# Navigation/Grapical User Interface

## Overview

ClubUML is has been an ongoing graduate school project under the order of Northeastern University Graduate School of Engineering. The effort of the Spring 2014 team is to provide sustainment to the system by providing upgrades and modernization to ClubUML. ClubUML is comprised of several layers of components that build up the entire system. The Graphical User Interface (GUI) supports the functionality and visualization to the user. The Spring 2014 team identified and prioritized the tasks level of effort during the course of the semester.

## Development Approach

### Sprint Overview

The Spring 2014 Team followed an Agile Development approach to support the development of the Navigation/GUI Development for the semester. It is an iterative and incremental approach for development where requirements and solutions are evolved. Agile development promotes dynamic planning during development and the delivery of the software. As seen in the schedule, there are iterations and code deliveries after a level of effort. Each level of effort will consist of different functionality or a big milestone.

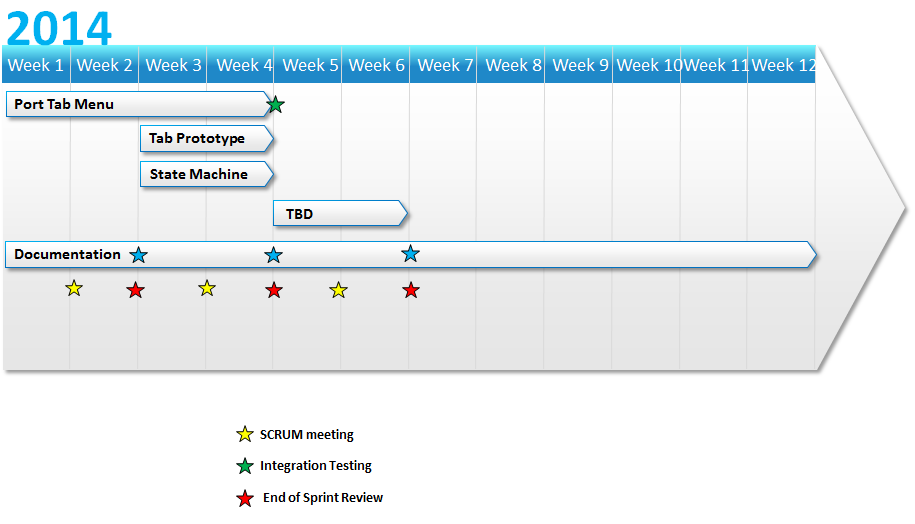


Figure : Agile Development

Each sprint lasts for a two week period and it broken up into 2 chunks with a scrum meeting halfway through the sprint. The GUI team members participated within each scheduled scrum meeting throughout the semester.

### Schedule

The GUI team has allocated the semester in a series of sprints to build and provide capability to the customer. Each sprint is designated for a specific capability to be developed, integrated and delivered. The team did an analysis of which capabilities should be lined up and implemented within each sprint. The following figure depicts the schedule for the entire semester.



A scrum meeting was a biweekly Tag up of the GUI team internally and also with the entire ClubUML team. The purpose of the scrum is to synchronize with all of the team members of each member’s current status. The GUI team members thoroughly communicated throughout the scrum, however it is beneficial for them to meet with the entire ClubUML team for an up to date status.

#### Sprint 1

Sprint 1 was contained of one major effort broken down into tasks. The goal of Sprint 1 was to migrate the existing Tab Navigation to the pages that did not exist the navigation to the user. The tasks included:

1. Identify Code to migrate

2. Identify Pages to be supported

3. Migrate Code

4. Test

5. Documentation

#### Task 1

The team identified the following code to be migrated:

<div id=*"myHeader"*>

<h1 id=*"banner"*>Class Diagrams</h1>

</div>

<div id=*"Tabs1"*>

<ul>

<li><a href=*"#tabs-1"*>Home</a></li>

<li><a href=*"ManageProject"*>Project</a></li>

<li><a href=*"ManageContext"*>Context</a></li>

<li><a href=*"ManagePolicy"*>Policy</a></li>

</ul>

</div>

In addition, the corresponding libraries also have to be carried over and be referenced.

#### Task 2

The team identified the following pages to be updated:

ClassMerge.JSP

Promote.JSP

Compare.JSP

These pages are built of JavaScript, CSS and HTML to provide the user interface to the user. The goal and implementation of the sprint is to migrate the existing Tab navigation to these pages.

These pages initially did not have any navigation which was confusing to the user as they weren’t able to leave a project.

#### Task 3

Task 3 is the effort to actually migrate and commit the code. Currently, the team migrated it over but have not committed anything yet.

#### Task 4

The integration testing is comprised of once committed, the developers test the build with the main repository.

#### Task 5

Documentation is a huge part of software development and provides current and future developers the ability to understand the system. The documentation for the sprint will be elaborated within this document.

### Sprint 2

#### **Task 1**

The team has identified the software GUI to be architected in a very inefficient way. The current implementation contains files and folders to be organized and designed not to a standard software implementation. For example, the JSP point to individual CSS files, compared to pointing to a single source. Also, CSS logic is embedded within HTML files and JSPs.

The task is for the team to research and investigate a new design and architecture for the application to be implemented for the GUI.

#### Task 2

The team has identified the possibility to migrate to new GUI look and feel for the navigation pane. This is to replace the current tab implementation. The team will look into the effort to replace and integrate with a new set of javascript libraries that are available.

#### Task 3

The team has identified pieces of code that are spread across the system, not modular at all. Logic to support the tab menu was pieced all over different files, compared to a standard framework. There is CSS logic for the tab in different JSP files which is not a practice for web development. This drove the Task 1 for Sprint 2.

### Sprint 3

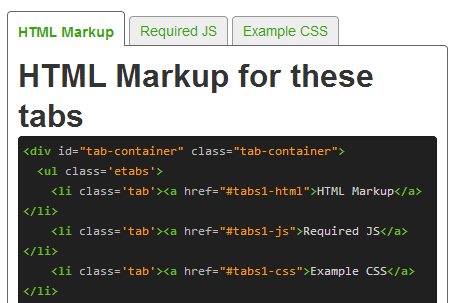
#### Task 1: Tab Modernization Research

The GUI team initiated the analysis to look into different modern tab navigation components. The purpose of providing a new tab view is to give the application a new look and feel to the user. instead of the team diving into creating a new tab navigation, the team did research fir existing implementations that could be reused and ported to clubuml in an easy low cost effort.

The Team looked into multiple different options that are both open source and licensed. The team found the following:

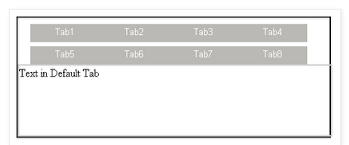
***Easy tabs*** (http://os.alfajango.com/easytabs/)

The easy tabs implementation provides the ability for a developer to apply a tab to multiple different CSS instances giving a different look and feel for multiple applications.



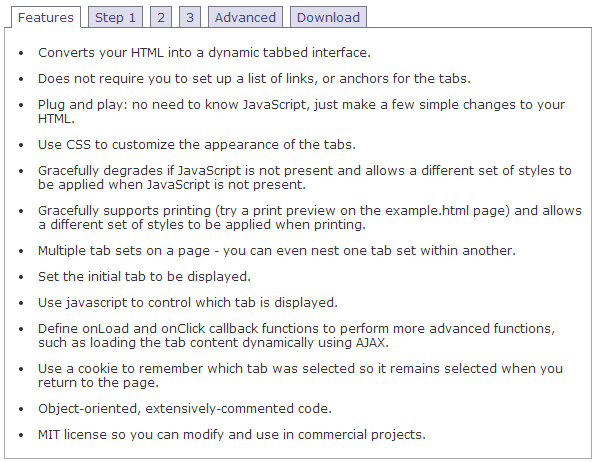
***Open Source Programmer*** (http://opensourceprogrammer.blogspot.com/2010/01/tab-menu-in-javascript.html?m=1)

The Open source programmer provides a library to create a simple tab menu within javascript with default actions and events to support the user.



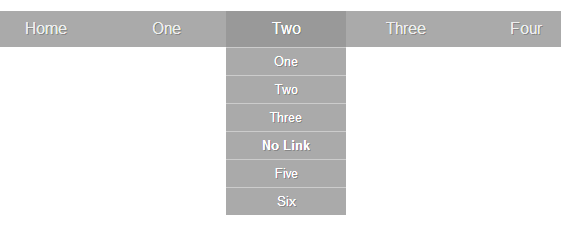
***Tabber*** (http://www.barelyfitz.com/projects/tabber/)

Tabber is an easy to use javascript tab library to be implemented within a web application. It automatically creates an HTML tab interface using the plug and play javascript.



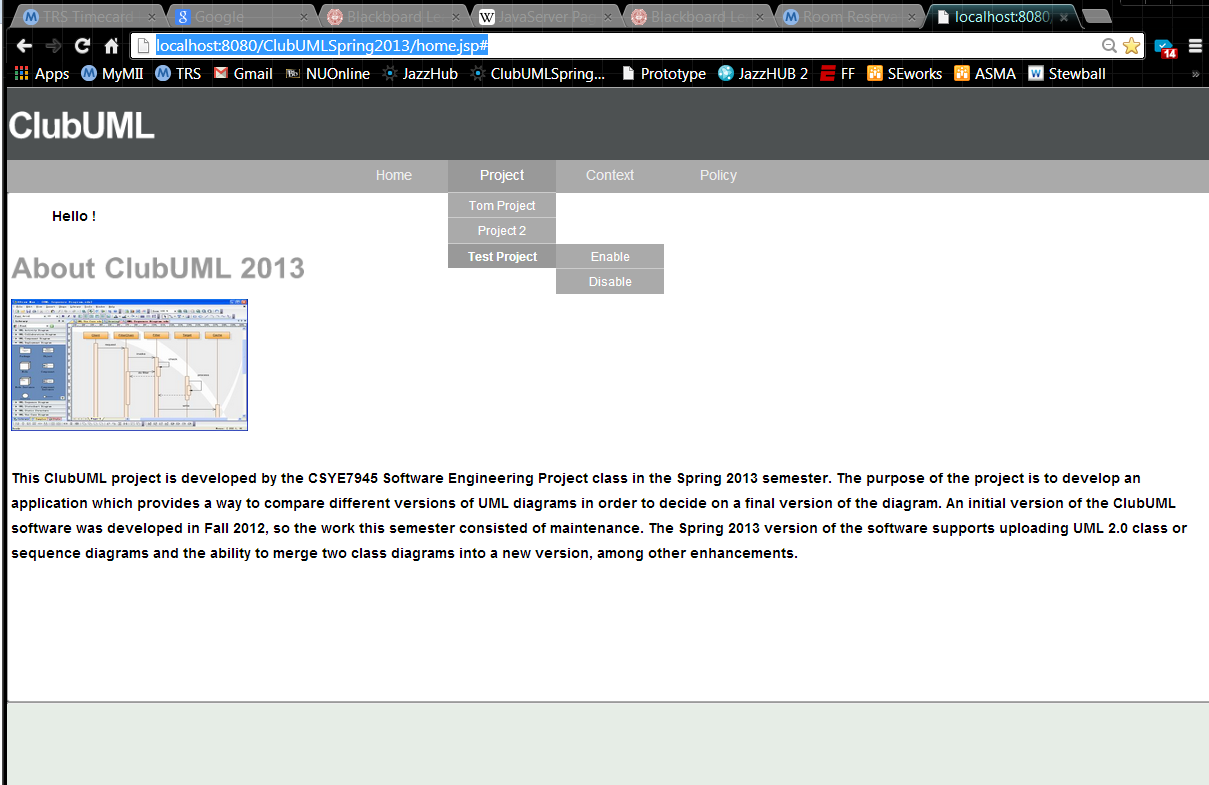
***TinyDropdown 2*** (http://sandbox.scriptiny.com/tinydropdown2/)

TinyDropdown provides a modern easy to use tabbing interface. It gives the user the ability to see an animated tab bar for easy access to multiple different GUI panels and implementations. It is easy for a developer to integrate and implement against their application.



#### Tab Modernization Prototype

The GUI team created a prototype to give the rest of the team a vision of a new implementation of a modern interface. The decision was made to utilize the TinyDropDown 2 library to use within the prototype. Changes were made to the home.jsp and also adding the set of the libraries used in referenced. A demonstration will be conducted to the team which there will be a decision be made for the next direction. A screen shot of the prototype is shown below.



#### GUI Modernization Research

#### The current implementation provides a messy and ugly GUI development. A couple of examples of poor development:

* CSS implementation embedded in JSP and JS
* Multiple instances of the same CSS file
* JSPs in WEB-INF directory, this directory is for web information and references
* No folder/file structure, files are scattered with no organization

The proposed new framework is as follows under the Web Content Folder:

1. Global Servlet

- JSPs

2. Home

- JSPs/HTML Files

3. Project

- JSPs/HTML Files

4. Policy

- JSPs/HTML Files

5. Context

- JSPs/HTML Files

6. JS Libraries

6.1 Global

- JS Files

6.2 Home

- JS Files

6.3 Project

- JS Files

6.4 Context

- JS Files

7. CSS Libraries

7.1 Global

- CSS Files

7.2 Home

- CSS Files

7.3 Project

- CSS Files

7.4 Context

- CSS Files

8. Images

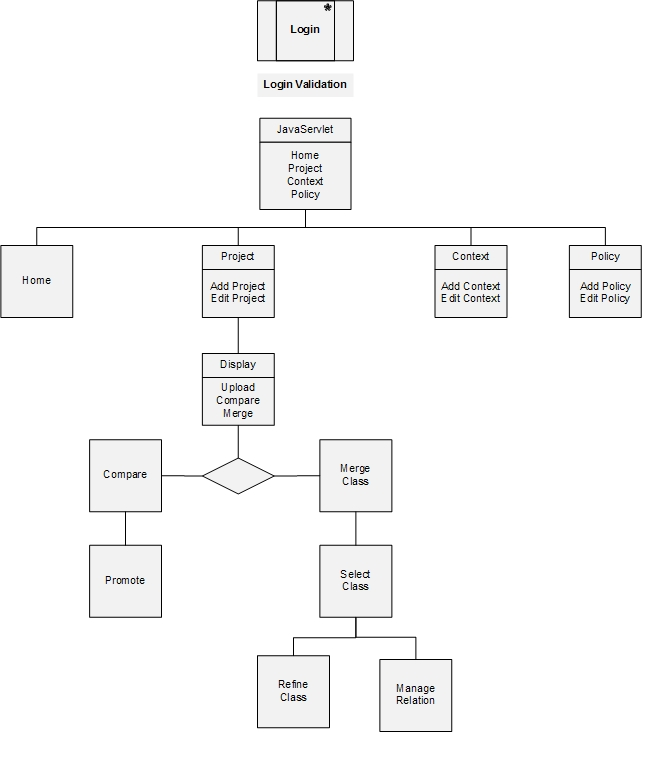
- Image Files

9. lib

- JAR Files

#### State Machine

The application has the ability to navigate through different pages for different functionality to the user. However the user is only able to get into certain pages by specific links. This provides the application to be in different states throughout the running of program. A state machine provides the developer a high level look of how a user can navigate and become in different states.



### Sprint 4

TBD

## GUI Architecture

### Software Stack

ClubUML is comprised of multiple layers of components that are built together for the entire system. Information and data is exchanged through each layer of the stack starting at the User Interface. The UI provides the ability for the user to enter and retrieve information from the system. It provides the visual representation of the system that is provided by ClubUML. For the purpose of this section of the document, we are focused on the top level layer of the software stack.

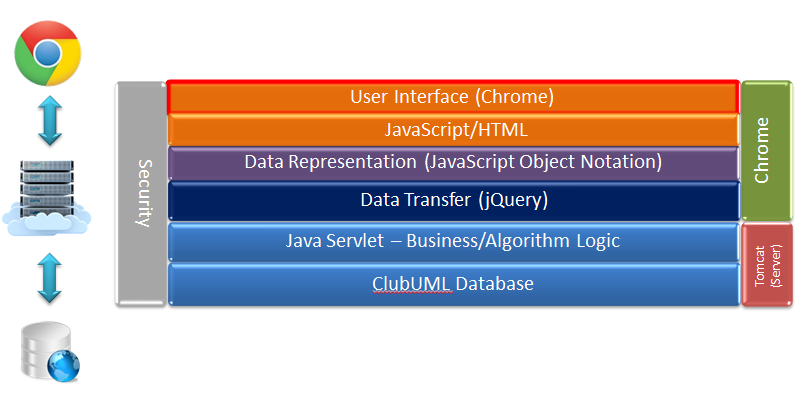


Figure : Architecture Stack

### GUI Layout

The user is presented with a series of different functionalities when first logged into the system. Prior to the main compare/merge capability of the system, the user is able to navigate through different options of projects, contexts and policies. The high level navigation can be seen below.

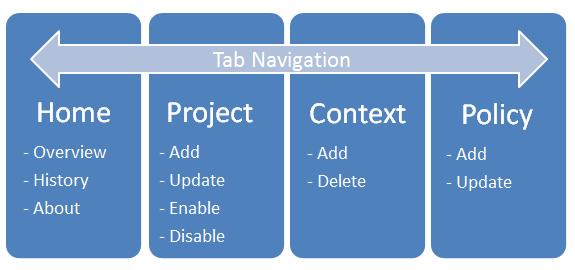


Figure : Main GUI Layout

#### Home

This is the main page for the ClubUML application. It provides an overview, history and “about” of the system. This is the first page the user sees onced logged in.

#### Project

The project page provides different functionalities to manage projects. It allows the user to Add, update, enable or disable the project. Once a project is added, the GUI brings the user to a new interface to work on the project

#### Context

The context page provides the user two different functionalities including Adding and Deleting a context. The purpose of the context is to organize projects, similar to a folder/file system

#### Policy

The policy page provides the user two different functionalities including Adding and Updating a policy. The purpose is to support roles/permissions within the system.

## Navigation Components

### Tab Navigation

Club UML is supported by a tab display to navigate throughout the entire application for easy access to the user.



Figure : Tab GUI

## Naviation/GUI Modernization

## Schedule

## Testing

The Software Test Plan (STP) for the ClubUML Application GUI/Navigation describes the testing program and protocols. The purpose of the STP is to describe the testing approach and criteria, the testing activities to be performed, the schedule for these activities, the assigned responsibilities, and the resources required.

### Resource requirements

The following applications and tools will be used for ClubUML Testing and verification.

|  |  |  |
| --- | --- | --- |
| **Component** | **Application** | **Description** |
| **Development Environment** | Eclipse | Integrated Development Environment: Open source for Java development |
| **Deployment Server** | Apache Tomcat | Open source web server |
| **Browser Client** | Mozilla Firefox 16.01 | Open source browser to run ClubUML to the user |
| **Browser Debugger** | Firebug 1.10.6 | Ability to debug JavaScript within the web browser |

### Test Cards

Library Migration URL Testing

|  |  |  |
| --- | --- | --- |
| Test Case #: 1 |  | Date: 4/4/14 |
|  | | |
| **Title** | **Description** | **P/F/M** |
| Home.jsp Library Test | The user will be able to successfully enter this page and load properly due to jquery library relocation |  |

|  |  |  |
| --- | --- | --- |
| Test Case #: 2 |  | Date: 4/4/14 |
|  | | |
| **Title** | **Description** | **P/F/M** |
| managePolicy.jsp Library Test | The user will be able to successfully enter this page and load properly due to jquery library relocation |  |
|  |  |  |

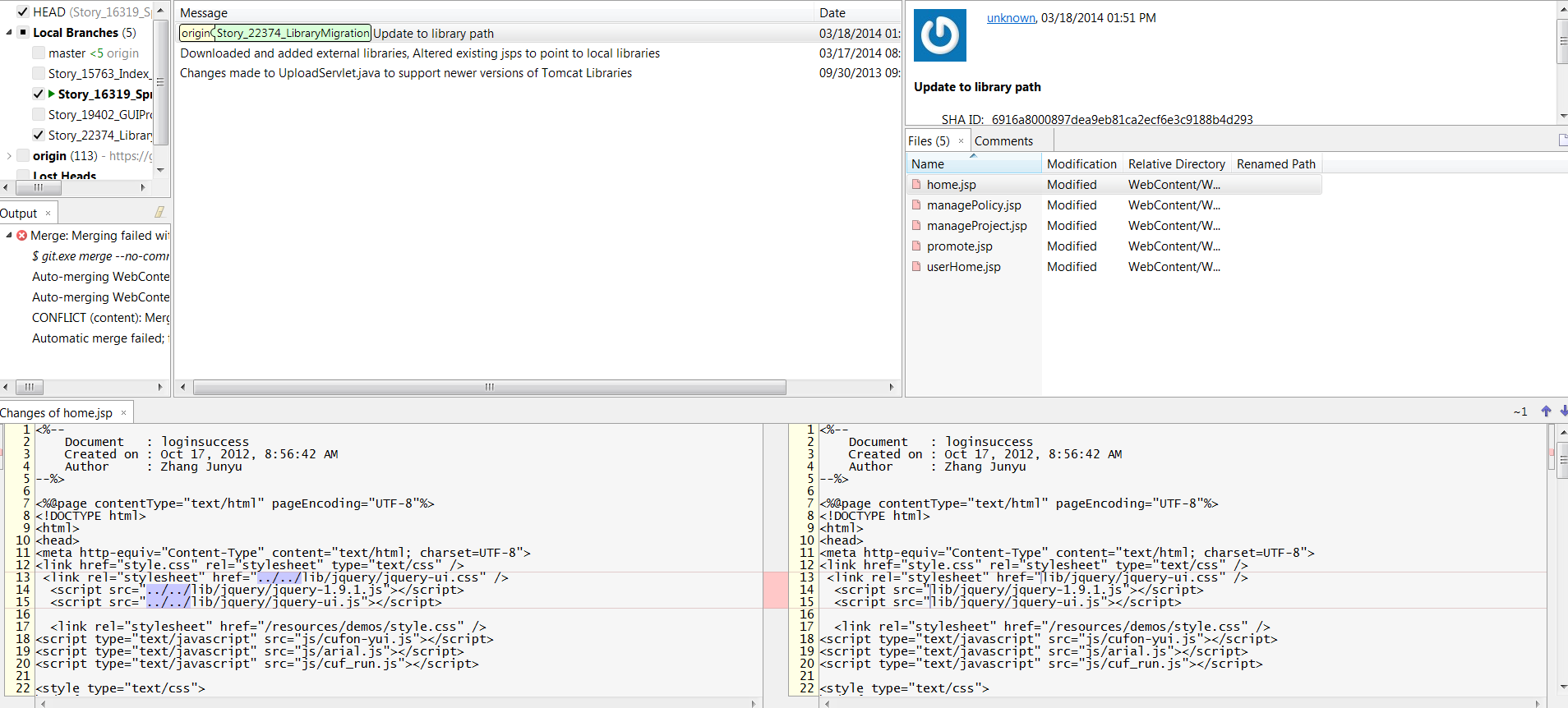
|  |  |  |
| --- | --- | --- |
| Test Case #: 3 |  | Date: 4/4/14 |
|  | | |
| **Title** | **Description** | **P/F/M** |
| promote.jsp Library Test | The user will be able to successfully enter this page and load properly due to jquery library relocation |  |

|  |  |  |
| --- | --- | --- |
| Test Case #: 4 |  | Date: 4/4/14 |
|  | | |
| **Title** | **Description** | **P/F/M** |
| manageProject.jsp Library Test | The user will be able to successfully enter this page and load properly due to jquery library relocation |  |

|  |  |  |
| --- | --- | --- |
| Test Case #: 5 |  | Date: 4/4/14 |
|  | | |
| **Title** | **Description** | **P/F/M** |
| userHome Library Test | The user will be able to successfully enter this page and load properly due to jquery library relocation |  |

### Issues:

One of the issues with integration was a branch conflict that was unable to be resolved, this caused the inability to integrate against the navigation branch. A new strategy is to find a manual way to integrate or find other resources.



## JazzHub Report

\*As of 2/2/2013

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Type** | **Id** | **Status** | **Priority** | **Severity** | **Summary** | **Owned By** | **Created By** |
| Epic | 15369 | In Progress | Medium | Normal | Navigation | Ken Canaan | Ken Canaan |
| Task | 16576 | New | Low | Normal | Create a Header in Manage Policy | Thomas Cashavelly | Thomas Cashavelly |
| Task | 16489 | New | Unassigned | Normal | Sprint 1 Navigation Documentation | Thomas Cashavelly | Thomas Cashavelly |
| Task | 16488 | New | High | Normal | Port Tab-Menu to ManagePolicy.JSP | Thomas Cashavelly | Ken Canaan |
| Task | 16487 | New | Medium | Normal | Port Tab-Menu to ManageContext.JSP and Create Header | Ken Canaan | Ken Canaan |
| Task | 16486 | New | Medium | Normal | Port Tab-Menu to ManageProject.JSP and Create Header | Ken Canaan | Ken Canaan |
| Task | 16483 | New | Medium | Normal | Port Tab-Menu to ClassMerge.JSP | Thomas Cashavelly | Ken Canaan |
| Task | 16482 | New | Medium | Normal | Port Tab-Menu into Promote.JSP | Ken Canaan | Ken Canaan |
| Task | 16479 | New | Medium | Normal | Port Tab-Menu to Compare.JSP | Ken Canaan | Ken Canaan |
| Story | 16477 | New | Low | Normal | Navigation Modernization | Thomas Cashavelly | Thomas Cashavelly |
| Task | 16472 | Done | Medium | Normal | Port Tab-Menu in Display.JSP | Ken Canaan | Afshin Chaharmahalian |
| Story | 16319 | New | Medium | Normal | Project, Context, or Policy page Navigation | Ken Canaan | Ken Canaan |
| Task | 16249 | New | Low | Normal | Import Loading Bar | Thomas Cashavelly | Thomas Cashavelly |
| Task | 16111 | Done | Medium | Normal | Research into the files need to migrate navigation pane | Thomas Cashavelly | Thomas Cashavelly |
| Task | 16110 | New | Low | Normal | Re-Design and Architect GUI Framework | Thomas Cashavelly | Thomas Cashavelly |
| Task | 16109 | New | Low | Normal | Research into JavaScript libraries to modernize navigation panels | Thomas Cashavelly | Thomas Cashavelly |
| Story | 15759 | In Progress | High | Normal | Sprint 1 Navigation: Port Tab-Menu | Ken Canaan | Jeremy Lerch |

## Future Recommendations

TBD