Shopping Cart

New to Tiki5

We now have a basic Cart

What

The plan is to build a lightweight, basic e-commerce system in TikiWiki by making good use of existing features and by adding a few things. E-commerce is was one of the rare Use Cases not addressed by Tiki until Tiki5. Please read Skype log below for debate/discussion.

A typical use will be a mostly informational website to sell a few products & services.

- Selling registration to various packages for events. (Ex.: day 1 package, day 2 package, full event package)
- Selling memberships (Ex.: 1 year, 3 year, etc.)
- Selling a few dozen products (books, software downloads, etc)

For this, the overhead of a full-featured shopping cart like ZenCart is not necessary.

Wishes

For Shopping Cart and Payments.

Open & Pending

<table>
<thead>
<tr>
<th>Rating</th>
<th>Subject</th>
<th>Submitted by</th>
<th>Importance</th>
<th>Easy to solve?</th>
<th>Priority</th>
<th>Category</th>
<th>Volunteered to solve</th>
<th>LastModif</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>★★★★</td>
<td>Plugin</td>
<td>heb</td>
<td>7</td>
<td>0</td>
<td></td>
<td>Feature</td>
<td>2010-10-04</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Relation By</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>request</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>★★★★</td>
<td>Views / “Other users also read”</td>
<td>xavi-30 Jun 16</td>
<td>7</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>2014-04-19</td>
<td></td>
</tr>
<tr>
<td>★★★★</td>
<td>likewise to “Customers also bought”</td>
<td>xavi-30 Jun 16</td>
<td>7</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Arild Berg  | ▼▼▼▼ Shopping Cart does not handle decimal prices | 2014-04-19 | 7 | ▼▼▼▼ Error | xavi-30 Jun 16 |
Closed

[+]

Who & What

- Marc Laporte, requirements & testing
- Sylvie Greverend, coding
- Nelson Ko
- Robert Plummer

Details

Basic online ordering functionality for a Tiki-powered site can be seen at reseau-environnement.com. Wiki pages are used for product information and the tracker forms are filled thanks to PluginTrackerPrefill. Also, some rewrite-rules were used to handle https. It's sufficient for this project but it's clunky. And it only works for single item shopping. There is no connection to a payment gateway. No calculated totals.
There are some efforts underway for a full-fledged integration with ZenCart. This will be the option for people with more advanced e-commerce needs (upsell / cross-sell, multiple currencies, inventory, discounts, etc.) and it will require more work to install and maintain.

We need something better than the forms we have but not to the level of a standalone shopping cart.

**Plan**

For the new lightweight Tiki Cart or (Tiki Commerce?), from current code base, we'll use:

- User trackers for profiles (name, phone number, ship to address, bill to address, etc).
- trackers to manage products, with categories and filters
- Wiki pages for products which include structured data from trackers, and use nice template (editable as a wiki page).
- As suggested by Alain (on the Skype chat), only ask to fill out profile after the order is ready to go.
- To keep things simple, there would be an address (maybe 2 since sometimes, *bill to* is different than *ship to*) in the user profile, and this info would be used at the time of the order (and thus recorded with the order)
- tiki-cart.php should be not only to add items to a shopping cart, it should also be more generic to add items to other batch actions (ex.: print all wiki pages in cart, mass delete all these pages, mass watch all these pages, etc)

**Needed:**

- "Add to cart" button in wiki pages
- A simple cart like this
- better https handling
- link to PayPal and Google Checkout
- Email confirmations

**Nice to have:**

- Order logging
  - For sure, a copy of the order will be sent out by email (to site administrator and to the customer).
  - If we use Paypal, a copy of the info will be logged in Paypal. However, for b2b, payment will not be handled. So it would be nice to have a Web interface to check previous orders. Users should be able to access this from their user profile.

A full shopping cart would be the next step.

- VirtueMart features (for Joomla!). Tom Aratyn likes it.
- Zen cart features
Considerations for Tiki 9

Handling of cart payment validity period and abandoned carts

Jonny and I was having a short discussion about this.

When the original payments system was created it was more in the context of requesting a payment from some other user, hence the payment delay (before it becomes overdue) was set to days that made sense. However, for a cart situation, it does not make sense that the user can have the whole day to pay for his order.

To change the interval from days to minutes is easy enough in the code, simply to change "DAY" to "MINUTE" in the query in request_payment() in lib/payment/paymentlib.php. However this function is called not only from cartlib. It is called from a bunch of other places as well, and to add "* 1440" in all these other places is not a clean solution. So it seems best to be to change the pref to support minutes as well. But this is too late for 8.x now we will do it in trunk.

The other thing I had tried to make the payment system more cart like is to run a cron job that runs every 5 minutes that checks if there are any overdue requests. If so, it cancels them, thus reinstating the inventory and all that since the "cancel_cart_order" behavior executes when the payment request is cancelled... This isn't critical if you are not keeping track of inventory I suppose, but cancelling overdue payment requests does make sense anyway otherwise you will have a whole bunch of them sitting around after a while, and it means shoppers can still come back and pay even if it's overdue (which is kind of weird in a cart situation). Another thing for trunk.

The cron job:

```php
if (php_sapi_name() != 'cli') {
    die;
}
require_once ('tiki-setup.php');
global $paymentlib;
require_once ('lib/payment/paymentlib.php');
global $perms;
$perms->setGroups(array('Admins'));
$overdue = $paymentlib->get_overdue(0, -1);
foreach($overdue['data'] as $ov) {
    $paymentlib->cancel_payment( $ov['paymentRequestId'] );
}
```
This is also not yet committed. Is there some new fancy cron things in Tiki that can do this kind of thing? Timed events? Also, theoretically, payment system could be used for not just the cart on a particular Tiki, and so different handling logic for different overdue payments could be needed.... in the case of cart, to cancel it, or in other cases sending reminders perhaps?

Just some things to think about.

Nelson.

Related links

- Shopping Cart Tiki7
- Accounting
- Credits
- Payment
- Payments Pro
- PayPal Subscription
- Community Currencies take2: see CCLite integration with OsCommerce

Please also see:

- Review of 9 open source e-commerce platforms
- PayPal Shopping Cart Integration
- Shopping Cart / ecommerce / online orders
- Comparative chart of ecommerce solutions (French)
- http://www.treasurechestcart.com
- http://www.instinct.co.nz/e-commerce/

Skype chat log 2008-06-11

See also

- Multi-sided platform
- Auction

Alias

- Cart or Carts
- Shopping Carts
- Commerce, E-Commerce, or ECommerce