Commissions Input, Reconciliation,

and Reporting System (CIRRS)

Vision Document

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 11/21/2016 | 1.0 | Public Revision 1 | Matt Kresha |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Project Background**

Rogers Benefit Group (RBG) is a general insurance agency (GA), and acts as a middleman between insurance carriers (suppliers) and insurance brokers (customers) who sell group health insurance and other ancillary products. RBG gets paid a percentage on all business (policies) it helps its brokers sell.

Each insurance carrier provides RBG with monthly spreadsheets as statement of payments that are made to RBG. Since these spreadsheets come from different sources, they are in a variety of disparate formats. Each carrier also uses different identifiers to identify what policy payment is for, and often the identifiers are neither entirely unique or consistent. This has been a major pain point for trying to reconcile commissions for sold policies.

The goal of this project is to provide a user-friendly system for commission statement input, reconciliation and reporting. This way non-technical staff can be used for monthly input, and commissions entries can be reported against on a regular basis.

**Required Skills**

**.NET/C#, Microsoft Dynamics CRM API, Regular Expressions**

**Analysis, Design, and Implementation**

The following document is meant to provide some informal analysis and design for the proposed system. Any analysis, design and implementation questions should be referred to the project manager, Matt Kresha.

The project will be divided into three milestones; these milestones will implement functionality the next milestone will be dependent upon. The following sections will detail informal uses cases for each milestone and user interface mockups for these use cases.

**Milestone I – Input Wizard Component**

**Milestone Background**

The first priority for processing commissions statements is inputting them into the CRM database. Each carrier provides a statement in different formats, typically an Excel spreadsheet. Some payment entries may include multiple rows, and some spreadsheets may have multiple worksheets that need to be imported. Not all payment entries are in first normal form (1NF) and not all the columns may have a one-to-one mapping with database attributes.

The goal of this first milestone is the creation of a user-friendly statement input and field mapping wizard. The most complex portion of this wizard will be the field mapping component. There will need to be a way to predefine and save field mappings templates for each carrier, as well as a means to manually change field mappings on the fly if needed. Since some columns may hold multiple attribute mappings, there will need to be the ability to create and save templates for mapping these multi-field columns (most likely using regular expressions).

This input wizard component should consist of the following tasks:

* Upload Task
* Field Mapping Task
* Confirmation Task

These tasks will be discussed in further detail below.

**Upload Task**

Users will be presented a page (Figure 1) for uploading a carrier statement spreadsheet with one or more worksheets containing commissions line item payment data. Each upload task will require the user browse for the statement file, select the carrier, select the charge to (statement payee) and whichever worksheets need to be imported into the CIRRS for processing; some worksheets may not be necessary depending on the statement. The CIRRS system should also be able to determine what the carrier and charge to fields should be default to by the filename and worksheets contained in the spreadsheet, but allow the user to modify these items as necessary.

The user will click the a “Next” button to proceed to the next step in the Input Wizard.

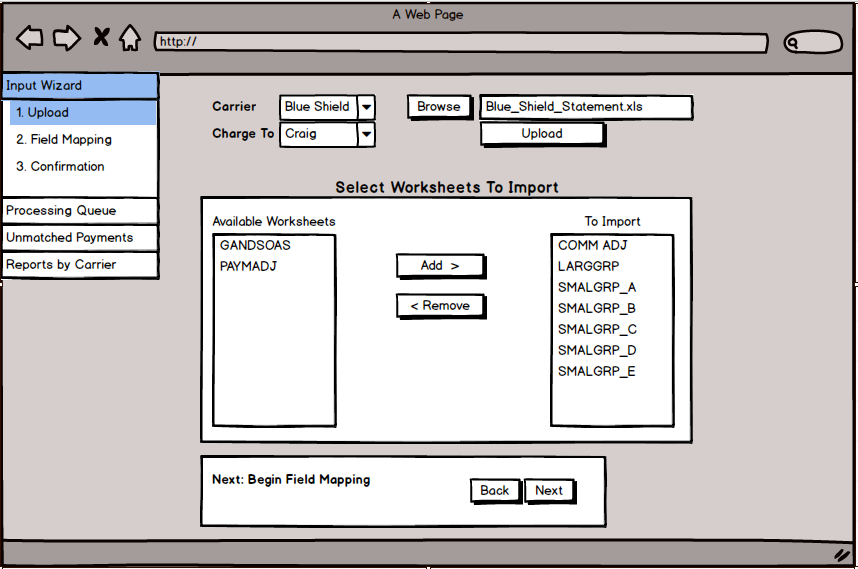
**UI Mockup**

Figure 1

**Field Mapping Task**

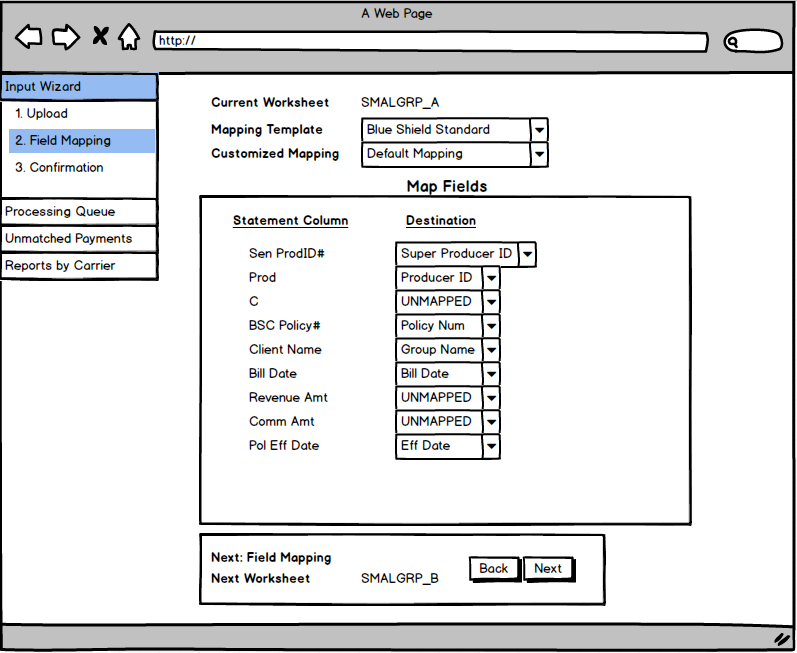
Users will then be presented a page for mapping the columns from the spreadsheet into the appropriate attributes within the database (Figure 2 and Figure 3). A mapping page for each worksheet should be presented to verify the mappings are correct. CIRRS should allow for stored mapping templates that can be selected to automatically map the columns; this can be user selected or automatic based on the carrier and worksheet names. The users can also select whether to use default (using the mapping template) or customized mapping for the current worksheet. If customized mapping is selected, the user can edit the destination attributes.

Some columns may not have a correlating attribute to map to and can remain “UNMAPPED”. Some columns may not be in first normal form (1NF) and if a mapping to a “MULTI-FIELD COLUMN” is picked, the user is given a secondary option to pick a multi-field template used to extract data from the multi-field column.

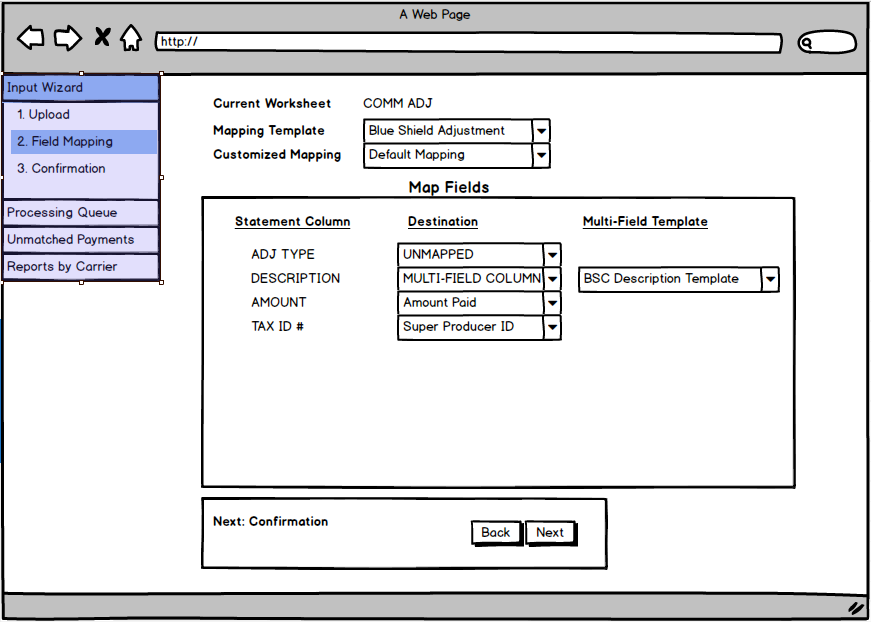
The user can click a “Next” button to proceed to the next step in the Input Wizard or a “Back” button to return to the previous step. The previous step may be another Field Mapping Task or the Upload Task depending on where in the overall Field Mapping Task the user is.

**UI Mockup**

**Figure 2**



**Figure 3**

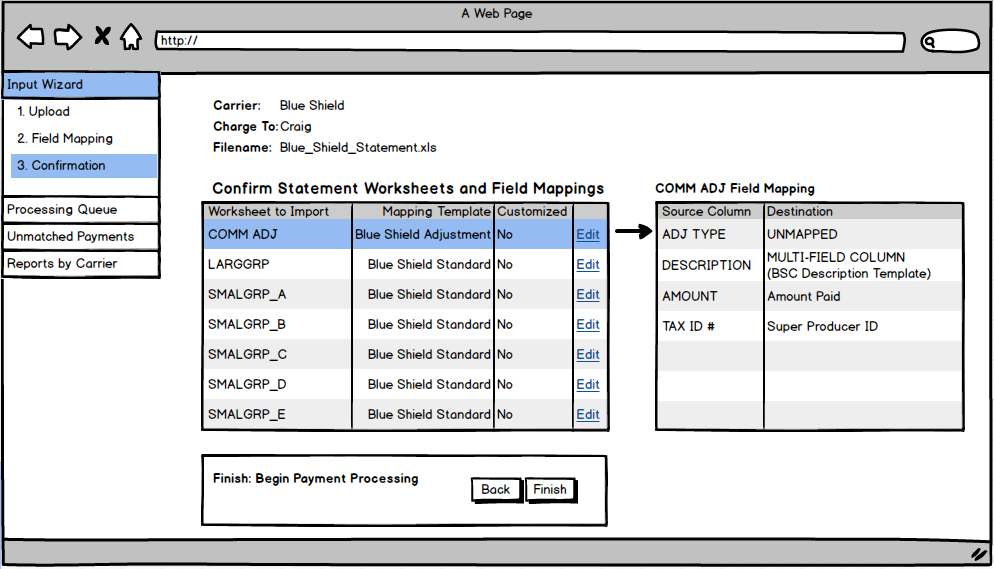


**Confirmation Task**

Users will be presented with a list of selected options for the overall spreadsheet import as well each individual worksheet. This list is for confirmation purposes to ensure incorrect mapping or other options were not selected. Once the user clicks the finish button to confirm the import, the selected worksheets will be queued for processing.

UI Mockup

**Figure 4**



**Milestone 2 – Processing Queue and Unmatched Payments Queue**

**Milestone Background**

After commissions spreadsheets have been input into the CRM database, they will be queued for processing. Processing involves matching each payment entry to sold policies within the CRM database. Due to inconsistent and in some cases non-unique identifiers used on the statements, matching payment entries can be problematic.

Payment processing and matching is the key to the success of this project. To accurately match payments to policies, a matching algorithm will need to be engineered. This algorithm will most likely need to output a matching confidence percentage, which will then be used to determine how likely the policy matched with the payment is to be correct.

**Processing Queue**

Lists the history of statements imported and any imported lists currently being processed by CIRRS. Each imported statement will list its status (processed/complete), count of payments matched within confidence threshold, count of payments outside of confidence threshold needing manual matching by a user. Each of the counts will link to a page listing these payments which will include each payment’s full details from the statement, the matching confidence (as a percentage), the matched policy information within the CRM (if matched within confidence threshold), as well as a lookup field to replace the matched policy with a different one.

**UI Mockup**

**TBD (Version 2.0)**

**Unmatched Payments Queue**

Same as above drill down page to payments falling outside of confidence threshold. This queue will list only these unmatched payments and will act as the main queue for users to perform updates to these unmatched payments.

**UI Mockup**

**TBD (Version 2.0)**

**Milestone 3 – Reports**

**Milestone Background**

Management will require some reports to be run based on the processed commissions entries. Some of these reports will be able to be run from the CRM, but other more complex reports may be needed. Reporting requirements have yet to be completely analyzed and will be included in later revisions of this document.

**TBD (Version 3.0)**