1. PitchPoint API & Credential Information

Census data link request:   
   
<https://www.usinfosearch.com/resell/Census.php?username=&password=&zipcode=&clientReference=test>  
   
Username= dls1  
Password= dls2012!  
   
Sample Request:  
   
<https://www.usinfosearch.com/resell/Census.php?username=dls1&password=dls2012!&zipcode=43125&clientReference=test>  
   
   
Sample Response:  
   
<response>  
<stats>  
<provider>U.S.Infosearch.com</provider>  
<searchInput>ZipCode=43125</searchInput>  
<clref/>  
<searchTime>12/16/2012 8:48 am</searchTime>  
<rows>1</rows>  
</stats>  
<census>  
<zip>43125</zip>  
<zipdetails>  
<zipdetail>  
<state>OH</state>  
<city>GROVEPORT</city>  
<lat>39.8572</lat>  
<lng>82.8912</lng>  
<ac>614</ac>  
</zipdetail>  
</zipdetails>  
<population>9601</population>  
<farm>122</farm>  
<nonfarm>1834</nonfarm>  
<white>8805</white>  
<black>380</black>  
<indian>104</indian>  
<asian>121</asian>  
<hawaiian>6</hawaiian>  
<raceOther>185</raceOther>  
<hispanic>150</hispanic>  
<eduLess9>242</eduLess9>  
<edu912>880</edu912>  
<eduHighsc>2739</eduHighsc>  
<eduSomecl>1871</eduSomecl>  
<eduAssoc>476</eduAssoc>  
<eduBach>782</eduBach>  
<eduProf>291</eduProf>  
<hhIncome>46438</hhIncome>  
<houseVal>115700</houseVal>  
</census>  
</response>

1. ICE API & Credential Information

Agent  
URL:  <https://ws.ourvacationstore.com/membership/Agent.asmx>  
WSDL:  <https://ws.ourvacationstore.com/membership/Agent.asmx?WSDL>  
   
Certificate  
URL:  <https://ws.ourvacationstore.com/membership/Certificate.asmx>    
WSDL:  <https://ws.ourvacationstore.com/membership/Certificate.asmx?WSDL>  
   
Content  
URL:  <https://ws.ourvacationstore.com/membership/Content.asmx>  
WSDL:  <https://ws.ourvacationstore.com/membership/Content.asmx?WSDL>  
   
Member  
URL:  <https://ws.ourvacationstore.com/membership/Member.asmx>    
WSDL:  <https://ws.ourvacationstore.com/membership/Member.asmx?WSDL>  
   
Sailing  
URL:  <https://ws.ourvacationstore.com/membership/Sailing.asmx>    
WSDL:  <https://ws.ourvacationstore.com/membership/Sailing.asmx?WSDL>  
   
Transaction  
URL:  <https://ws.ourvacationstore.com/membership/Transaction.asmx>    
WSDL:  <https://ws.ourvacationstore.com/membership/Transaction.asmx?WSDL>

1. StickyStreet API & Credential Information

ReferralClix StickyStreet Credentials:  
[https://rewards.clienttoolbox.com](https://rewards.clienttoolbox.com/)  
UN: hwproduction  
Password: hw2010  
  
Client to use to test while developing:  
hwproduction  
  
StickyStreet API:  
<http://api.clienttoolbox.com/>  
  
Enterprise API URL & Login  
<http://api.clienttoolbox.com/agency/index.html>  
username: hwproductions  
pswd: quir4whan

1. Meridian API & Credential Information

Meridian Link API:  
<https://demo.mortgagecreditlink.com/inetapi/AU/get_credit_report.aspx?HELP>  
  
Here is the production URL.  
<https://ucs.meridianlink.com/inetapi/AU/get_credit_report.aspx>  
   
New Submitting Party ID:  
DestinationLifestyles02152013  
   
Your credit login is as follows:  
   
Login: destination  
Pass: lifestyles8

1. Stupefix API

http://developer.stupeflix.com  
  
<http://stupeflix-api.readthedocs.org/en/latest/>  
  
  
  
[markwitte@me.com](mailto:markwitte@me.com)  
Freedom1

1. Google+ API

<https://developers.google.com/+/api/>

1. Google Analytics

<https://developers.google.com/analytics/>

1. Mailchimp API

http://apidocs.mailchimp.com  
  
  
  
Mandrill  
[markwitte@me.com](mailto:markwitte@me.com)  
Freedom1!

1. Facebook API

<http://developers.facebook.com>

1. Pinterest API

<https://help.pinterest.com/entries/21151603-I-m-a-developer-Does-Pinterest-have-a-public-API->

1. Clicky API

Clicky:  
U: resortrewards  
P: abc123

1. Header API Calls

The header on each page will include the business name, logged in user's username, a support phone number, support email address and the business logo  
  
Here are the API calls that relate to that information:  
  
When you issue a Client Info API call (account\_info -- <http://api.stickystreet.com/account_info.html> ) include this undocumented parameter to retrieve that information:  
  
include\_agency = Y  
  
This will result in the additional XML nodes like these:  
  
 <agency\_id>salongeeks</agency\_id>  
 <agency\_name>Salon Geeks 2</agency\_name>  
 <agency\_email>[info2@salongeeks.com](mailto:info2@salongeeks.com)</agency\_email>  
 <agency\_phone>555-5552</agency\_phone>  
 <agency\_logo>salongeeks\_logo.jpg</agency\_logo>  
  
The URL / full path for the logos is:  
  
<https://salongeeks.clienttoolbox.com/branding/salongeeks_logo.jpg>  
  
Make sure to replace the subdomain with the returned <agency\_id> node content.

1. Template Files

Sprint 1

1. User Info Screen

This screen is going to have identical functionality as the StickyStreet Saint 2 login screen.  
  
Here is the link:  
<http://hwclients.com/rewardclix/user_info.html>  
  
On this screen, a logged in user of a business/merchant can update their first name, last name, password, preferred currency and timezone.  
  
This screen is for users who are in the system already. We should pre-load all the data that is currently saved for the user in the text fields.  
  
None of the fields are required, as we are only updating that particular field for an individual user.  
  
Here are the links to the appropriate API documentation:  
<http://api.clienttoolbox.com/user_update.html>

1. Account Info Screen

This screen is going to have identical functionality as the StickyStreet Saint 2 screen.  
  
Here is the link:  
<http://hwclients.com/rewardclix/account_settings.html>  
  
This screen allows a business user to update their business information such as business name, address, city, state/province, zip, and country.  
  
All of the information that is already associated with the account should pre-populate in the form fields.    
  
None of these fields are required. However, there should be some validation to make sure that if a new password is entered then it should match in the "Re-type" field. Also, if one password field is completed then the other should be as well.  
  
Here is the link to the appropriate API documentation:  
<http://api.clienttoolbox.com/account_info.html>  
<http://api.clienttoolbox.com/account_update.html>

1. Login to StickyStreet

This screen is going to have identical functionality as the StickyStreet Saint 2 login screen.  
  
Here is the link:  
<http://hwclients.com/rewardclix/login.html>  
  
This screen is going to be the new login screen to the dashboard.  
  
The dashboard we are building is for clients/businesses. Clients/Businesses have Account ID's which allow them to log into a client-centric interface we call the Client Dashboard.  
Clients can define "Users" with specific roles and permissions.  
  
The User Login API call (<http://api.stickystreet.com/agency/user_login.html>) allows the validation of a User or Client ("owner") login credentials, for access to client-centric interfaces.  
  
Important Information after Login:  
Different users will have different access to items within the dashboard based on the roles assigned to them. We asked StickyStreet now to determine which roles get which access to which portions of the dashboard. Here is how they explained it:  
  
The best way to address this issue is to look at the permissions that an administrator user has, which is returned as part of the User Info API call (user\_info - <http://api.stickystreet.com/user_info.html> ). When using an account "owner" id, the XML will return all possible permissions (see XML list below). When the call is made from a different user's login, only those permissions that that user has are returned. An example below, as well, shows the much shorter list that a "clerk" level user has.  
  
The implementation strategy I would suggest would wrap each affected section, button, etc with a conditional (if statement) that tests whether the corresponding node in the XML exists, and if so, if its value is 'Yes'. This is the most flexible system we can devise, allowing tweaking of user permissions to occur at the definition layer, without affecting the display and business logic layer (ie: if everyone votes/agrees to restrict a "clerk" user from a particular capability, removing that XML node from the list in the API result should automatically turn off that capability without the interface needing to know that a change occurred.  
  
Let me know if this doesn't make sense -- I can demonstrate.  
  
Below are the XML returns for the HW Productions demo account "HW 2" (id: nicole2)  
  
OWNER API CALL:  
  
Data submitted to API at: <https://app.clienttoolbox.com/api.php>  
Array  
(  
 [user\_id] => nicole2  
 [user\_password] => fe24fc239e6c3f4cc80106dfa425fe48c5fcd1ff  
 [type] => user\_info  
 [account\_id] => nicole2  
)  
  
--- XML Result: (5105 bytes) ----  
  
<?xml version="1.0" encoding="utf-8" ?>  
<response status="success">  
 <user>  
 <user\_id>nicole2</user\_id>  
 <user\_is\_owner>true</user\_is\_owner>  
 <user\_biz\_name>HW 2</user\_biz\_name>  
 <user\_language>EN</user\_language>  
 <user\_language\_custom>No</user\_language\_custom>  
 <user\_timezone>14</user\_timezone>  
 <user\_level>A</user\_level>  
 <user\_permissions>  
 <activate\_account>Yes</activate\_account>  
 <view\_account\_info>Yes</view\_account\_info>  
 <edit\_account\_info>Yes</edit\_account\_info>  
 <manage\_campaign\_fields>Yes</manage\_campaign\_fields>  
 <api\_user\_long\_account\_info>Yes</api\_user\_long\_account\_info>  
 <add\_campaign>Yes</add\_campaign>  
 <delete\_campaign>Yes</delete\_campaign>  
 <activate\_campaign>Yes</activate\_campaign>  
 <deactivate\_campaign>Yes</deactivate\_campaign>  
 <open\_campaign>Yes</open\_campaign>  
 <access\_campaign\_settings>Yes</access\_campaign\_settings>  
 <update\_campaign>Yes</update\_campaign>  
 <edit\_campaign\_name>Yes</edit\_campaign\_name>  
 <view\_gift\_card\_prefs>Yes</view\_gift\_card\_prefs>  
 <edit\_gift\_card\_prefs>Yes</edit\_gift\_card\_prefs>  
 <view\_points\_prefs>Yes</view\_points\_prefs>  
 <edit\_points\_prefs>Yes</edit\_points\_prefs>  
 <view\_events\_prefs>Yes</view\_events\_prefs>  
 <edit\_events\_prefs>Yes</edit\_events\_prefs>  
 <view\_buyx\_prefs>Yes</view\_buyx\_prefs>  
 <edit\_buyx\_prefs>Yes</edit\_buyx\_prefs>  
 <view\_earn\_prefs>Yes</view\_earn\_prefs>  
 <edit\_earn\_prefs>Yes</edit\_earn\_prefs>  
 <view\_campaign\_email>Yes</view\_campaign\_email>  
 <edit\_campaign\_email>Yes</edit\_campaign\_email>  
 <view\_point\_to\_dollar\_ratio>Yes</view\_point\_to\_dollar\_ratio>  
 <edit\_point\_to\_dollar\_ratio>Yes</edit\_point\_to\_dollar\_ratio>  
 <view\_profit\_ratio>Yes</view\_profit\_ratio>  
 <edit\_profit\_ratio>Yes</edit\_profit\_ratio>  
 <view\_campaign\_points\_expiration>Yes</view\_campaign\_points\_expiration>  
 <add\_campaign\_points\_expiration>Yes</add\_campaign\_points\_expiration>  
 <edit\_campaign\_points\_expiration>Yes</edit\_campaign\_points\_expiration>  
 <delete\_campaign\_points\_expiration>Yes</delete\_campaign\_points\_expiration>  
 <view\_promotions>Yes</view\_promotions>  
 <edit\_promotions>Yes</edit\_promotions>  
 <add\_promotions>Yes</add\_promotions>  
 <delete\_promotions>Yes</delete\_promotions>  
 <view\_rewards>Yes</view\_rewards>  
 <edit\_rewards>Yes</edit\_rewards>  
 <add\_rewards>Yes</add\_rewards>  
 <delete\_rewards>Yes</delete\_rewards>  
 <show\_dollars\_earned>Yes</show\_dollars\_earned>  
 <edit\_dollars\_earned>Yes</edit\_dollars\_earned>  
 <view\_buyx\_items>Yes</view\_buyx\_items>  
 <edit\_buyx\_items>Yes</edit\_buyx\_items>  
 <add\_buyx\_items>Yes</add\_buyx\_items>  
 <delete\_buyx\_items>Yes</delete\_buyx\_items>  
 <view\_buyx\_default>Yes</view\_buyx\_default>  
 <edit\_buyx\_default>Yes</edit\_buyx\_default>  
 <view\_depreciations>Yes</view\_depreciations>  
 <add\_depreciation>Yes</add\_depreciation>  
 <add\_depreciations>Yes</add\_depreciations>  
 <edit\_depreciation>Yes</edit\_depreciation>  
 <edit\_depreciations>Yes</edit\_depreciations>  
 <delete\_depreciation>Yes</delete\_depreciation>  
 <delete\_depreciations>Yes</delete\_depreciations>  
 <view\_recurring\_fee>Yes</view\_recurring\_fee>  
 <add\_recurring\_fee>Yes</add\_recurring\_fee>  
 <edit\_recurring\_fee>Yes</edit\_recurring\_fee>  
 <delete\_recurring\_fee>Yes</delete\_recurring\_fee>  
 <view\_customer\_search>Yes</view\_customer\_search>  
 <add\_customer>Yes</add\_customer>  
 <edit\_customer\_card\_number>Yes</edit\_customer\_card\_number>  
 <print\_customer\_transactions>Yes</print\_customer\_transactions>  
 <edit\_customer\_info>Yes</edit\_customer\_info>  
 <customer\_visit\_add>Yes</customer\_visit\_add>  
 <customer\_visit\_redeem>Yes</customer\_visit\_redeem>  
 <customer\_visit\_delete>Yes</customer\_visit\_delete>  
 <customer\_visits\_view>Yes</customer\_visits\_view>  
 <manage\_cards>Yes</manage\_cards>  
 <manage\_fields>Yes</manage\_fields>  
 <list\_fields>Yes</list\_fields>  
 <import\_customers>Yes</import\_customers>  
 <import\_transactions>Yes</import\_transactions>  
 <api\_batch\_customers>Yes</api\_batch\_customers>  
 <api\_batch\_transactions>Yes</api\_batch\_transactions>  
 <view\_user>Yes</view\_user>  
 <edit\_user>Yes</edit\_user>  
 <delete\_user>Yes</delete\_user>  
 <add\_user>Yes</add\_user>  
 <edit\_users\_campaigns\_list>Yes</edit\_users\_campaigns\_list>  
 <edit\_my\_info>Yes</edit\_my\_info>  
 <show\_administrator\_option>Yes</show\_administrator\_option>  
 <show\_manager\_option>Yes</show\_manager\_option>  
 <show\_coaltion\_store\_manager\_option>Yes</show\_coaltion\_store\_manager\_option>  
 <show\_agency\_client\_import\_option>Yes</show\_agency\_client\_import\_option>  
 <show\_agency\_client\_option>Yes</show\_agency\_client\_option>  
 <show\_campaign\_manager\_option>Yes</show\_campaign\_manager\_option>  
 <show\_fulfillment\_contractor\_option>Yes</show\_fulfillment\_contractor\_option>  
 <show\_accountant\_option>Yes</show\_accountant\_option>  
 <show\_observer\_option>Yes</show\_observer\_option>  
 <access\_reports>Yes</access\_reports>  
 <export\_apex>Yes</export\_apex>  
 <export\_excel>Yes</export\_excel>  
 <export\_mailchimp>Yes</export\_mailchimp>  
 <make\_bookmarks>Yes</make\_bookmarks>  
 <affiliate>Yes</affiliate>  
 </user\_permissions>  
 </user>  
</response>  
  
  
  
CLERK API CALL:  
  
Data submitted to API at: <https://app.clienttoolbox.com/api.php>  
Array  
(  
 [user\_id] => nicole\_clerk  
 [user\_password] => 143f032b69aa3ba521eb703bb72b450212ec2a6a  
 [type] => user\_info  
 [account\_id] => nicole\_clerk  
)  
  
--- XML Result: (1352 bytes) ----  
  
<?xml version="1.0" encoding="utf-8" ?>  
<response status="success">  
 <user>  
 <user\_id>nicole\_clerk</user\_id>  
 <user\_is\_owner>false</user\_is\_owner>  
 <user\_first\_name>Nicole</user\_first\_name>  
 <user\_last\_name>Clerk Role</user\_last\_name>  
 <user\_addtl\_info>She likes cheese.</user\_addtl\_info>  
 <user\_language>EN</user\_language>  
 <user\_language\_custom>No</user\_language\_custom>  
 <user\_timezone>14</user\_timezone>  
 <user\_level>K</user\_level>  
 <user\_permissions>  
 <view\_account\_info>Yes</view\_account\_info>  
 <open\_campaign>Yes</open\_campaign>  
 <view\_promotions>Yes</view\_promotions>  
 <view\_rewards>Yes</view\_rewards>  
 <view\_buyx\_items>Yes</view\_buyx\_items>  
 <view\_buyx\_default>Yes</view\_buyx\_default>  
 <view\_depreciations>Yes</view\_depreciations>  
 <view\_recurring\_fee>Yes</view\_recurring\_fee>  
 <view\_customer\_search>Yes</view\_customer\_search>  
 <add\_customer>Yes</add\_customer>  
 <print\_customer\_transactions>Yes</print\_customer\_transactions>  
 <edit\_customer\_info>Yes</edit\_customer\_info>  
 <customer\_visit\_add>Yes</customer\_visit\_add>  
 <customer\_visit\_redeem>Yes</customer\_visit\_redeem>  
 <customer\_visits\_view>Yes</customer\_visits\_view>  
 <list\_fields>Yes</list\_fields>  
 <api\_batch\_customers>Yes</api\_batch\_customers>  
 <api\_batch\_transactions>Yes</api\_batch\_transactions>  
 </user\_permissions>  
 </user>  
</response>

1. Custom Fields

This screen is going to have identical functionality as the StickyStreet Saint 2 screen.  
  
Here is the link:  
<http://hwclients.com/rewardclix/custom_fields.html>  
  
This page has two tabs for both the Customer Fields and Transaction Fields that can be assigned to campaigns.  
  
Both Customer Fields and Transaction fields have the ability to add on as many custom fields as a merchant deems necessary for their campaigns.  
  
Both the Customer Fields and Transaction fields tabs have a "Add" button in which the custom fields can be created.   
  
If the fields are created under "Customer Fields", then those fields will only display under "Customer Fields".  
  
Customer Fields:  
These fields are pre-defined customer fields and are always available to be used:  
Card Number  
First Name  
Last Name  
Phone Number  
Email Address  
Birthday  
Address Line 1  
Address Line 2  
City  
State/Province  
Zip/Postal Code  
Country  
User Name  
Password  
PIN  
  
None of these fields are required to be used when creating a new customer in the StickyStreet system. They can be added and removed at any time.  
  
Card Number is the only pre-defined custom field that can have it's label changed to meet the needs of the merchant/business.    
  
In addition to the pre-defined fields, a merchant/business can create as many custom fields as necessary to store information.  
  
Any of the custom fields that have been created by a business/merchant can also be deleted at any time. Deleting a custom field has no affect on any future data.  
  
We will need to list all of the pre-defined customer fields and custom created customer fields on the tab labeled "Customer Fields"    
  
All of the custom created customer fields will include a delete button to the right of them.  
  
All of the custom fields, both standard and business/merchant created, have the ability to toggle off and on at any time by selecting a checkbox to the left of the field label. If the checkbox is checked on, then the field will show anywhere custom fields are shown throughout the dashboard. If they are unchecked, then the fields should not be shown on any other screens where custom fields can be shown.  
  
Here are the data types that can be used with the custom fields:  
Text, Date, List, Pick List  
  
Text - displays as a text field and allows any data entry  
Date - displays as a text field. Accepts dates in the following format by default: YYYY-MM-DD  
List - a text field that accepts a comma separated list. This list will then display on the main custom field screen in a select menu where one item can be chosen from the list  
Pick List - a text field that accepts a comma separated list. This list will then display on the main custom field screen as a list of options with checkboxes. Multiple items can be selected  
  
The screen to create these fields is located here:  
<http://hwclients.com/rewardclix/create_custom_fields.html>  
  
Here is the API that corresponds to the Customer Fields:  
<http://api.clienttoolbox.com/customer_fields_list.html>  
<http://api.clienttoolbox.com/customer_fields_add.html>  
<http://api.clienttoolbox.com/customer_fields_update.html>  
<http://api.clienttoolbox.com/customer_fields_delete.html>  
  
Transaction Fields:  
If the fields are created under "Transaction Fields", then those fields will only display under "Transaction Fields".  
  
We will need to list out any currently created "Transaction Fields" on this tab.  
  
Here are the data types that can be used with the custom fields:  
Text, Date, List, Pick List  
  
Text - displays as a text field and allows any data entry  
Date - displays as a text field. Accepts dates in the following format by default: YYYY-MM-DD  
List - a text field that accepts a comma separated list. This list will then display on the main custom field screen in a select menu where one item can be chosen from the list  
Pick List - a text field that accepts a comma separated list. This list will then display on the main custom field screen as a list of options with checkboxes. Multiple items can be selected  
  
Here is the API documentation to list the current fields:  
<http://api.clienttoolbox.com/transaction_fields_list.html>  
  
We will also include a button to the right of all the transaction fields that will allow the ability to delete the custom transaction field at any time.  
  
Here is the API documentation to delete any transaction field:  
<http://api.clienttoolbox.com/transaction_fields_delete.html>  
  
This is the screen that will create custom "Transaction Fields":  
<http://hwclients.com/rewardclix/create_custom_fields_transaction.html>  
  
On this screen, a business/merchant can create any new custom Transaction Field.  
  
Here is the API documentation to create a new custom Transaction Field:  
<http://api.clienttoolbox.com/transaction_fields_add.html>  
  
Here is information on how to hide and display the customer and transaction fields depending on whether or not they are checked on this page:  
  
When you request a list of custom fields (either customer or transaction, <http://api.stickystreet.com/customer_fields_list.html> or<http://api.stickystreet.com/transaction_fields_list.html> ) one of the XML nodes returned is the visibility setting: <show>Y</show>  
  
Based on that, the interface should show that data or not.  
  
In a screen where the custom fields are listed for those users with permissions to edit their settings, the full list should be shown, with a column indicating whether the field is visible or not. When a "edit" button is clicked, the following screen should show the available parameters, and the update sent using the custom fields' Update API calls (<http://api.stickystreet.com/customer_fields_update.html> or<http://api.stickystreet.com/transaction_fields_update.html>)

1. Quick Add Customer / Add New Customer

This screen is going to have identical functionality as the StickyStreet Saint 2 screen.  
  
Here is the link:  
<http://hwclients.com/rewardclix/add_new_customer.html>  
  
The button to this screen appears only in the "Participants" section of the dashboard.  
  
This screen allows a merchant/business the ability to quickly add a single user into the system at any given time.  
  
None of the information is required, but at least one field on this screen must be filled in when creating a new user. By default, let's make sure that we generate a card number when creating a user.  
  
There is a button next to the card number field that should generate a card number at random if needed.  
  
<http://api.clienttoolbox.com/customer_card_number.html>  
  
Each time a new customer is added, a unique customer ID is generated on the StickyStreet side to keep them unique to the database.  
  
The ability to add a new customer to any existing campaigns is also on this screen.  
  
Here is the API documentation for creating a new user in the system:  
<http://api.clienttoolbox.com/customer.html>  
  
Here is information from StickyStreet about having the system generate a card number for a customer:  
  
1) Using the Record Customer API call (record\_customer - <http://api.stickystreet.com/customer.html>) and passing the parameter "card\_number\_generate" and assigning to it the number of digits it should have (ex: "card\_number\_generate=8")  
  
2) Using the Customer - Generate Card # API call (generate\_card\_number - <http://api.stickystreet.com/customer_card_number.html> ) as an AJAX call in the background, triggered by a click of a button, which fills the card # text field that is next to it, just as in the current Dashboard.  
  
This supposes that the number of digits is hard-coded into the code -- In Dashboard, we just return a 16-digit number.  
Alternatively, the number of digits could be one of the parameters in a (local) interface settings screen, but that preference would have to be stored locally, since the API does not have a method to store a preferred "generate card #" length.   
  
  
Here is the API documentation for adding a user to a new campaign:  
  
A customer "belongs" to a campaign simply by having a transaction in that campaign. So to "Add" a customer to a particular campaign, you just need to record an zero-amount transaction with a description such as "Account activated." and iterate for each campaign selected.  
  
The reason behind the "belong-ness" is that in certain scenarios like a coalition account where each campaign is a different business, a user with access to only one campaign shouldn't be able to pull up customers that don't have transactions in their campaign, unless it's by an "ID" field (card\_number or any custom field marked "Is ID"). For example, Fred a manager at the Bakery store shouldn't be able to put in a zip code and get a list of all the customers in that zip code -- that would not be ethical and not legal due to privacy laws in some countries, unless they already have a relationship with his store (a previous transaction) -- however if the customer walks into Fred's store and presents the proper ID (card #, custom ID field) then Fred should certainly be able to pull them up by that identifier.

Sprint 2

1. Search For, View & Edit New Leads

This screen is going to have the majority of the same functionality as the "View All Customers" StickyStreet Saint 2 screen.  
  
The major difference is that on this screen, only users who are have been identified as "New Leads" in the StickyStreet system will display on this screen. We will need to query those users and only display them when this page is generated.  
  
Here is the link:  
<http://hwclients.com/rewardclix/view_all_new_leads.html>  
  
By default, we will display only 10 records on this page. But the ability to display more via a pull down menu in the top, top left corner exists to display records in increments of 10, 25, 50 and 100.  
  
Also, in the bottom right corner of the table is a pagination feature that allows someone to scroll through the table contents.  
  
This should all be included in the bootstrap functionality of the framework.  
  
In the table, we will be displaying the users card number, first name, last name, email address, date of last transaction, a button to view the individual customer's full details and the ability to edit the customer's details.  
  
We also must include functionality to allow a business/merchant to bulk delete users, if necessary, from this screen. The checkbox in the table header should automatically select all the records in the table. There is a button titled "Action" under the table that can be used to delete multiple customers from the system as well.  
  
Here is the API documentation that needs to be used to list all of the existing customers:  
<http://api.clienttoolbox.com/customer_find.html>  
  
This API call allows us to look across all fields and pull back all customer information.  
  
When the "Edit" button in the table is selected, the following screen will display:  
<http://hwclients.com/rewardclix/manage_customers_edit.html>  
  
On this screen, we should pre-fill all of the customer information that is currently in the system for that particular custom. This includes all of the custom fields that are checked/active in the system.  
  
None of this information is required and can be added or removed at any time.  
  
Here is the API information that should be referenced to query the customer's information:  
<http://api.clienttoolbox.com/customer_info.html>  
  
Here's the API information that should be referenced to update the customer's information on this screen:  
<http://api.clienttoolbox.com/customer.html>

1. Search For, View & Edit Advocates

This screen is going to have the majority of the same functionality as the "View All Customers" StickyStreet Saint 2 screen.  
  
The major difference is that on this screen, only users who are have been identified as "Advocates" in the StickyStreet system will display on this screen. We will need to query those users and only display them when this page is generated.   
  
Here is the link:  
<http://hwclients.com/rewardclix/view_all_advocates.html>  
  
By default, we will display only 10 records on this page. But the ability to display more via a pull down menu in the top, top left corner exists to display records in increments of 10, 25, 50 and 100.  
  
Also, in the bottom right corner of the table is a pagination feature that allows someone to scroll through the table contents.  
  
This should all be included in the bootstrap functionality of the framework.  
  
In the table, we will be displaying the users card number, first name, last name, email address, date of last transaction, a button to view the individual customer's full details and the ability to edit the customer's details.  
  
We also must include functionality to allow a business/merchant to bulk delete users, if necessary, from this screen. The checkbox in the table header should automatically select all the records in the table. There is a button titled "Action" under the table that can be used to delete multiple customers from the system as well.  
  
Here is the API documentation that needs to be used to list all of the existing customers:  
<http://api.clienttoolbox.com/customer_find.html>  
  
This API call allows us to look across all fields and pull back all customer information.  
  
When the "Edit" button in the table is selected, the following screen will display:  
<http://hwclients.com/rewardclix/manage_customers_edit.html>  
  
On this screen, we should pre-fill all of the customer information that is currently in the system for that particular custom. This includes all of the custom fields that are checked/active in the system.  
  
None of this information is required and can be added or removed at any time.  
  
Here is the API information that should be referenced to query the customer's information:  
<http://api.clienttoolbox.com/customer_info.html>  
  
Here's the API information that should be referenced to update the customer's information on this screen:  
<http://api.clienttoolbox.com/customer.html>

1. Add New Campaign

Here is the link to this screen:  
<http://hwclients.com/rewardclix/add_new_campaign.html>  
  
This screen will allow a merchant/business to create a new campaign.  
  
There are 5 types of campaigns that can be created:  
Points  
Buy X, Get 1 Free  
Gift Cards  
Event Based  
Earn Per Event  
  
On this screen, a business/merchant chooses which type of campaign they want to create and give it a name.  
  
After creating the campaign, they will then see a screen that is specific to each type of campaign.  
  
This is the Points Campaign configuration screen:  
<http://hwclients.com/rewardclix/configure_points_program.html>  
  
This is the Buy X, Get 1 Free Campaign configuration screen:  
<http://hwclients.com/rewardclix/configure_buy_x_get_free_campaign.html>  
  
This is the Gift Cards Campaign configuration screen:  
<http://hwclients.com/rewardclix/configure_gift_card_campaign.html>  
  
This is the Event Based Campaign configuration screen:  
<http://hwclients.com/rewardclix/configure_event_campaign.html>  
  
This is the Earn Per Event Based Campaign configuration screen:  
<http://hwclients.com/rewardclix/configure_earn_per_event_campaign.html>  
  
Here is the API documentation for creating a new campaign:  
<http://api.clienttoolbox.com/campaign_new.html>  
  
Here is the API documentation for setting up campaign rewards for either points campaigns or event based campaigns:  
<http://api.clienttoolbox.com/campaign_new_reward.html>  
  
Here is the API documentation for adding a promotion to points based campaigns:  
<http://api.clienttoolbox.com/campaign_new_promo.html>  
  
Here is the API documentation for adding a new item to a Buy X Campaign:  
<http://api.clienttoolbox.com/campaign_new_item.html>  
  
Here is the API documentation for setting up campaign depreciation:  
<http://api.clienttoolbox.com/campaign_new_depreciation.html>  
  
Here is the API documentation for setting up campaign fees:  
<http://api.clienttoolbox.com/campaign_new_fee.html>  
  
Here's the API documentation for listing out the rewards for a given campaign:  
<http://api.clienttoolbox.com/campaign_rewards.html>  
  
Here is the documentation for listing out the promotions for a given campaign:  
<http://api.clienttoolbox.com/campaign_promos.html>  
  
Here is the documentation for listing out the depreciations for a given campaign:  
<http://api.clienttoolbox.com/campaign_depreciations.html>  
  
Here is the documentation for listing out the fees of a given campaign:  
<http://api.clienttoolbox.com/campaign_fees.html>  
  
Here is the documentation for listing out the Buy X items for a given campaign:  
<http://api.clienttoolbox.com/campaign_buyx_list.html>  
  
Here is the documentation for deleting a reward from a given campaign:  
<http://api.clienttoolbox.com/campaign_delete_reward.html>  
  
Here is the documentation for deleting a promotion from a given campaign:  
<http://api.clienttoolbox.com/campaign_delete_promo.html>  
  
Here is the documentation for deleting a buy x campaign item from a given campaign:  
<http://api.clienttoolbox.com/campaign_delete_item.html>  
  
Here is the documentation for deleting a depreciation entry from a given campaign:  
<http://api.clienttoolbox.com/campaign_delete_depreciatn.html>  
  
Here is the documentation for deleting a fee from a given campaign:  
<http://api.clienttoolbox.com/campaign_delete_fee.html>

1. View All Campaigns

Here is the link to this screen:  
<http://hwclients.com/rewardclix/view_all_programs.html>  
  
This screen will simply display all of the current campaigns within StickyStreet that each merchant/business has created.  
  
Here is the StickyStreet API documentation:  
<http://api.clienttoolbox.com/campaigns_list_1.5.html>  
  
This screen will have 2 tables; Active Campaigns and Deactive Campaigns.  
  
These tables will show 10 records on the table by default. The html files include the ability to change the amount of rows to display in each table so that if someone needs to see more than 10 records, they can.  
  
The first table will need to display all of the campaigns that are currently listed as Active within the merchant/business' StickyStreet account.  
  
Clicking on an individual row in this table will take you to the screen to configure the type of campaign. The configuration screen will change based on the campaign type.  
  
This is the Points Campaign configuration screen:  
<http://hwclients.com/rewardclix/configure_points_program.html>  
  
This is the Buy X, Get 1 Free Campaign configuration screen:  
<http://hwclients.com/rewardclix/configure_buy_x_get_free_campaign.html>  
  
This is the Gift Cards Campaign configuration screen:  
<http://hwclients.com/rewardclix/configure_gift_card_campaign.html>  
  
This is the Event Based Campaign configuration screen:  
<http://hwclients.com/rewardclix/configure_event_campaign.html>  
  
This is the Earn Per Event Based Campaign configuration screen:  
<http://hwclients.com/rewardclix/configure_earn_per_event_campaign.html>  
  
  
This table has a bulk action button in which multiple campaigns can be changed to deactive status with a single click.  
  
Here is the API documentation to deactivate a campaign:  
<http://api.clienttoolbox.com/campaign_deactivate.html>  
  
The second table will need to display all of the campaigns that are currently listed/designated as Deactive within the merchant/business' StickyStreet account.  
  
Here is the specific API documentation to list the deactive campaigns:  
<http://api.clienttoolbox.com/campaigns_deactivated_list.html>  
  
Clicking on an individual row in this table will take you to the screen to configure the type of campaign even though it is deactivated.  
  
Clicking on an individual row in this table will take you to the screen to configure the type of campaign. The configuration screen will change based on the campaign type.  
  
This is the Points Campaign configuration screen:  
<http://hwclients.com/rewardclix/configure_points_program.html>  
  
This is the Buy X, Get 1 Free Campaign configuration screen:  
<http://hwclients.com/rewardclix/configure_buy_x_get_free_campaign.html>  
  
This is the Gift Cards Campaign configuration screen:  
<http://hwclients.com/rewardclix/configure_gift_card_campaign.html>  
  
This is the Event Based Campaign configuration screen:  
<http://hwclients.com/rewardclix/configure_event_campaign.html>  
  
This is the Earn Per Event Based Campaign configuration screen:  
<http://hwclients.com/rewardclix/configure_earn_per_event_campaign.html>  
  
This table has a bulk action button in which multiple campaigns can be changed to active status or completely deleted from the system with a single click.  
  
Here is the API documentation to reactivate a campaign:  
<http://api.clienttoolbox.com/campaign_reactivate.html>  
  
Here is the API documentation to delete a campaign:  
<http://api.clienttoolbox.com/campaign_delete.html>  
  
  
5- Edit Existing Campaign

This screen will allow a merchant/business to edit an existing campaign and is similar to the add new campaign screen. The major difference will be that the values for the campaign should prepopulate in the fields based on the campaign that is being edited.  
  
This is the Points Campaign configuration screen:  
<http://hwclients.com/rewardclix/configure_points_program.html>  
  
This is the Buy X, Get 1 Free Campaign configuration screen:  
<http://hwclients.com/rewardclix/configure_buy_x_get_free_campaign.html>  
  
This is the Gift Cards Campaign configuration screen:  
<http://hwclients.com/rewardclix/configure_gift_card_campaign.html>  
  
This is the Event Based Campaign configuration screen:  
<http://hwclients.com/rewardclix/configure_event_campaign.html>  
  
This is the Earn Per Event Based Campaign configuration screen:  
<http://hwclients.com/rewardclix/configure_earn_per_event_campaign.html>  
  
Here is the API documentation for creating a new campaign:  
<http://api.clienttoolbox.com/campaign_new.html>  
  
Here is the API documentation for setting up campaign rewards for either points campaigns or event based campaigns:  
<http://api.clienttoolbox.com/campaign_new_reward.html>  
  
Here is the API documentation for adding a promotion to points based campaigns:  
<http://api.clienttoolbox.com/campaign_new_promo.html>  
  
Here is the API documentation for adding a new item to a Buy X Campaign:  
<http://api.clienttoolbox.com/campaign_new_item.html>  
  
Here is the API documentation for setting up campaign depreciation:  
<http://api.clienttoolbox.com/campaign_new_depreciation.html>  
  
Here is the API documentation for setting up campaign fees:  
<http://api.clienttoolbox.com/campaign_new_fee.html>  
  
Here's the API documentation for listing out the rewards for a given campaign:  
<http://api.clienttoolbox.com/campaign_rewards.html>  
  
Here is the documentation for listing out the promotions for a given campaign:  
<http://api.clienttoolbox.com/campaign_promos.html>  
  
Here is the documentation for listing out the depreciations for a given campaign:  
<http://api.clienttoolbox.com/campaign_depreciations.html>  
  
Here is the documentation for listing out the fees of a given campaign:  
<http://api.clienttoolbox.com/campaign_fees.html>  
  
Here is the documentation for listing out the Buy X items for a given campaign:  
<http://api.clienttoolbox.com/campaign_buyx_list.html>  
  
Here is the documentation for deleting a reward from a given campaign:  
<http://api.clienttoolbox.com/campaign_delete_reward.html>  
  
Here is the documentation for deleting a promotion from a given campaign:  
<http://api.clienttoolbox.com/campaign_delete_promo.html>  
  
Here is the documentation for deleting a buy x campaign item from a given campaign:  
<http://api.clienttoolbox.com/campaign_delete_item.html>  
  
Here is the documentation for deleting a depreciation entry from a given campaign:  
<http://api.clienttoolbox.com/campaign_delete_depreciatn.html>  
  
Here is the documentation for deleting a fee from a given campaign:  
<http://api.clienttoolbox.com/campaign_delete_fee.html>

Sprint 3

1. Manage New Lead Detail Screen

Here is the screen:  
<http://hwclients.com/rewardclix/manage_new_leads.html>  
  
This screen appears after clicking on one of the rows on the "New Leads" table.  
  
The "Customer Information" box at the top of the screen contains a "Lead Score" which will be calculated via another portion of the dashboard and populate into that box, the customer's name, and up to 20 custom fields that the system will use to generate lead information.  
  
The default fields (for now) will be Phone Number, Email Address, Birthday, Address, Marital Status, Travel with Children, Own a Timeshare, Household Income, Referred by, Credit Score, Public Records, Revolving Trades, Collections, Installment Trades, Tradelines and Historic Negative Occurences.  
  
The ability to edit this information for the customer will be performed by clicking the "Edit Customer" button in this box.  
  
The second box on this page is Manage Customer Transactions. This is where all the transactions that a customer has made for a particular campaign will be recorded.  
  
On this screen, we show 5 tabs that outline each of the 5 types of campaigns that can be created.   
  
All of the campaigns that a user is actively assigned to should appear in the "Manage Customer Transactions" area.  
  
If it's a points campaign, then their points total should appear above the title "Points Campaign" in the tab.  
  
If it's a gift card campaign, the balance should appear above "Gift Card" in the tab.  
  
If it's a buy x get 1 free reward, then the amount of rewards they have available should appear above the title in the tab.  
  
If it's an earn per event campaign, the amount they have earned should appear above the title in the tab.  
  
And if it's an event based campaign, the total amount of activity should appear above the title in the tab.  
  
Here is the documentation for recording transactions within StickyStreet:  
<http://api.clienttoolbox.com/transaction_record.html>  
  
Here is the documentation for redeeming transactions within StickyStreet:  
<http://api.clienttoolbox.com/transaction_redeem.html>  
  
Here is the documentation for deleting transactions within StickyStreet:  
<http://api.clienttoolbox.com/transaction_delete.html>  
  
Finally, the table at the bottom of the screen called "Customer Activity" should display any activity that has occured on the customer's account.  
  
We should limit it to 10 rows at first glance. Then, like other tables throughout the dashboard, it will have pagination to display more information as needed.  
  
It will also include the ability to bulk delete activity if necessary via the "Action" button under the table.  
  
This may include undocumented API from StickyStreet. We will work on getting that information added here as soon as we can.

1. Manage Advocates Detail Screen

Here is the link to the screen:  
<http://hwclients.com/rewardclix/view_all_advocates.html>  
  
This screen appears after clicking on one of the rows on the "Manage Advocates" table.  
  
The "Customer Information" box at the top of the screen contains a "Lead Score" which will be calculated via another portion of the dashboard and populate into that box, the customer's name, and up to 20 custom fields that the system will use to generate lead information.  
  
The default fields (for now) will be Phone Number, Email Address, Birthday, Address, Marital Status, Travel with Children, Own a Timeshare, Household Income, Referred by, Credit Score, Public Records, Revolving Trades, Collections, Installment Trades, Tradelines and Historic Negative Occurences.  
  
The ability to edit this information for the customer will be performed by clicking the "Edit Customer" button in this box.  
  
The second box on this page is Manage Customer Transactions. This is where all the transactions that a customer has made for a particular campaign will be recorded.  
  
On this screen, we show 5 tabs that outline each of the 5 types of campaigns that can be created.   
  
All of the campaigns that a user is actively assigned to should appear in the "Manage Customer Transactions" area.  
  
If it's a points campaign, then their points total should appear above the title "Points Campaign" in the tab.  
  
If it's a gift card campaign, the balance should appear above "Gift Card" in the tab.  
  
If it's a buy x get 1 free reward, then the amount of rewards they have available should appear above the title in the tab.  
  
If it's an earn per event campaign, the amount they have earned should appear above the title in the tab.  
  
And if it's an event based campaign, the total amount of activity should appear above the title in the tab.  
  
Here is the documentation for recording transactions within StickyStreet:  
<http://api.clienttoolbox.com/transaction_record.html>  
  
Here is the documentation for redeeming transactions within StickyStreet:  
<http://api.clienttoolbox.com/transaction_redeem.html>  
  
Here is the documentation for deleting transactions within StickyStreet:  
<http://api.clienttoolbox.com/transaction_delete.html>  
  
Finally, the table at the bottom of the screen called "Customer Activity" should display any activity that has occured on the customer's account.  
  
We should limit it to 10 rows at first glance. Then, like other tables throughout the dashboard, it will have pagination to display more information as needed.  
  
It will also include the ability to bulk delete activity if necessary via the "Action" button under the table.  
  
This may include undocumented API from StickyStreet. We will work on getting that information added here as soon as we can.

1. Advance Search

Here is the link to this screen:  
<http://hwclients.com/rewardclix/customer_search.html>  
  
This screen allows a merchant to search for customers by as many or as little custom fields as they want to drill down into data.  
  
All of the Custom Fields that are active within the dashboard should populate in the "Search For" dropdown.  
  
The value that they are filtering by custom field should appear in the "Value" text box.  
  
The Add Field button should populate another "Search For" dropdown and "Value" text box. The merchant can add as many fields as they want to search for on this screen.  
  
A table should display the customer information after the "Run Report" button has been pressed.  
  
It should look like this screen:  
<http://hwclients.com/rewardclix/view_all_advocates.html>  
  
The information screen will also have a button that will export the data out as a .csv file.  
  
Here is the search customers API documentation:  
<http://api.clienttoolbox.com/customer_search.html>