Software Design

Specification

Traffic Prediction  
*Revision 1.0*

**Table of Contents**

**Table of Contents ii**

**Revision History iii**

**Approved By iii**

**1.** **Introduction 1**

1.1 Purpose 1

1.2 System Overview 1

1.3 Design Map 1

1.4 Supporting Materials 1

1.5 Definitions and Acronyms 1

**2.** **Design Considerations 1**

2.1 Assumptions 1

2.2 Constraints 2

2.3 System Environment 2

2.4 Design Methodology 2

2.5 Risks and Volatile Areas 2

**3.** **Architecture 2**

3.1 Overview 2

3.2 Subsystem, Component, or Module 1 …N 2

3.3 Strategy 1…N 3

**4.** **Database Schema 3**

4.1 Tables, Fields and Relationships 3

4.1.1 Databases 3

4.1.2 New Tables 3

4.1.3 New Fields(s) 3

4.1.4 Fields Change(s) 3

4.1.5 All Other Changes 3

4.2 Data Migration 4

**5.** **High Level Design 4**

5.1 View / Model Element 1…N 4

**6.** **Low Level Design 4**

6.1 Module 1…N 4

**7.** **User Interface Design 4**

7.1 Application Controls 4

7.2 Screen 1… N 4

**Appendix A: Project Timeline 5**

**Revision History**

| **Version** | **Name** | **Reason For Changes** | **Date** |
| --- | --- | --- | --- |
| *1.0* | *Sue Dillehay*  *David Kurth*  *Jon Fletcher* | *Initial Revision* | *8/19/2004* |
|  |  |  |  |

**Approved By**

*Approvals should be obtained for project manager, and all developers working on the project.*

| **Name** | **Signature** | **Department** | **Date** |
| --- | --- | --- | --- |
| Bill Currie |  | BP-IT-Development |  |
|  |  |  |  |

# 1.**Introduction**

## **Purpose**

## **System Overview**

## **Design Map**

## Requirements

Requirements for the testing purpose of the project is on the basis of hardware and software aspects.

Hardware required for the project is a simple operating computing device.

Software Requirements -

* 1. Operating System -

The project can be tested on any computing device regardless of the operating system and the device used. Windows, Linux and MacOS based operating systems are most used for a computer .

* 1. Apache Server and Web browser

Apache server is needed to host and the project as a web application on a browser. Installation of Apache are posted on their main blog page. The project supports all web browsers. Hence, a web browser needs to be installed on the user’s computing device.

* 1. Node.js

The concerned software is built on the AngularJS framework of Node.js. Hence, Node.js needs to be installed on the tester’s device. The instruction manual is provided on the organization’s URL. Angular CLI is a framework that is employed to build this particular project and Angular CLI also needs to be installed on the device. Most commonly, the Node Package Manager is used to install the Angular CLI with a simple single command “npm install -g @angular/cli”

* 1. MySQL

MySQL is a free Relational Database Management System. It is based on the concept of “tables” which proves to be very helpful in visualization and representation of data. It has various constraints that links and tunes different data properly. Hence, MySQL is used.

* 1. Python

Python is a popular, powerful programming language. The project’s domain also has its roots in Deep Learning areas. Here the python language is employed for visualization, manipulation, processing, tuning, training the data for a predictive model and then predict on the same.

## **Definitions and Acronyms**

# **Design Considerations**

## **Assumptions**

## **Constraints**

None that we are aware of.

## **System Environment**

## **Design Methodology**

## **Risks and Volatile Areas**

# **Architecture**

## **Overview**

## **Subsystem, Component, or Module 1 …N**

## **Strategy 1…N**

# **Database Schema**

## **Tables, Fields and Relationships***.*

### **Databases**

### **New Tables**

### **New Fields(s)**

| **Table Name** | **Field Name** | **Data Type** | **Allow Nulls** | **Field Description** |
| --- | --- | --- | --- | --- |
| BinderRequest | SellingRep | Varchar(50) |  | Get this field from BankPlan.ProposalRepCode  But use actual name (Roy Pinnell) |
| BinderRequest | SigningRep | Varchar(50) |  | Get this field from Bankplan.SigningRepCode  But use actual name (Roy Pinnell) |
| ProcessBP | ParentProcessID | Int | Yes | This will tie a subprocess to a process |
| BinderRequest | ProjectedWireDate | Date |  | The earliest Policy.ProposalWireDate of all included policies in the scenario. |
|  |  |  |  |  |
|  |  |  |  |  |

### **Fields Change(s)**

*For each field change (such as data types, required/not required, or renaming), please complete a row of the following table. (Insert additional rows as needed.)*

| **Table Name** | **Field Name** | **What to change?** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

### **All Other Changes**

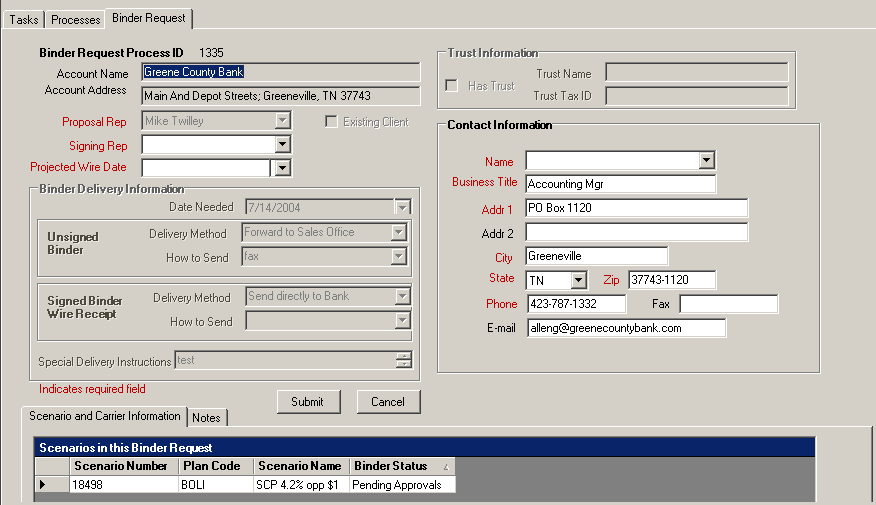
*If any other changes are requested (stored procedures, indexes, relationships, security settings, DTS packages, maintenance plans, etc), please describe what is needed here.*

## **Data Migration**

*(Optional) - Provide a description of how existing data should be migrated to new tables and fields.*

# **High Level Design**

## **Binder Request Form**



The notes tab should have fields displayed in the order Date|Author|Note.

Users will select a contact from a dropdown on the binder request form. If no binder contact exists for the current binder, the user will need to go to the Contacts module and add a binder contact.

Users will now have two places to click to save their binder requests. A new “Submit” button will be added to the binder request form.

Users will be able to delete binder requests as well. A new “Cancel” button will be added to the binder request form.

Automatic bank check-in will be removed. The user will be required to manually check in the bank before a binder request is submitted.

There will be three new fields on the binder request form. “Proposal Rep”, “Signing Rep”, and “Projected Wire Date” will be added to the form. Whenever a user selects email or fax as a delivery method, these fields must be set to ‘required’ in the Contacts area of the binder request form.

Account Manager will be added as a new delivery option for both the unsigned and signed delivery method.

## **User Interface Modifications**

Formatting modifications:

SUE –Will recheck the TRMs for Binder REQUEST.

Process modifications to the User Interface.

Make the task name subprocess reflect the delivery option name.

Null the task completion dates for each process at it’s creation . Teamplate will enter the completion date.

Automatically generate a note when binder request is created.

Add 30 day, 60 day, 90 day and 120 day tasklists for the user.at the menu level.

## **Workflow sub-processes**

Workflow subprocesses will be created for each individual binder in each binder request. The Teamplate Model that will be used to create the subprocess is called BinderPerCarrier.

Binder Extension subprocesses will be created each time an extension is needed for each binder. The Teamplate Model name used to create the subprocess is called BinderExtension.

# **Low Level Design**

*This section provides low-level design descriptions that directly support construction of modules. Normally this section would be split into separate documents for different areas of the design.*

## **Binder Request Form**

### **Contact Changes**

The format of the name in the drop down will be “firstname lastname”. Only ‘Binder Request’ contacts will be displayed. If more than one Binder Contact exists for this bank, initially all the Contact information will be filled in with the information from the Binder Contact that is listed as Primary in the contacts module. If no Binder Contact exists for this bank, then no Contact information will be filled in. The user will then need to go to the Contacts module to add a Binder Contact Type and Save the new Contact. NOTE: It is very important that the user remembers to “Save” the Binder Contact in the Contacts Module. Otherwise, the Binder Request Form will not be able to populate the Contact information.

Any time the user returns to the binder request node before the binder request has been saved, the system will refresh the contact name dropdown with the latest contact information for the bank.

Any Contact data that is entered on the binder request screen will be saved to the BinderRequest table and only valid for that binder. The Contact information saved with that Binder will not be saved to the account Contact’s information. If a user wishes to change the data for a contact, they must do that in the Contact module

### **Submit Button**

The submit button will only be enabled for those users that are in the BinderRequestCreate group. Once the user has entered all the information for the binder request, they will press “Submit” and it will submit the binder request. All fields marked in red will be validated before submission. The “Submit” button will work function like the ‘Save’ button on the toolbar.

### **Cancel Button**

The “Cancel” button will only be enabled for users who belong to the BinderRequestApprove group. This button will only be enabled and used one the binder request is passed the “Approved” task. The user will be prompted “Are you sure you would like to cancel this binder request”. The cancellation will delete all Teamplate processes associated with the binder request and moark the policy groups as “Not Taken”. There is currently a stored procedure called spRemoveProcess that deletes from the correct tables. This stored procedure should be used when removing a process. It will need to be modified to handle deleting the new subprocesses spawned from the main process.

?????? What do we want to do with the binder status (in regards to the binder process report of binder status)

### **Automatic Bank Check-in**

There will no longer be an automatic bank check-in when the binder request is submitted. Before a binder request is saved, the user must check in a bank. If the bank is not checked in, a message will be displayed indicating to the user that they must first check in the bank before creating a binder request. There is a column located in the bank table called checkout that can be used to determine if the bank is still checked out.

### **Other Changes**

This sub-section will group together some of the other small changes with the Binder Request form.

Whenever a user selects e-mail or fax as a delivery method, the e-mail or fax fields located in the Contact area will be required by the user to fill in before they are allowed to save. Change the labels to display in red so that the user knows this is required field.

The binder request form will have three new fields. These fields will be read only and will be populated by data stored in TOPS.

## **Workflow sub-processes**

Once the Binder Request has been approved, the Binder Request process will spawn off 1 to n child processes (BinderRequestPerCarrier). There will be one child process per carrier (or policy group). There could be more than one child process per carrier if the product or interest rate is different. The child processes will be connected to the main process via the ProcessBP table. Since each process is inserted into this table, we will add a parentprocessid column to the table. For each process that is inserted into the processBP table, the parentprocessid will be the process ID of the main process.

### **Binder Extension**

*A Binder Extension Process will be created for a Binder if the “Submit Case to Carrier” step for that Binder has not been completed within 5 days of the Wire Expiration Date. A long running Windows Service will need to be created and run daily to find any Binders that need an extension. This service will need to check the due date of the “Submit Case to Carrier” task in Teamplate of each Binder and see if the current date is within the 5 days. If it is, the windows service will need to start the Binder Extension process. A row will need to be inserted into the ProcessBP table for the new process and the parentprocessid column filled in with the process id of the binder the extension was created for. The process type column in the ProcessBP table will be populated with the value “BinderExtension”.*

*The following query will find Binders that need to have a Binder Extension process created for them that currently do not:*

*SELECT ProcessID, DueDate FROM ProcessBP*

*INNER JOIN TeamPlate..Process TP ON ProcessBP.ProcessID = TP.ID*

*INNER JOIN TeamPlate..Task TPT ON TP.ID = TPT.PID*

*WHERE ProcessType = 'BinderRequest'*

*AND TP.Status <> 'Complete'*

*AND TPT.Name = 'Submit Case to Carriers'*

*AND TPT.DueDate <= dateAdd( day, 5, getdate())*

*AND ProcessID NOT IN ( SELECT parentProcessID from ProcessBP WHERE ProcessType = ‘BinderExtension’)*

*The processID from the query will be used to populate the parentprocessId column in the processBp table. When the Binder Extension process is created it calls the CreateProcessBp located in WorkFlowCommon. This function will need to get updated to handle another parameter called parentprocessID. This parameter will need to be passed in when the Binder Extension is created and when each individual Binder is created. When modifying this function, please keep in mind that the first phase of the Binder request calls this as well as the sub process Binder Delivery in Teamplate. These will need to be modified as well.*

*When the task tskSubmitCaseToCarrier in a Binder process is advanced and asked to complete, it will first need to check and make sure that there is no Binder Extension that was created for that Binder. The following SQL will find if there is a Binder Extension for that Binder:*

*SELECT ID FROM Teamplate..Process TP*

*WHERE Status <> ‘Complete’*

*AND ID IN ( SELECT ISNULL(ProcessID,0) as processID FROM ProcessBP*

*WHERE parentProcessID = xxxx*

*AND ProcessType = ‘BinderExtension’ )*

*Where xxxx is the processID of the Binder we are checking for. If the query returns an ID greater than 0, then we know that there is a Binder Extension process that still has not completed. If this is the case, the Binder will NOT be allowed to complete. This code will be called from a consumed web service on the execute event of the task tskSubmitCaseToCarrier of the Binder.*

*If the Binder Extension’s last task reaches it’s due date, then the Binder Extension needs to be reworked and start back at the beginning step.*

# **User Interface Design**

*This section provides user interface design descriptions that directly support construction of user interface screens.*

## **Application Controls**

*Detail the common behavior that all screens will have. Common look and feel details such as menus, popup menus, toolbars, status bar, title bars, drag and drop mouse behavior should be described here.*

## **Screen 1… N**

*Illustrate all major user interface screens and describe the behavior and state changes that the user will experience.*

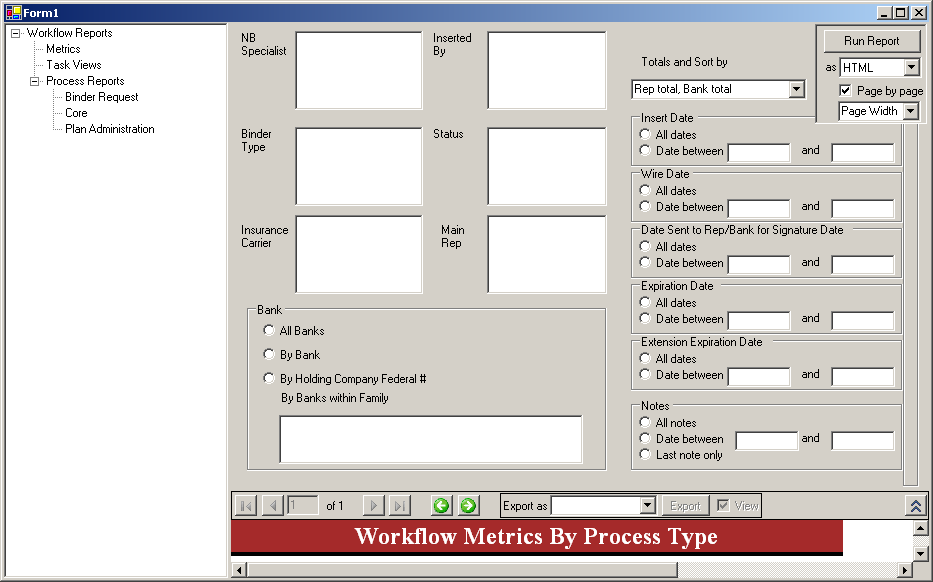
*A screen transition diagram or table can optionally be created to illustrate the flow of control through the various screens.*

### **Workflow Reports**

??? Are these the correct parameters for this report???

Add calendar control next to all dates.

Remove the Binder Status report and the Binder Extension Report from BPS..



The following is a breakdown of the values in the above Workflow Reports screen:

| **Label Name** | **Note** | **Source** |
| --- | --- | --- |
| NB Specialist |  | SELECT UWUserID FROM PolicyGroup GROUP BY UWUserID ORDER BY PolicyGroup.UWUserID; |
| Inserted By | Change to be a dropdown containing the NB Cordinators (approve group).  Add to the BinderRequest table a new filed ‘NBCordinator’ | SELECT InsertBy FROM PolicyGroup GROUP BY InsertBy ORDER BY InsertBy;  Query active directory to return the NBCordinator group. |
| Binder Type |  | SELECT CodeToText.Code, CodeToText.Text FROM CodeToText WHERE (((CodeToText.TableDotField)='PolicyGroup.Type')) ORDER BY CodeToText.SortOrder; |
| Status | This will not be used. | SELECT PolicyStatus.Status, PolicyStatus.PolicyGroupApply FROM PolicyStatus WHERE (((PolicyStatus.PolicyGroupApply)<>0)) ORDER BY PolicyStatus.Status; |
| Insurance Carrier |  | SELECT InsCo.InsCo, InsCo.CompanyName FROM InsCo ORDER BY InsCo.InsCo, InsCo.CompanyName; |
| Main Rep | Will this be the ***selling*** or ***signing*** rep? | SELECT Rep.RepCode, [LName]+", " & [FName] AS Name, Rep.RepCode AS AcctgRepCode FROM Rep ORDER BY [LName]+", " & [FName]; |
|  |  |  |

**Appendix A: Project Timeline**

*Reference the Microsoft project Binder Request Release 2 – Development..*