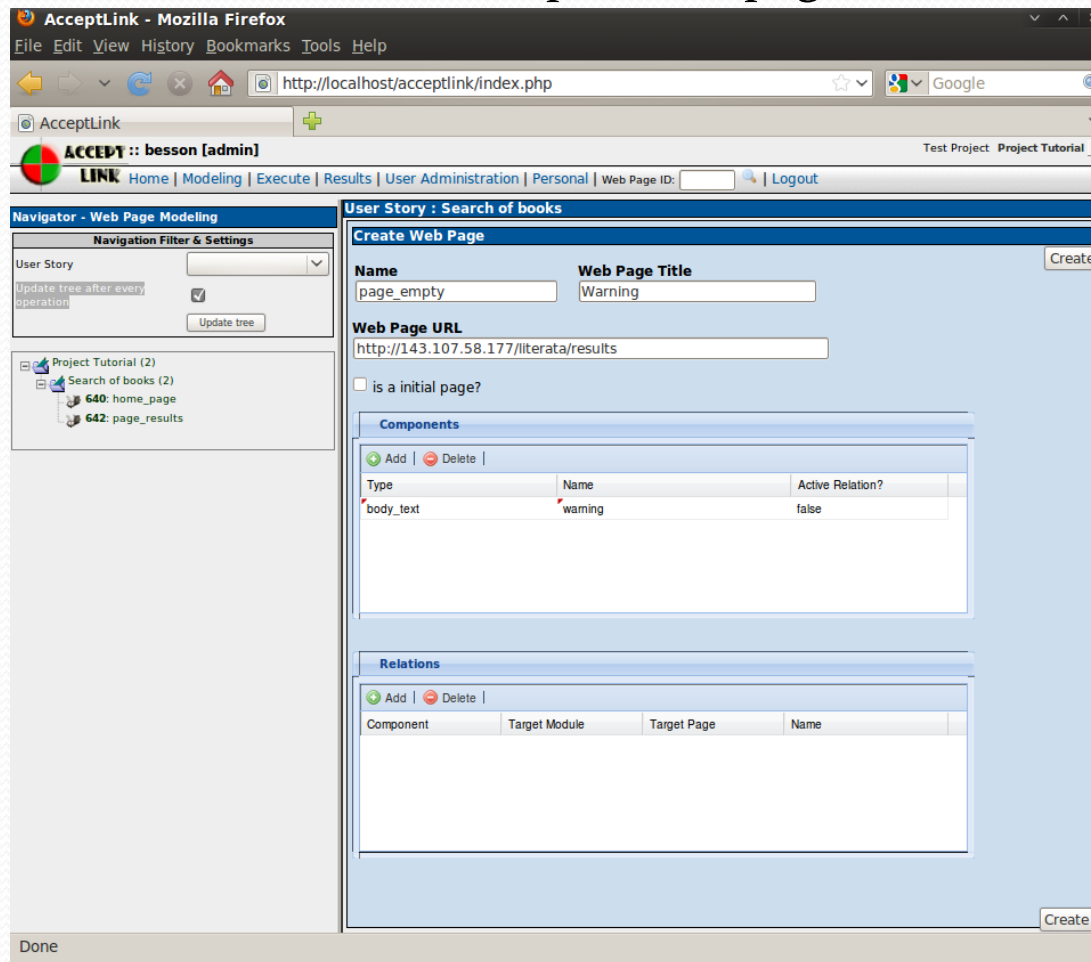
Relations

Component	Target Module	Target Page	Name
-----------	---------------	-------------	------

Done

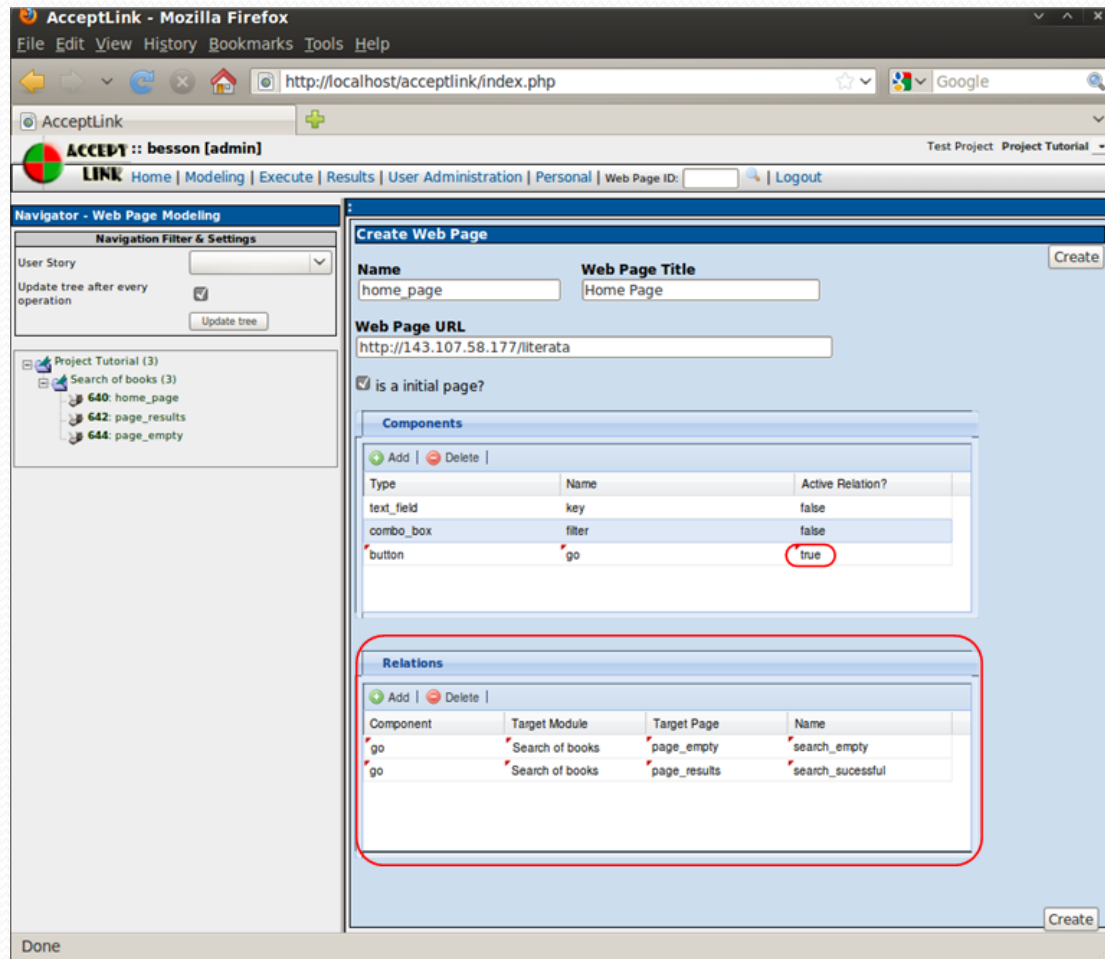
Modelar Páginas Web e seus componentes

- 3.3 Depois de criar uma página web, opção **Update Tree** precisa ser acessada para atualizar a árvore de páginas web no lado esquerdo da tela. Uma outra opção é deixar a opção **Update tree after every operation** marcada para atualizar automaticamente a árvore após cada página ser criada.



Modelar Páginas Web e seus componentes

- 3.4 Depois de modelar todas as páginas, as relações entre elas devem ser definidas. Em nosso caso, duas relações foram definidas: **search_sucessfull** e **search_empty**.



Modelar Páginas Web e seus componentes

- Ambas as relações iniciam na **home_page**, o componente **go** ativa as relações; isso significa que quando se clica em **go** uma destas duas páginas (**page_results** ou **page_empty**) é chamada.

AcceptLink - Mozilla Firefox
http://localhost/acceptlink/index.php

ACCEPT :: besson [admin]
LINK Home | Modeling | Execute | Results | User Administration | Personal | Web Page ID: | Logout

Navigator - Web Page Modeling

Navigation Filter & Settings

User Story:
Update tree after every operation: ☒
Update tree:

Project Tutorial (3)

- Search of books (3)
- 640: home_page
- 642: page_results
- 644: page_empty

Create Web Page

Name: home_page Web Page Title: Home Page Create

Web Page URL: http://143.107.58.177/literata

☒ is a initial page?

Components

Type	Name	Active Relation?
text_field	key	false
combo_box	filter	false
button	go	true

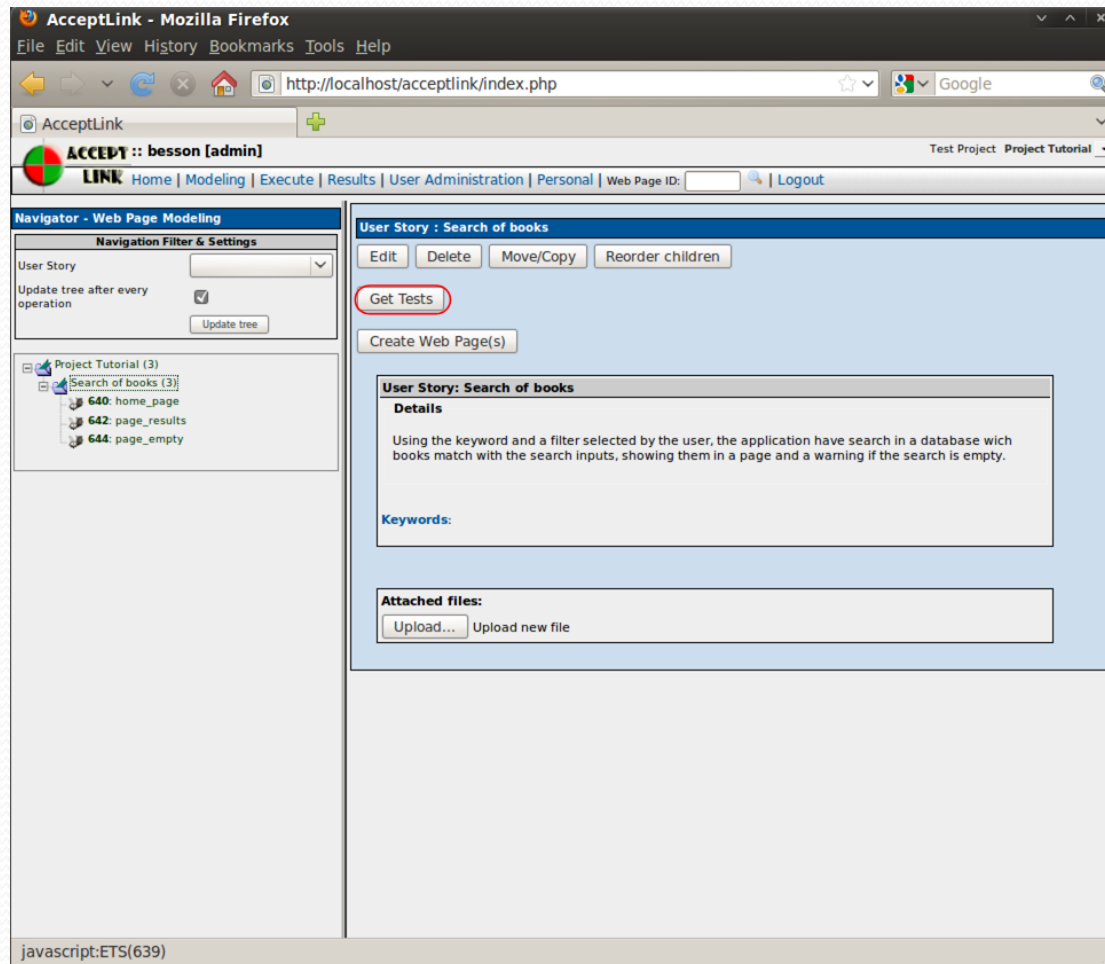
Relations

Component	Target Module	Target Page	Name
go	Search of books	page_empty	search_empty
go	Search of books	page_results	search_sucessful

Done

Modelar Páginas Web e seus componentes

- 3.5 Finalmente, os cenários de testes devem ser gerados para cada contexto de uso sob teste/desenvolvimento. Para fazer isso, clique no **contexto de uso** (e.g. Search of Books) e na opção **Get Tests** como mostrado na figura abaixo.

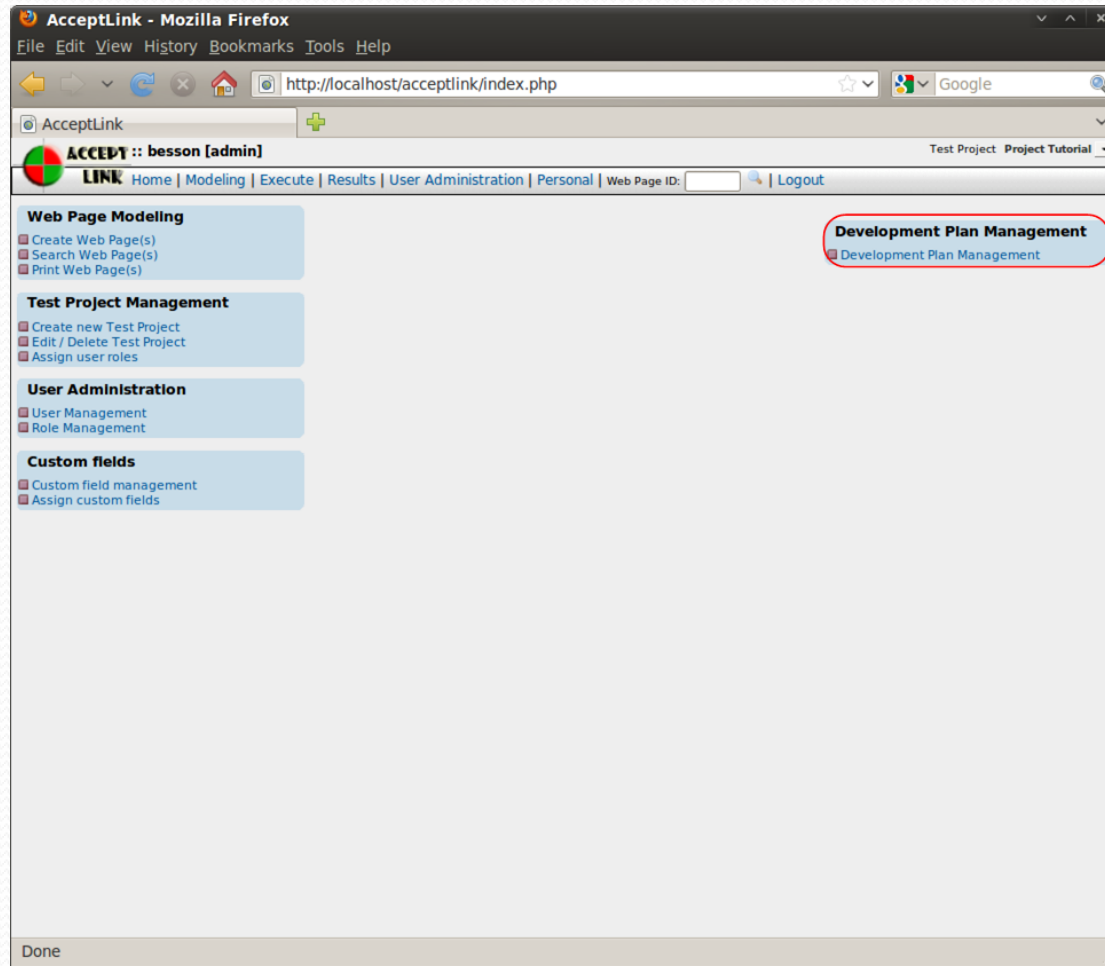


Criando um plano de desenvolvimento

- Uma vez que todos os cenários de testes foram gerados, um plano de desenvolvimento tem que ser criado para manter estes cenários.

Criando um plano de desenvolvimento

- 4.1 Vá para a home page da AcceptLink e escolha a opção **Development Plan Management** como apresentado na figura abaixo:



Criando um plano de desenvolvimento

- 4.2 Então, clique na opção **Create** e preencha os campos como mostrado na figura abaixo:

The screenshot shows a web browser window titled "AcceptLink - Mozilla Firefox" with the URL "http://localhost/acceptlink/index.php". The browser's address bar and search bar are visible. The page header includes the "ACCEPT :: besson [admin]" logo and navigation links: "Home", "Modeling", "Execute", "Results", "User Administration", "Personal", "Web Page ID:", and "Logout". The main content area is titled "Development Plan Management" and contains a form titled "Create a new Development Plan for Test Project Project Tutorial". The form has two main sections: "Name" and "Description". The "Name" field contains the text "First Release". The "Description" field contains the text "Example of development Plan". Below the form, there are "Create" and "Cancel" buttons. The status bar at the bottom of the browser window shows "Done".

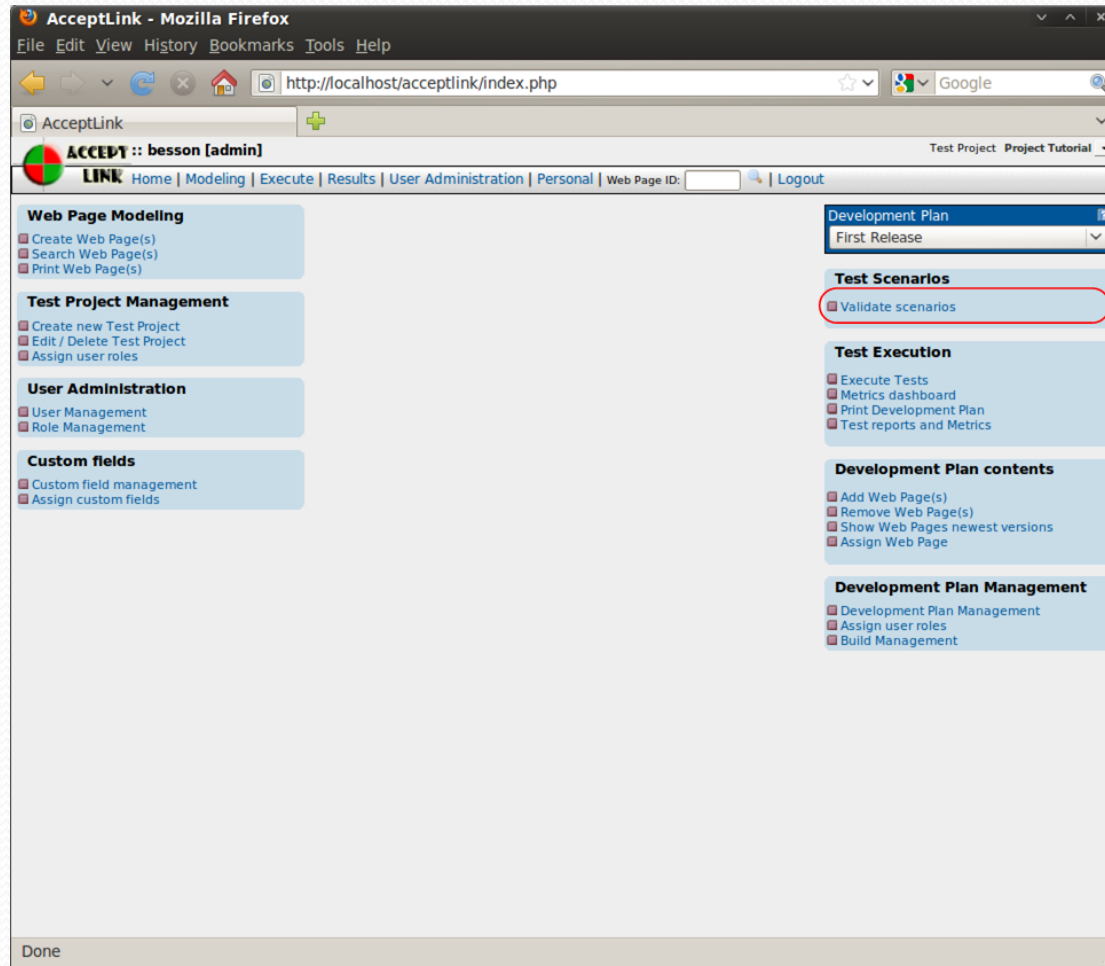
AcceptLink - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/acceptlink/index.php
AcceptLink
ACCEPT :: besson [admin] Test Project Project Tutorial
LINK Home | Modeling | Execute | Results | User Administration | Personal | Web Page ID: | Logout
Development Plan Management
Create a new Development Plan for Test Project Project Tutorial
Name First Release
Description
Font Size
Example of development Plan
Create Cancel
Done

Preparar e validar Cenários de Teste

- O ultimo passo do fluxo de trabalho da AcceptLink é validar o modelo de páginas web e preparar os cenários de teste com os clientes.
- Mais tarde, a Especificação Genérica de Testes “Generic Test Specification” (GTS) – um arquivo XML executável no AWTAF (Automated Web Test Application Framework) – é gerado.

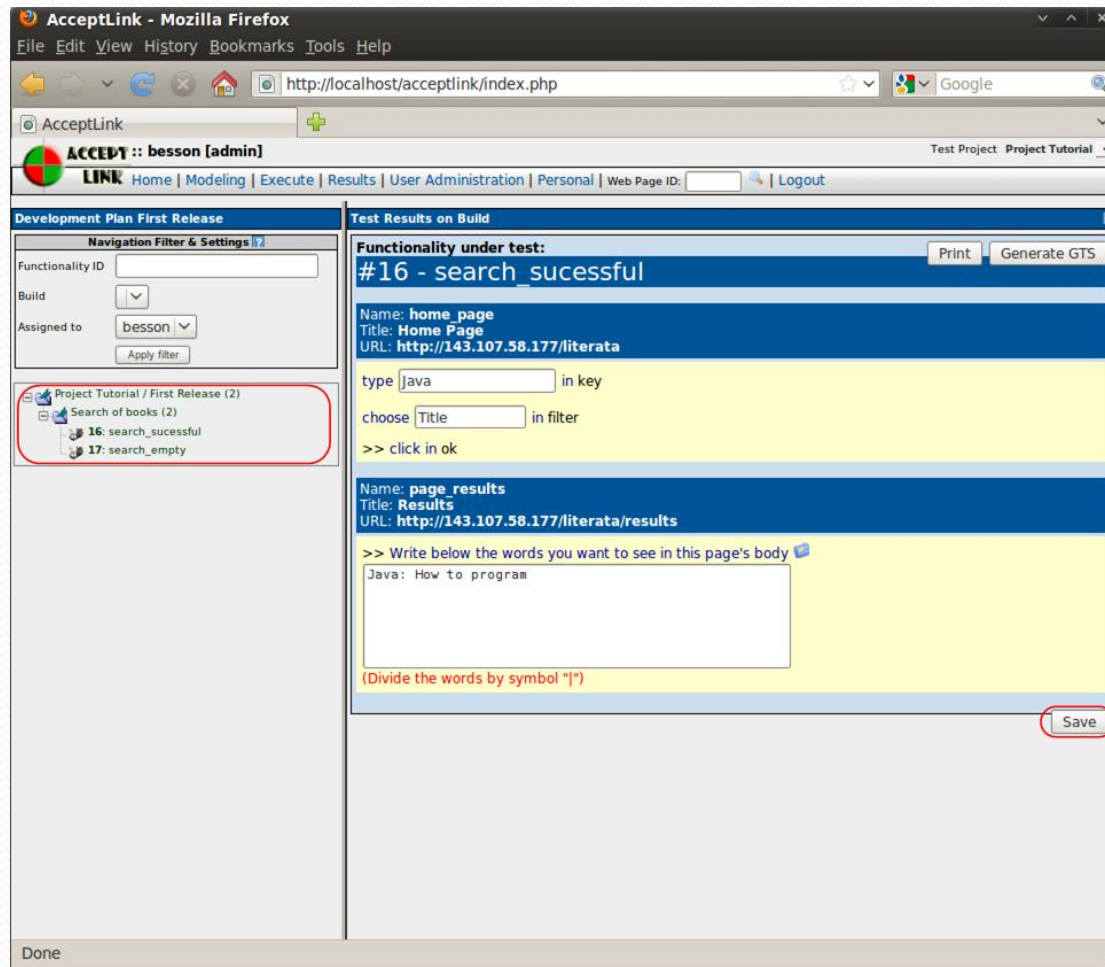
Preparar e validar Cenários de Teste

- 5.1 Vá para a home page da AcceptLink e escolha a opção **Validate Scenarios** como mostrado na figura abaixo:



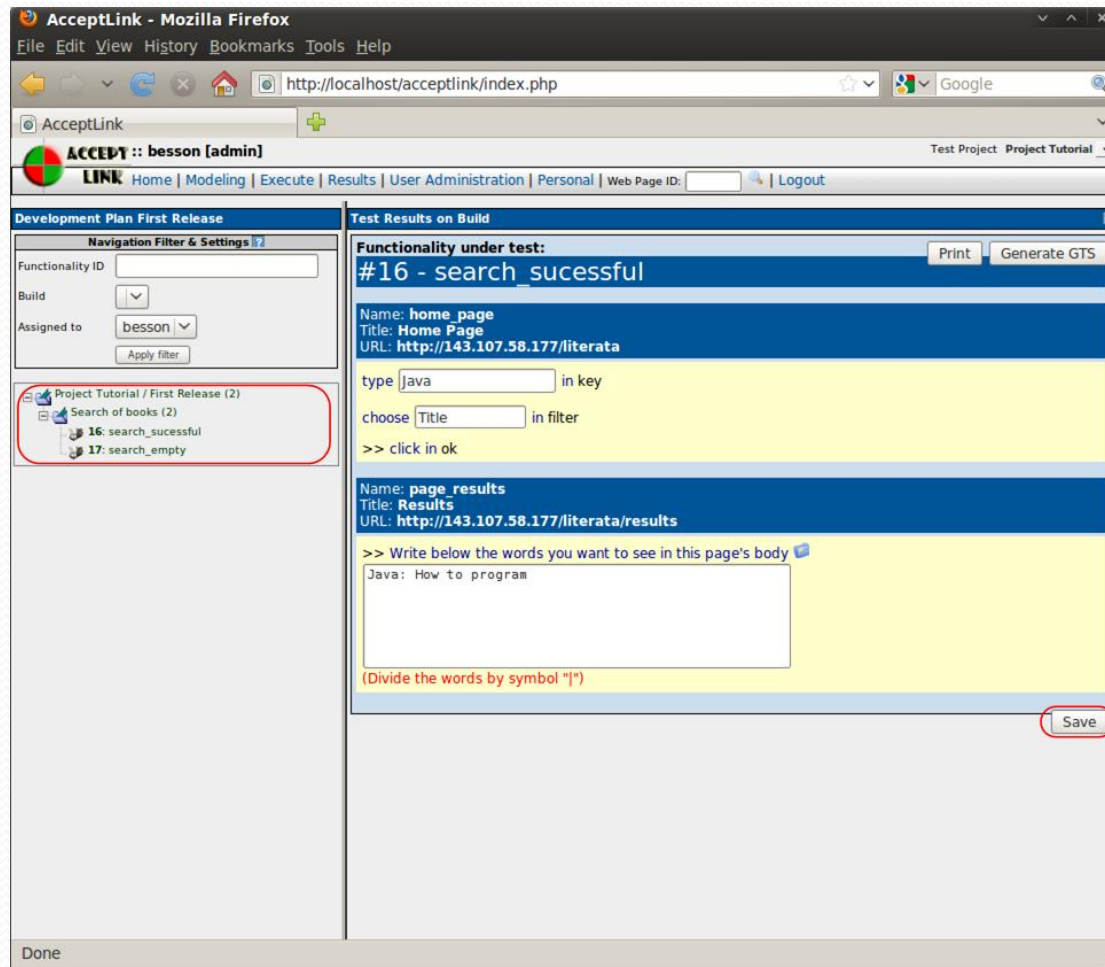
Preparar e validar Cenários de Teste

- 5.2 Todos os testes gerados serão apresentados no lado esquerdo da painel como mostrado na figura abaixo. No nosso caso, apenas dois cenários de testes foram gerados (**search_sucessful** e **search_empty**) porque nosso contexto de uso é bem simples.



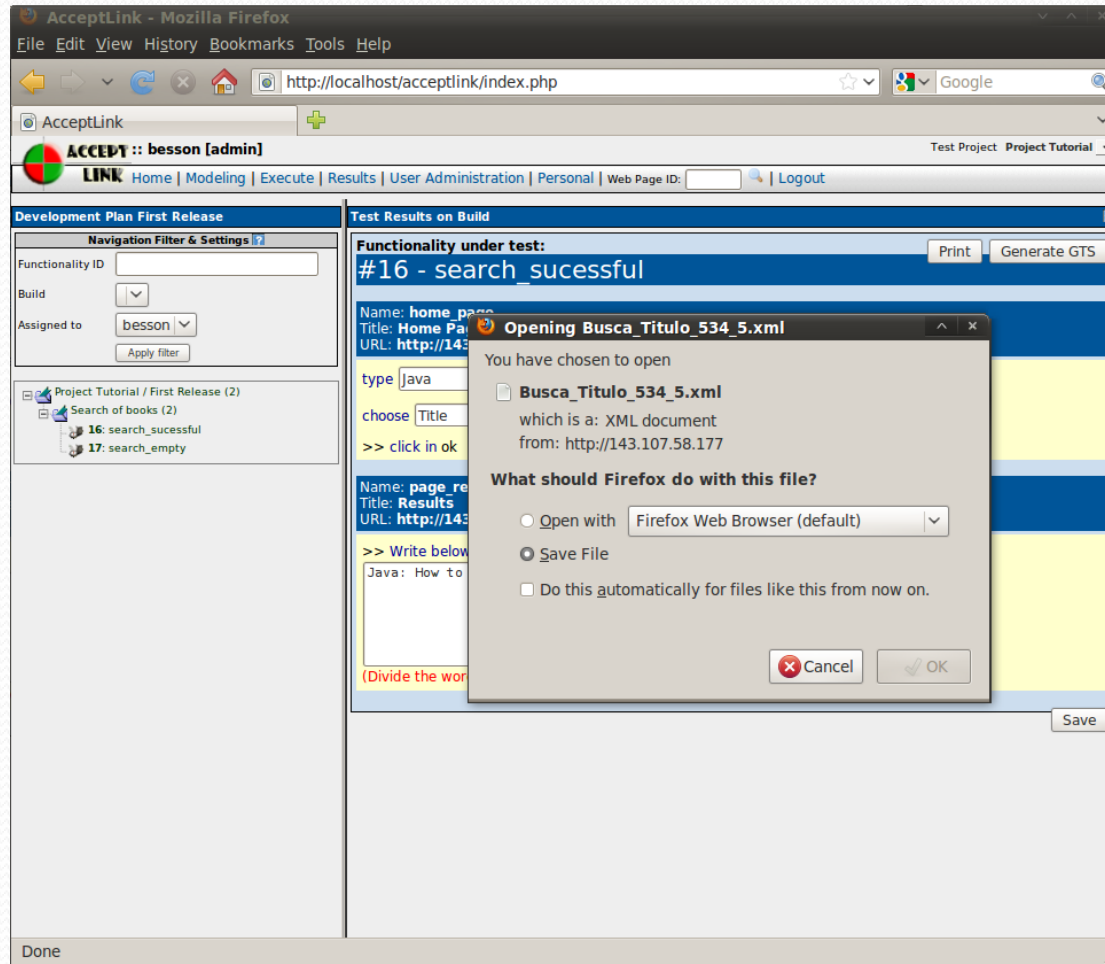
Preparar e validar Cenários de Teste

- Para validar e preencher os cenários o usuário precisa clicar em um dos cenários, preencher seus campos e salvar o trabalho. A figura abaixo se refere ao cenário **search_sucessful**.



Preparar e validar Cenários de Teste

- 5.3 Finalmente, clique na opção **Generate GTS** para criar o arquivo GTS que será utilizado na AWTAf.



Bibliografia

- BESSON, F. M. **Uma Ferramenta para Modelagem Ágil de Casos de Teste de Aceitação para Aplicações Web.** 2009 50 f. Monografia (Bacharelado em Sistemas de Informação) – Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, São Paulo, 2009.
- BESSON, F.; BEDER, D.; CHAIM, M. **Framework para execução automatizada de testes de aplicações web.** In: Experimental Software Engineering Latin American Workshop, 5. Salvador: Universidade Federal da Bahia, 2008.
- **Tutorial AcceptLink** disponível em: <
<http://143.107.58.177/acceptlink/tutorial/index.php> >