ExpensAble: Project Report

Mitch Dempsey, David Purpura, Ashwani Singh, and Naiyu Chen
Department of Informatics
University of California, Irvine
{ mitch.dempsey | dpurpura | ashwanis | naiyuc } @uci.edu

1 OVERVIEW

The goal of our project is to revise the current website of ExpensAble, a company who provides expense reporting services for other companies. In doing so, we sought to provide a consistent, usable experience for the three primary user roles of the system. In this report, we seek to provide an overview of usability issues that we found in the existing ExpensAble web site, and describe how we addressed these in our redesigned web application.

1.1 Process

We started by performing a cognitive walkthrough of the existing system from the perspective of each role—expense reporter, expense approver, and accountant. From these studies the predominant issues can be classified into two categories: (1) the user interface was inconsistent in terms of styling and interaction and (2) the steps required to accomplish a common task was often unclear.

Having become accustomed to the various roles, our next step was to design a coherent experience for members in each role. To do this, we started by paper prototyping different possible designs of both the layout of the site and the flow through each role using Balsamiq[1], a software tool for designing quick "paper" prototypes of user interfaces. Settling on a vertical tab-based design due to its screen real estate efficiency, we split the group into two, wherein one half served as test subjects and the other served as designers for a particular role. Once the designers had finished their prototype, the test subjects would then survey the prototypes the following week in our team meeting.

The "test subjects," having walked through the prototyped-role, would then perform a heuristic evaluation of the design with the help of the designers. The goal of this process was to be able to (1) avoid the problem of having to many designers for a single role, which we believed would slow down our design process, (2) separate the designers and the "users," as we figured that the designers would often be blind to their own mistakes, and, (3) while the designers were working, the test subjects could be working on another part of the project (usually designing or implementing another part of the system).

After a design had passed our evaluation phase, the design team began implementing the mockups following a Model-View-Presenter (MVP) architectural style using Google Web Toolkit (GWT), a Java-based user interface toolkit that compiles its code into JavaScript and HTML. Our goal was to eventually demonstrate the final implementation to a select group of

[1] Paper prototype mockups are available in the "mockups" folder of our deliverable.

current ExpensAble customer; however, we could not find a common time as a group to meet with customers at the end of the quarter.

2 DASHBOARD

The current dashboard design has various problems that prevent it from being as efficient and as user-friendly as possible. Since the dashboard is the first screen a user sees after logging into the system, it is important that this screen display information to the user clearly and effectively so that the user can be alerted about changes, as well as updates to their expense reports. By providing a detailed set of information to the user, the system can better prevent user errors and support requests that result from a lack of visual information.

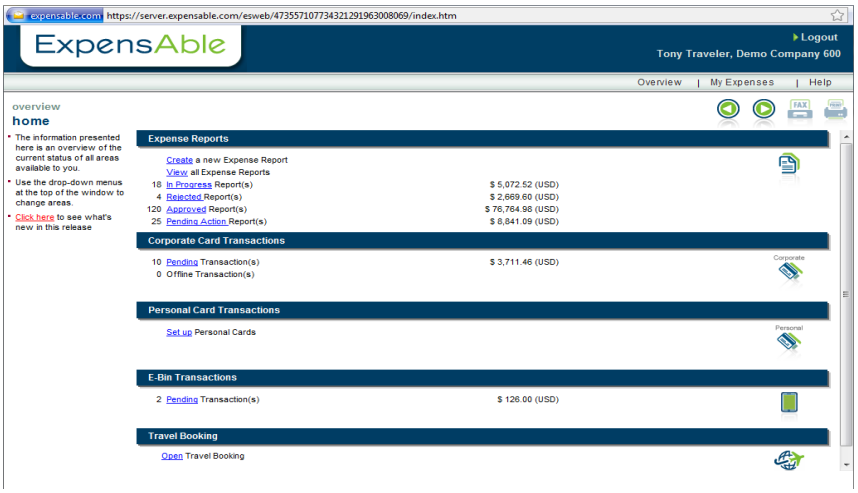

Figure 1: Screenshot of existing dashboard

The current dashboard (see Figure 1) displays a large amount of information, but in many cases this information is presented in a way that may be confusing to many users. For example, users are not clearly alerted about updates to their expense reports on the dashboard. If another user has rejected an expense report, the owner of the report is not given a clear notification that their report has been rejected. This may cause them to overlook this fact and assume that their report is all in order.

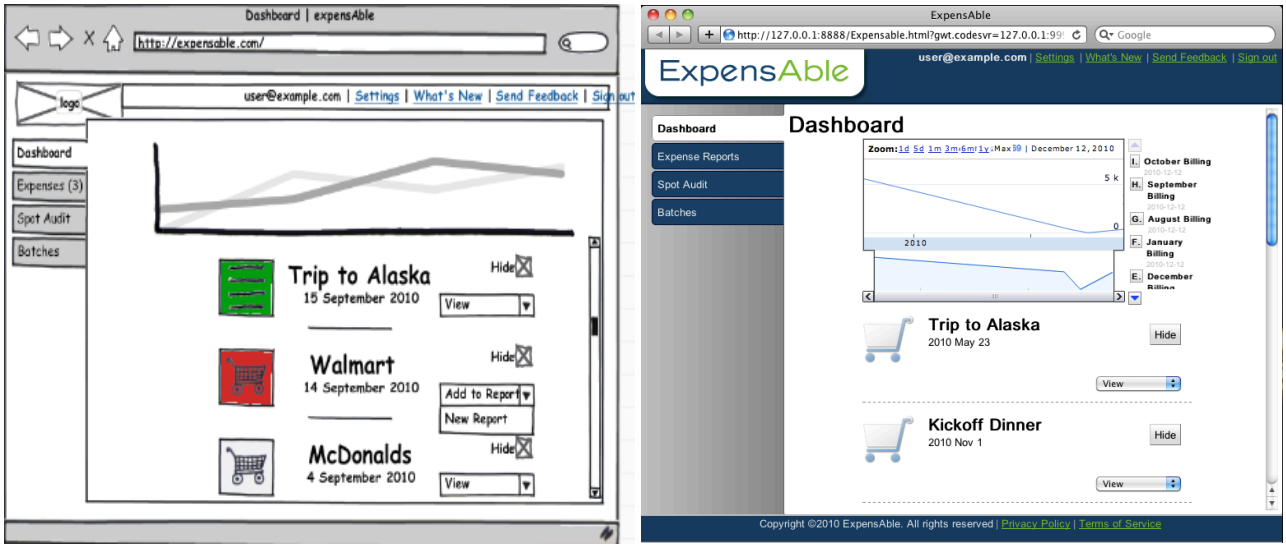

Figure 2: The dashboard mockup (left) and screenshot of new implementation (right)

In order to correct this problem, we redesigned the dashboard (see Figure 2) to provide a chart that provides a continuously updated "news feed" of recent expenses, which indicates recent activity on existing expense reports. This chart allows users to easily determine at a glance what changes have been made to their reports, and they can feel assured that they are aware of all changes. One other usability mistake made on the dashboard is the use of icons to indicate different sections. While icons can be extremely useful to clearly indicate relevant information to a user, these icons are not useful when they are not related to the item they are indicating. The icons on the current set are not clearly mapped on to their meanings (meaning that a notepad for example has no relation to digitally processed expenses). During our redesign of the dashboard, we opted to remove the icons used on the dashboard.

3 THE APPROVER ROLE

3.1 Overview

Once the expense report has been submitted for review by the Reporter (an employee), it comes to the approver who decides whether to approve it or to reject it. The approver may be the manager to whom the employee is a direct report or, in some case, it might be a designated employee whose job function is to approve or reject expense reports. Every company has some policy for reimbursements, and benefits and the expense reports are accepted or rejected based on the fact that they adhere to those policies.

3.2 System use

When the approver logs in to the ExpensAble website, using his username and password, he starts off at the dashboard screen (see Figure 1). Here, the primary goal for the

approver is to see if any new reports were submitted for his approval, and, since he is an employee himself, see the status of his own filed expense reports.

From a usability perspective, there are several issues with the design of the current screen. A good screen design provides the user with advanced information so that they know which actions are permissible and which ones are not. In our case, there are forward and back buttons on the interface to navigate between the different screens. When the approver first log in, there is no past history cached in, so the forward button should be grayed out to indicate that that action is not permissible. The fax and the printer buttons on the other hand have a slightly duller color than the rest of the icons on that row making it seem that they have been grayed out and that action is not permissible which is not the case.

Another aspect of providing advanced information to users is to provide a brief description as to what the icon does when the mouse is hovered over it. For this screen, some of the icons like the forward and the back buttons do display a description like "Forward" and "Back" but the other icons to the right of the right of Approval Information and Expense Reports do not, which violates another very important design guideline of Human-Computer Interaction, as laid down by Schneiderman, namely that of consistency. Moreover, these icons by themselves are not self-explanatory, so it confuses the users rather than helping them.

In the reports section of the dashboard, there are numbers listed next to some of the reports. The '$' sign next to these numbers indicates that they are currencies but no further information is provided as to what these currencies signify (Is this the total expenditure cost? The amount owed by the employee? Or, perhaps the amount owed to the employee?). A brief explanation should be displayed when the mouse is hovered over it.

Yet another important screen design guideline is reducing the short term memory load for the user, this can been done in this case by having the Next and the Back button display "Back to screen XYZ" rather than just displaying "Back".

Apart from that there are other things that go against this current design. Blue has been chosen as the color for foreground text when studies show that blue is best used as a background color and not a foreground color; since the fovea does not have receptors for the color blue and they are spread all across the retina. Another observation is that the options on the menu are right aligned which is non-traditional and the user takes longer time to find what he is looking for. Sub-options for these menu items are right aligned and thus, it takes a longer time for the user to read them.

Our version of this overview screen is shown in Figure 2. The dashboard immediately brings the user's attention to the things that need it most as opposed to the generic overview page that we just saw in which the user is basically "blind" to the things that need his attention unless he initiates the flow by clicking on one of the options.

Moving on from this screen, we will assume that the approver wants to see which reports require his approval. He goes ahead and clicks on the approval link and the approval screen comes is displayed (pictured below in Figure 3).

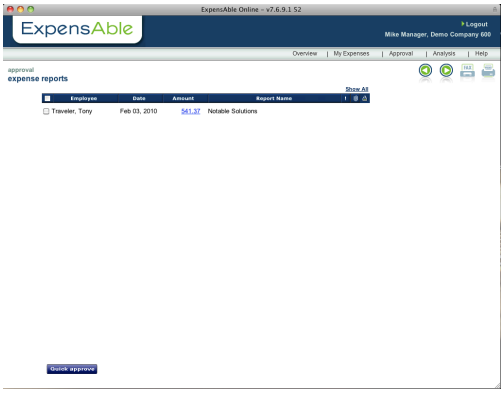

Figure 3: The expense report approval screen

The first detail that comes to attention is that the buttons and their alignment are different from the ones that we saw on the "Create Report" screen. The buttons here are purple and rectangular whereas the ones in the "Create Report" screen were gray and rounded at two corners. Moreover, the button here is left aligned whereas it was right aligned in the "Create Report" screen.

Another detail is that the folder metaphor employed in the "Create Report" and the "View Report" has been abandoned and we now have a table of expense reports waiting for approval. Thus, there is no consistency in the screens that the Reporter sees and the Approver sees which may lead to difficulties when the reporter communicates with the approver.

If the approver wants to get more information on a particular report, he has to click on the report amount in the current implementation, which is pretty bizarre. He should have to click on the "Report Id" item and not the "Report Amount" item to view the report in detail.

In our prototype, we fix these problems by providing consistent buttons and views. We also make the "Tracking Id" field as the link to view more information about the report.

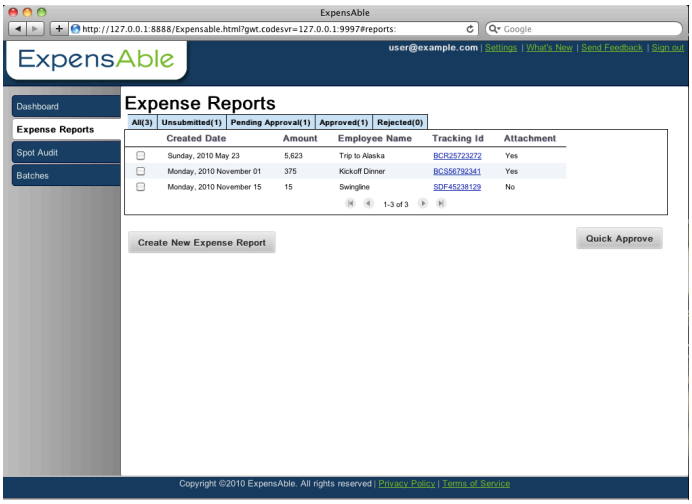

Figure 4: The expense reports list page. For approvers and accountants, a "Quick Approve" button appears on the bottom right corner of the screen.

For the Spot Audit screen, there should be no need for the “Select Search By” option as the approver can directly click on the column name to sort expense report by those column name entries. The form labels should all be right aligned and the “display” button should be below the list boxes so that the user can select all entries and then click on the button to display all the results, instead of it just floating away besides the other list boxes.

The “Select Employee” drop down menu has a different drop down button than the rest of the drop down buttons thereby breaking consistency.

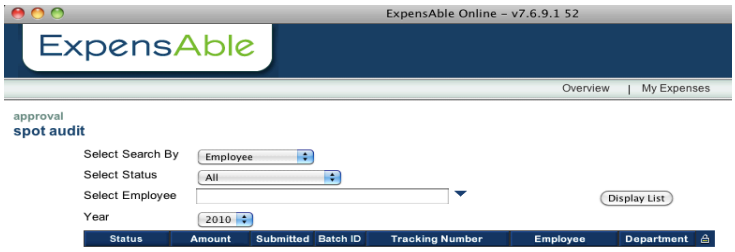

Figure 5: The existing Spot Audit screen

We take care of these issues in our prototype and add a pager mechanism to let the user know if there are more entries and they are displayed on the following pages so that the user can easily navigate between pages.

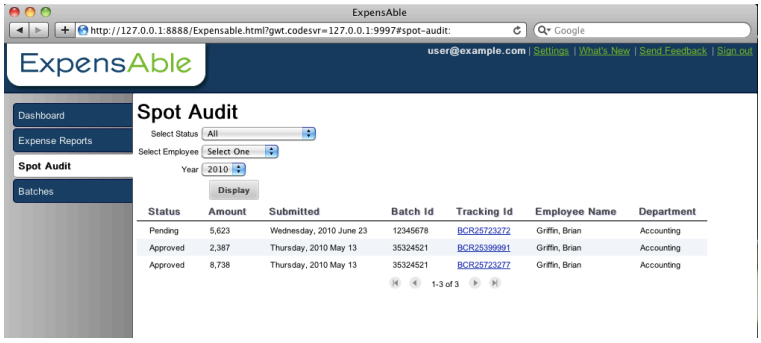

Figure 6: Our new Spot Audit screen

The rest of the options exercised by the Approver like creating and viewing expense reports have already been covered in the "Reporter" section.

4 ACCOUNTANT ROLE

4.1 Overview

Once the approver approves the expense report, it ends up at the Accounting department where the transaction is entered into the accounting book and money is dispensed at the end of the month or some other specific date when the books are closed. At the end of the month, all of the expense details for that month or the last two weeks are batched and might be exported to Excel or some other software.

4.2 System Use

The main role of the accountant in addition to the ones already described above is to do an accounting review and create batches. The accounting review screen looks as follows:

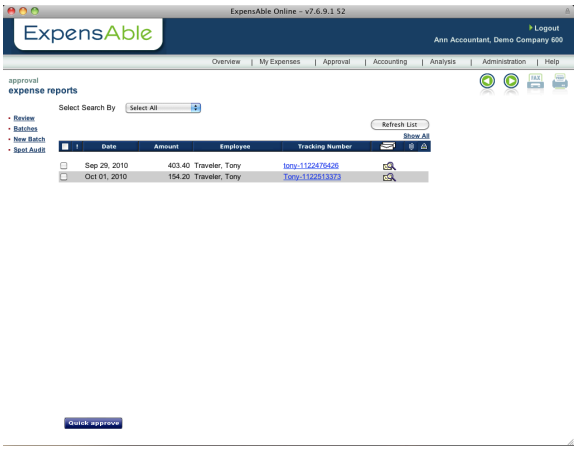

Figure 7: The accounting review screen

Again, the "Select Search By" option does not make much sense, as the accountant should be able to order searches just by clicking on the column names. The last three columns have no column name but an icon in the column name field. If this is the case, the icons should be highly intuitive and even then, a brief description should pop up to ensure that the user understands what the columns mean. In this case, the "envelope" icon and the "lock" icon are somewhat confusing and no description pops up as to what they are when the mouse is hovered over them.

The links on the left have been done with blue text which is not the best color for foregrounds because the fovea does not have receptors for the color blue and sometimes it is hard to notice text written in blue. Alternatively, the text could have been done in some other color and could have had a background color to emphasize its presence.

There is also no consistency between this view and the approver's view as the buttons are different and Tracking Id instead of the Amount links the expense reports.

For the "Create Batch" screen, the buttons are center-aligned as opposed to all other screens where they were either left aligned or right aligned. Even the list box is left aligned whereas they were left aligned in the previous screens.

Figure 8: The create batch screen

In the "View All Batches" screen, the rows in the table do not get selected even when they are clicked on, thus providing no feedback to the user. There is no consistency in the column alignments within the tables. 'Due Amount' and 'Total Amount' column values in this table are center aligned whereas they were right aligned in the table in the "Accounting Review" screen.

The alphabets and the bar below do not provide any advanced information at all. The alphabets are not links, so what are they there for? And the bar looks like a scroll bar but it is not. What is its purpose?

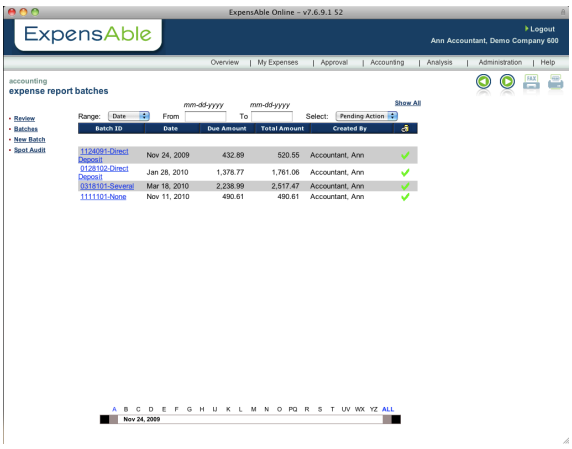

Figure 9: The view batches page

In our prototype, we combined 'Create batches' and 'View batches' because it seems like a single function. The accountant would want to see all existing batches and create one if the need arises. We also provide the ability to export a batch or to delete from the same screen.

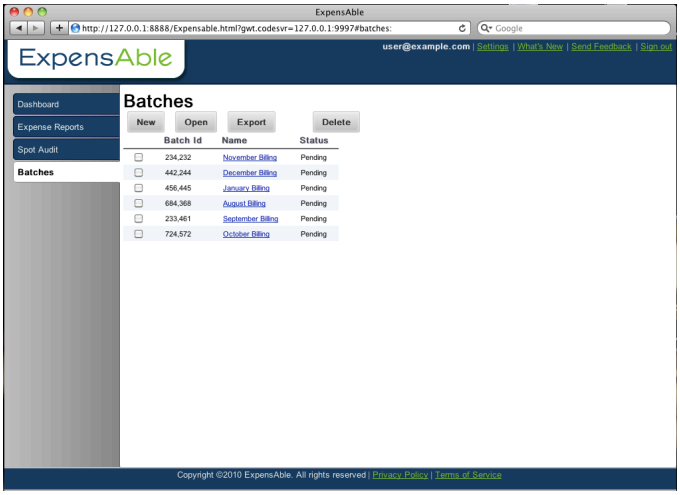

Figure 10: Our new implementation for viewing Batches

5 STYLE GUIDE

5.1 Fonts

For body text, we use a font-size of 84%, which corresponds to roughly 10 pixels. Though we target a size of 10 pixels, we use a percent based unit because some browsers have problems scaling layouts with pixel specified text.

Note that the font size and style does not cascade into inputs, buttons, or select in CSS, so we must set it manually. We also increase the font size to 100% (12 pixels) in order to make the options more readable.

5.1.1 CSS

```
body, table td {

  font-family: Arial Unicode MS, Arial, sans-serif;

  font-size: 84%;

}

input, button, select {
```

```
  font-family: Arial Unicode MS, Arial, sans-serif;
  font-size: 100%;
}
pre {
  font-family: "Courier New", Courier, monospace;
  font-size: 84%;
}
```

5.1.2 *Some useful links:*

- Describes the tradeoffs for different sizing techniques. <http://kyleschaeffer.com/best-practices/css-font-size-em-vs-px-vs-pt-vs/>
- Suggests different "font stacks" or font choices based on the fonts that clients are likely to have. <http://www.awayback.com/revised-font-stack/>

5.2 Header

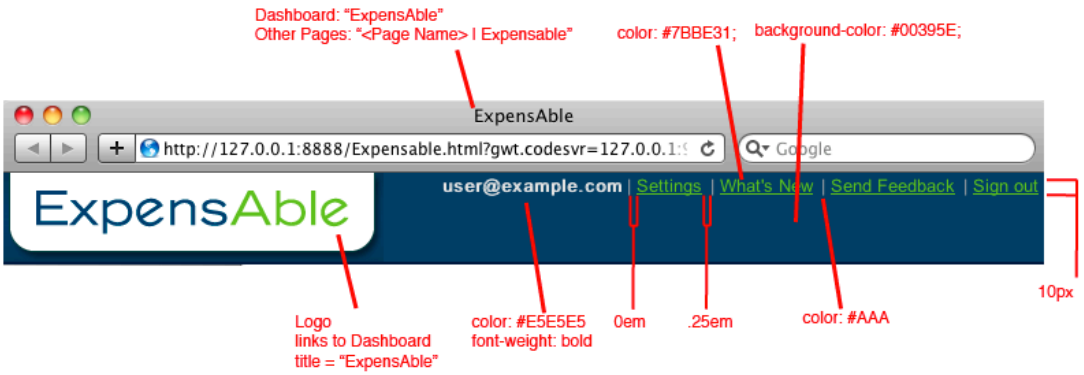

Figure 11: The application header

5.2.1 *Features*

1. **Logo:** The logo in the top-left corner should link to the main page. In the case of the ExpensAble site, this should link to the Dashboard.

2. **Email address:** Instead of displaying the user's name, use the user's email address. This way, if single user "Tony the reporter" has accounts for multiple companies on ExpensAble, he knows which account he is logged into.

3. **Settings:** links to the user settings page. This page is application dependent, but common features on this page are changing his name, password, and privacy settings.

4. **What's New:** Provides ExpensAble with a means by which to notify users of new features. For more important changes, consider changing the color of this to red, or another color that will stand out.

5. **Send Feedback:** Provides an email address for customers to quickly and easily provide feedback on the application.

6. **Sign out:** Logs the user out of the application.

5.2.2 HTML

```
  <div class="header">

    <span class="user-info">

      <!-- Note: formatting is intentional: span should start at EOL -->

      <span class="username>user@example.com</span>|<span>

        <a href="#">Settings</a></span>|<span>

            <a href="#">What's New</a></span>|<span>

        <a href="mailto:example@example.com">Send Feedback</a></span>|<span>

        <a href="#">Sign out</a></span>

    </span>

    <a href="#"><img title="ExpensAble"

        src="https://server.expensable.com
            /EComImages/menu/EXP_UI_logo.gif" border="0"/></a>

  </div>
```

5.2.3 CSS

```
.header {

  background: #00395E;

  border-bottom: 2px solid #001E44;

}

.user-info {

  color: #AAA;
```

```
  float: right;
}
.username {
  font-weight: bold;
}
.user-info a, .user-info span, .footer span {
  color: #E5E5E5;
  padding: 0 0.25em 0 0;
}
.user-info a, .user-info a:visited {
  color: #7BBE31;
}
```

5.3 Footer

Figure 12: The application footer

For the page footer, we opted to use a hanging footer (a footer that always remains visible, even if the page requires scrolling). The benefit of this is that users can quickly find the Privacy Policy and Terms of Service. However, since these links are likely not to be commonly used by a user, it is acceptable put the footer at the bottom of the scrolled page.

5.4 Vertical Tab Bar

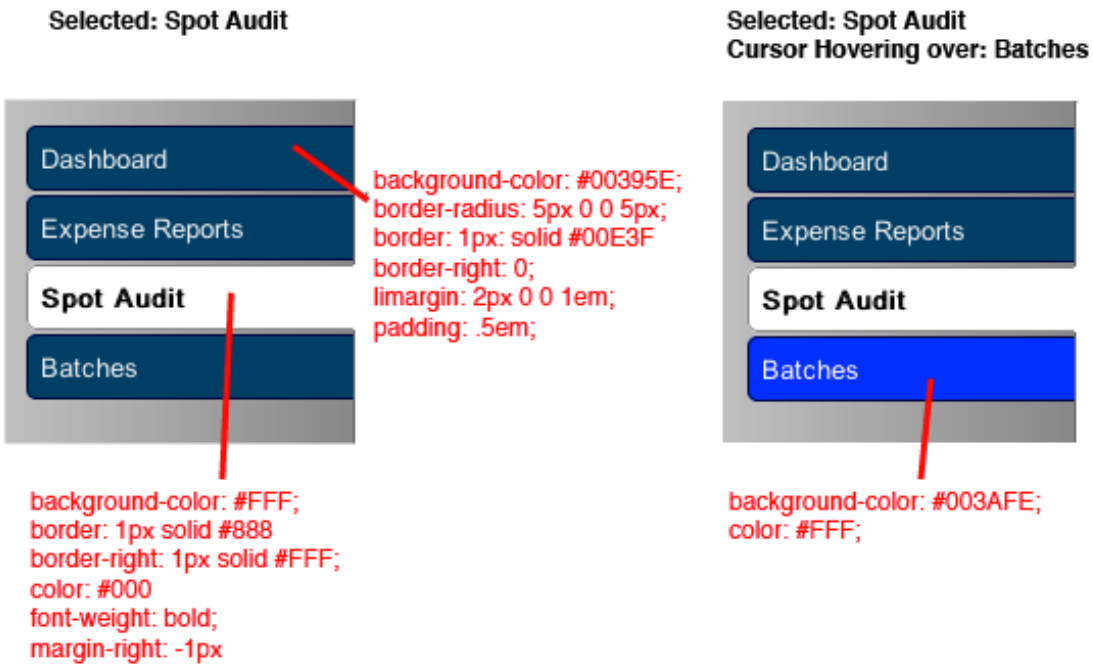

Figure 13: The vertical tab bar

In order to navigate through the web site, we use a vertical tab bar. This provides two major benefits. First, if we consider the navigation of the site as being a "left to right" process, the vertical tab bar shows the origin of the navigation process. Furthermore, the selected tab provides the user with a clear indication of where he is, so that he knows how to get back to it in the future.

5.5 Uniform Resource Locator (URL)

5.5.1 Each page should have its own URL

URLs should follow the rules of the Representational State Transfer (REST) architectural style. Accordingly, each page should have its own unique URL. By following this standard, pages can be bookmarked, and, if exposed to search engines, crawl-able.

Consider a site with a root URL of: `https://server.expensable.com`

	HTTP Method	URL
BAD	GET	/esweb/75723439340646129210454435 2/index.htm
GOOD	GET	/reports/1233
GOOD	GET	/batches/1235

GOOD	GET

Figure 14: URL Examples

5.5.2 Browser History should be managed by the browser

The application should **NEVER** manage its own history. By managing its own history, the application must provide its own back and forward buttons. Since a user is accustomed to using the back and forward buttons on the browser, having its own back and forward buttons (or even browser shortcuts) can confuse the user. Furthermore, without storing resource state on the URL when the user clicks the back and forward button, all state is lost. Instead, the application should use the DOM's API for manipulating browser history.

5.5.3 Relevant Material

1. Fielding, Roy. Chapter 5: Representational State Transfer (REST) <http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm >

2. Richardson, L, and S Ruby. *RESTful Web Services*. O'Reilly 2007.

3. Mozilla. "Manipulating the Browser History." <https://developer.mozilla.org/en/DOM/Manipulating_the_browser_history>

5.6 Buttons

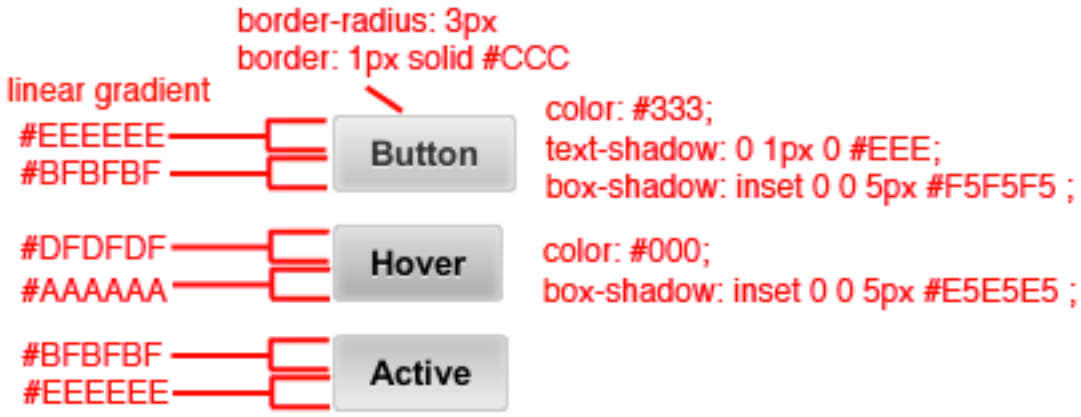

Figure 15: Our custom-styled buttons in normal, hover, and active states

5.6.1 Features

Buttons should be consistent across all properties. Although color changes are valid for special cases, such as a call to action, disabling a button, or providing a sense of warning.

Buttons should provide a sense of feedback to the user. In the example above, when the user hovers over a button, the background and text color get darker, while when the user clicks on the button, the button reverses the original color gradient.

5.6.2 HTML

```
<a class="button"/>
```

5.6.3 CSS

```
.button {

  background: #eee;

  background: -moz-linear-gradient(0% 100% 90deg, #bfbfbf, #eee);

  background: -webkit-gradient(linear, 0% 0%, 0% 100%, from(#eee), to(#bfbfbf));

  border: 1px solid #ccc;

  -moz-border-radius: 3px;

  -webkit-border-radius: 3px;

  border-radius: 3px;

  -moz-box-shadow: inset 0 0 5px #f5f5f5;

  -webkit-box-shadow: inset 0 0 5px #f5f5f5;

  box-shadow: inset 0 0 5px #f5f5f5;

  color: #333;

  font-size: 100%;

  font-weight: bold;

  line-height: 1;

  margin: 0.5em;

  padding: .5em 1em;

  text-align: center;

  text-shadow: 0 1px 0px #eee;

  text-decoration: none;

}
```

```
.button:hover {
  background: #dfdfdf;
  background: -moz-linear-gradient(0% 100% 90deg, #aaa, #dfdfdf);
  background: -webkit-gradient(linear, 0% 0%, 0% 100%, from(#dfdfdf), to(#aaa));
  -moz-box-shadow: inset 0 0 5px #e5e5e5;
  -webkit-box-shadow: inset 0 0 5px #e5e5e5;
  box-shadow: inset 0 0 5px #e5e5e5;
  color: #000;
}
.button:active {
  background: #bfbfbf;
  background: -moz-linear-gradient(0% 100% 90deg, #eee, #bfbfbf);
  background: -webkit-gradient(linear, 0% 0%, 0% 100%, from(#bfbfbf), to(#eee));
}
```

5.7 Button Groups

Figure 16: To prevent errors, separate irreversible actions from safe actions.

If a button corresponds to an irreversible action, it should be offset from the other buttons in the group in order to prevent the user from accidently clicking that button. In the figure above, notice how the "Delete" button is further from the "Export" button than the "Export" button is from the "Open" button.

Additionally, it may be wise to confirm such a change with a confirmation dialog.

6 TABLE

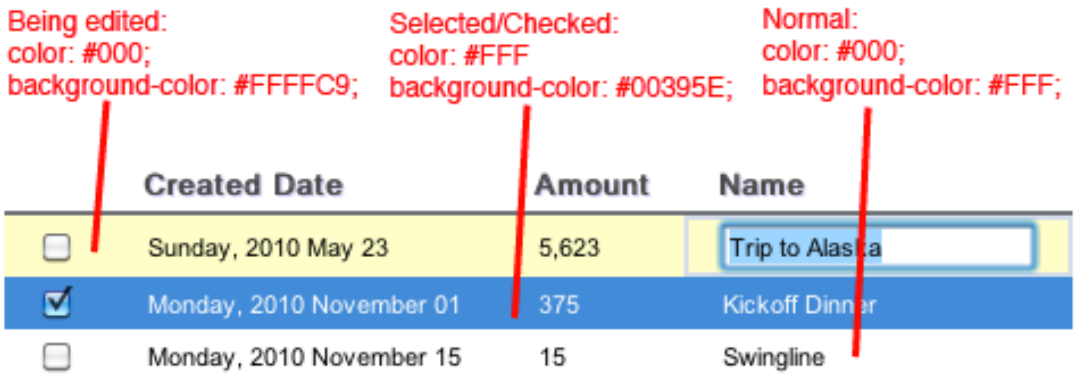

Figure 17: An editable table

6.1 More Information

The table provided is a GWT Cell Table widget, more information can be found about it from the links below.

- http://code.google.com/webtoolkit/doc/latest/DevGuideUiCellWidgets.html#celltable
- http://gwt.google.com/samples/Showcase/Showcase.html#!CwCellTable