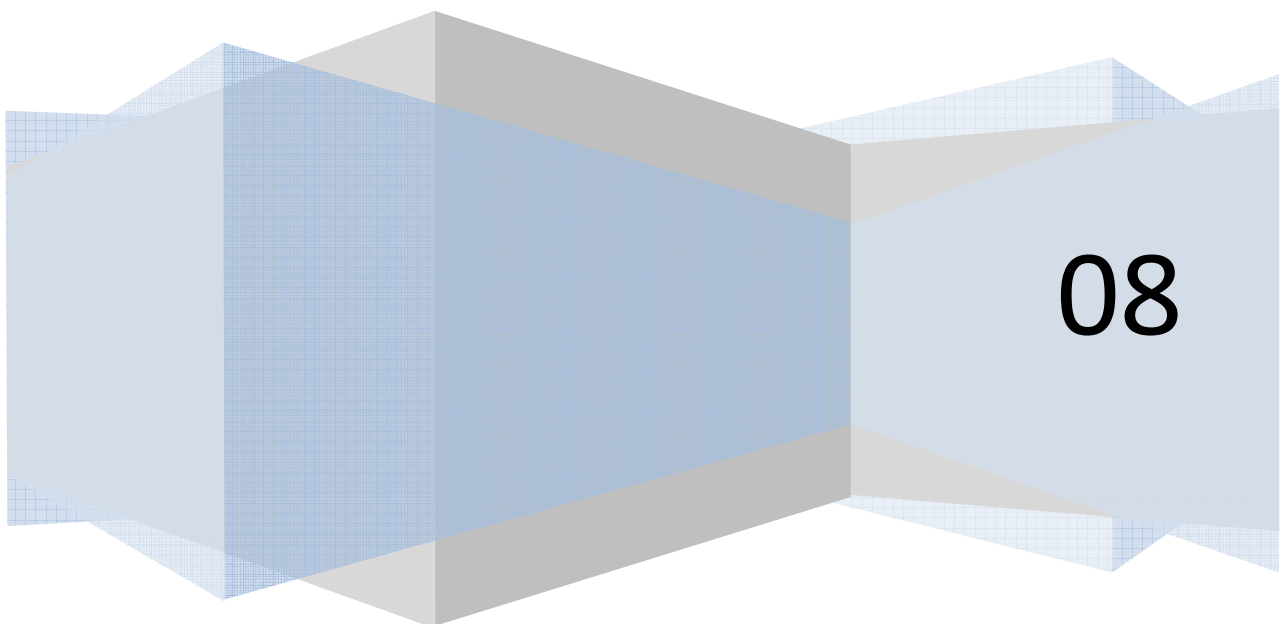


Functional Specification

eClosing System

Safedocs[™]





**Functional Requirements Document
Authorization Memorandum**

I have carefully assessed the Functional Specification Document for the Safedocs eClosing System. This all inclusive 41 page document has been completed in accordance with the specified requirements of the DSI – Safedocs agreement.

MANAGEMENT CERTIFICATION - Please check the appropriate statement.

_____ The document is accepted.

_____ The document is accepted pending the changes noted.

_____ The document is not accepted.

We fully accept the changes as needed improvements and authorize initiation of work to proceed. Based on our authority and judgment, the continued operation of this system is authorized.

Michael Morford

Project Leader

DATE

eClosing - Functional Specification 1.0

Contents

1	INTRODUCTION	7
1.1	POINTS OF CONTACT	7
2	DIAGRAMS	7
2.1	eCLOSING PROCESS – SMART NOTE	7
3	SYSTEM ADMINISTRATION	8
3.1	USER ROLES	8
3.2	LOGIN	9
3.2.1	<i>Behavior</i>	9
3.2.2	<i>Fields</i>	10
3.3	LOGOUT	10
3.3.1	<i>Behavior</i>	10
3.4	CREATE COMPANY	10
3.4.1	<i>Behaviors</i>	10
3.4.2	<i>Fields</i>	11
3.4.3	<i>Security</i>	11
3.5	CREATE BRANCH	11
3.5.1	<i>Behaviors</i>	11
3.5.2	<i>Fields</i>	12
3.5.3	<i>Security</i>	12
3.6	CREATE SYSTEM ADMINISTRATOR USER	12
3.6.1	<i>Behavior</i>	12
3.6.2	<i>Fields</i>	12
3.6.3	<i>Security</i>	13
3.7	CREATE COMPANY ADMIN USER	13
3.7.1	<i>Behaviors</i>	13
3.7.2	<i>Fields</i>	13
3.7.3	<i>Security</i>	13

3.8	CREATE FUNDER	14
3.8.1	Behaviors.....	14
3.8.2	Fields.....	14
3.8.3	Security.....	14
3.9	CREATE BRANCH USER	14
3.9.1	Behaviors.....	14
3.9.2	Fields.....	15
3.9.3	Security.....	15
3.10	CREATE INVESTOR	15
3.10.1	Behaviors	15
3.10.2	Fields.....	15
3.10.3	Security	16
3.11	CREATE ESCROW	16
3.11.1	Behaviors	16
3.11.2	Fields.....	17
3.11.3	Security	17
3.12	CREATE ESCROW CONTACTS.....	17
3.12.1	Behaviors	17
3.12.2	Fields.....	17
3.12.3	Security	18
3.13	CREATE RECORDER	18
3.13.1	Behaviors	18
3.13.2	Fields.....	18
3.13.3	Protocols.....	19
3.13.4	Security.....	19
3.13.5	Open issue	19
3.14	CREATE eVAULTING COMPANIES	19
3.14.1	Behaviors	19
3.14.2	Fields.....	19
3.14.3	Protocols.....	20
3.14.4	Security.....	20
3.15	CREATE REMOTE SIGNING SERVICE COMPANIES (RSSC)	20
3.15.1	Behaviors	20
3.15.2	Fields.....	20
3.15.3	Security.....	21
3.16	MY ACCOUNT.....	21
3.16.1	Behaviors	21
3.17	DELETE COMPANY	21
3.17.1	Behaviors	21
3.17.2	Security.....	21
3.18	DELETE BRANCH	21
3.18.1	Behaviors	21
3.18.2	Security.....	22
3.19	DELETE USER.....	22
3.19.1	Behaviors	22
3.19.2	Security.....	22

3.20	ADD COMPANY RESOURCE.....	22
3.20.1	Behaviors.....	22
4	CREATING ECLOSING PACKAGES.....	23
4.1	STEP 1: LOAN INFORMATION.....	23
4.1.1	Behaviors.....	23
4.1.2	Fields.....	23
4.1.3	Security.....	24
4.2	STEP 2: ADD BORROWERS.....	24
4.2.1	Behaviors.....	24
4.2.2	Fields.....	24
4.2.3	Security.....	25
4.3	STEP 3: ADD DOCUMENTS.....	25
4.3.1	Behaviors.....	25
4.3.2	Fields.....	25
4.3.3	Security.....	25
4.4	STEP 4: ADD NOTES.....	25
4.4.1	Behaviors.....	25
4.4.2	Fields.....	25
4.4.3	Security.....	26
4.5	FINISHING CREATE ECLOSING PACKAGE.....	26
4.5.1	Behaviors.....	26
5	PACKAGE MANAGEMENT.....	26
5.1	WORKFLOW.....	26
5.1.1	States.....	27
5.2	PACKAGES MODULE.....	31
5.2.1	Behaviors.....	31
5.3	OFFLINE SIGNING.....	31
5.3.1	Fields.....	32
6	BORROWER PACKAGE MANAGEMENT.....	32
6.1	BORROWER LOGIN.....	32
6.1.1	Behaviors.....	32
6.1.2	Fields.....	32
6.1.3	Security.....	33
6.2	BORROWER VIEWER.....	33
6.2.1	Behaviors.....	33
6.2.2	Signature Types.....	33
6.2.3	Open Issues.....	34
6.3	NOTARIZATION.....	34
6.3.1	Behaviors.....	34
7	INTERFACES TO EXTERNAL SYSTEMS – WEB SERVICES.....	35
7.1	VERSIONING & EXTENSIBILITY.....	35
7.2	XML MESSAGE VERSIONING.....	35

7.3	WEB SERVICE DATA REQUIREMENTS	36
7.4	CLIENT PROCESSING REQUIREMENTS.....	37
7.5	REFERENCES.....	37
7.6	SAFEDOCS ECLOSING PARTNER WEB SERVICE	37
8	AUDITING.....	38
9	REPORTS	38
	9.1.1 <i>Open Issues</i>	39
10	EXTERNAL ECLOSING TOOLS.....	39
11	DOCUMENT REFERENCES	39
12	APPENDIX A.....	39

1 Introduction

The project is an eClosing solution developed for DSI by Safedocs to allow a complete end-to-end all inclusive SMART Doc loan closing workflow to happen. Closing documents will be placed into the system to become available to all parties involved in the process. After executed, documents will be archived, registered and recorded to meet industry requirements.

This functional specification document (FSD) is a formal component of the original Safedocs-DSI's eClosing's product agreement and will serve as a functional guide to the developers. The developers agree to provide the capabilities specified. The client agrees to find the product satisfactory if product provides the capabilities specified in this FSD.

1.1 Points of Contact

The contact participants in this project are:

Contact	Role	Email
Terry VanBibber	Business Consultant/Project Overseer - Safedocs	tvb@sfdocs.com
Fernanda Azevedo	Lead Contact/Project Manager – Safedocs	fernanda@sfdocs.com
Michael Morford	Lead Contact/Project Manager – DocMagic	mikem@docmagic.com

2 Diagrams

2.1 eClosing Process – SMART Note

eClosing Package contains SMART documents. The Smart Note and Deed of Trust will be electronically (e-signed by borrower and notary), registered with MERS eRegistry. SMART scenario is explained in Figure 1.

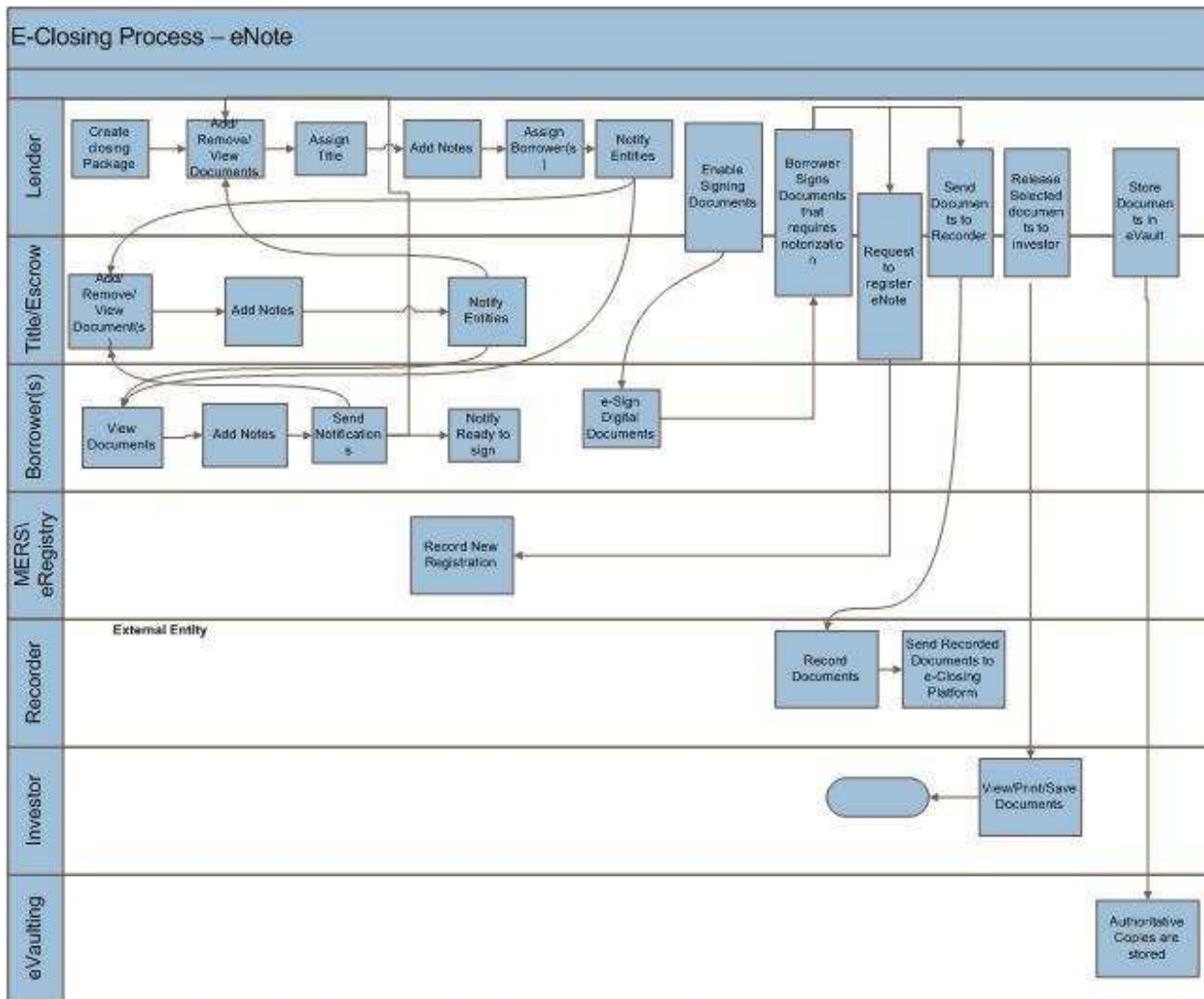


Figure 1. SMART eClosing Process

3 System Administration

This module deals with all the entity management. These entities are the ones used to support the eClosing process (i.e., Company, Branch, Users, etc.). This section will define functionality pertaining to the Administrative module. It also describes the user roles that exist in the system.

3.1 User Roles

The system supports the following user roles: System Administrator (Sys Admin), Company Administrator (Company Admin), Branch User, Funder, Escrow and Investor.

Role	Description
System Administrators	A System Administrator can create and manage companies, branches, users, recorders, escrow and resources. This type of user has also access to eClosing

	<p>Activity Report, which shows the total number of eClosing packages in a period (for more details, see Report chapter).</p> <p>This role does not allow eClosing package management</p>
Company Administrators	Company Administrators can create and manage branches, users and resources for the associated company. They can also access reports for the associated company.
Branch Users	<p>A branch user can be:</p> <p>Branch user admin (Branch Admin). He/She can only create other branch users in the system.</p> <p>Regular branch user. He/She can create and manage eClosing packages and access reports for the associated branch.</p>
Funders	Contact user at Company level, so it can be seen by all branches. Funders can access documents made available to them. They have no access to packages unless it is specifically assign to them.
Escrow	Escrow users belong to an Escrow Company in the system. They can view and modify eClosing packages. They can also allow the borrower to sign documents. Escrow may also be referred as Title company/Attorney.
Investor	Investor (also called controller or holder) is the entity/person that has control of the eNote (so named on the MERS® eRegistry) and its authoritative copy, or the original paper note.

3.2 Login

Login functionality is the entry point of the system. Based on login information and password the user is authenticated and his role is identified.

3.2.1 Behavior

1. The user login should be an email address and unique in the system
2. The user login should not be case sensitive
3. The password should be stored encrypted and compared with the one entered by the user
4. The user will get access to the system if login and password match with the data stored.
5. If user is associated to more than one account, system should present available options and let user select where to go.

6. The system should redirect the user to the specific module based on his/her credentials.
 - a. If the user is a Sys Admin, the system should redirect user to the Administrative module
 - b. If the user is a Company Admin, the system should redirect user to the Company Administrative module.
 - c. If the user is a Branch Admin, the system should redirect user to the Branch Administrative module.
 - d. If the user is a Branch user, the system should redirect user to the eClosing Package Management module.
 - e. If the user is an Escrow user, the system should redirect user to the eClosing Package Management module.
 - f. If the user is a Funder, the system should redirect user to the Funder View Package module.
 - g. If the user has multiple roles/types, the system should make all the options for all the modules associated with the user available.
7. The login screen should have “a forgot password” link. When clicked the user will be required to enter email (login information). The password will be emailed to the user.
8. When the user is logged, the system should keep a timeout for inactivity. If the user does not access the system, the session times out and the system warns the user that the session expired and presents the option to go back to the login screen.

3.2.2 Fields

Field	Description	Required	Type
Login	User identification previously created.	Yes	String
Password	Password previously defined.	Yes	String
Branch ID	Internal identification number of the branch.	Yes	Integer
User Role			

3.3 Logout

Log the user out of the system.

3.3.1 Behavior

1. The user session should be erased.
2. The system should redirect to the login screen.

3.4 Create Company

Creates a company in the system. A company in the system maps or represents the Lender entity.

3.4.1 Behaviors

1. The user must log in the system as a Sys Admin in order to have access to create company.

2. State field should be a dropdown
3. When user saves the company the system should validate
 - a. All the required fields
 - b. Email, phone and zip code format.

3.4.2 Fields

These fields are used in the company creation:

Field	Description	Required	Type
CompanyID	Existing ID in the Doc Magic's system. Unique.	Yes	Integer
CompanyName	Name of the company being registered	Yes	String
CompanyAddress	Mailing address of the company	Yes	String
CompanyCity	City of the company	Yes	String
CompanyState	State of the company	Yes	String
CompanyZipCode	Zip Code of the company	Yes	String
CompanyPhoneNumber	Phone number of the company	No	String
ContactEmail	Email address for the company	No	String

3.4.3 Security

Only users with System Admin role are able to create company.

3.5 Create Branch

Creates a branch in the system. A branch in the system maps or represents a physical branch entity (i.e. the branch of a bank). The system will be capable of creating branches on a transaction basis (when using Web services – automated process). If Branch data does not exist in the worksheet, DocMagic will send the LENDER information to be used to create the Default Branch.

3.5.1 Behaviors

1. The user must log in as a Sys Admin or Company Admin in order to create a branch
2. State field should be a dropdown
3. When user saves the company the system should validate
 - a. All the required fields
 - b. Email, phone and zip code format.

3.5.2 Fields

These fields are used in the branch creation:

Field	Description	Required	Type
Branch ID	Internal identification number of the branch	yes	Integer
CompanyID	ID of the selected company the new branch is associated with	Yes	Integer
BranchNumber	Number of the branch	Yes	String
BranchName	Name of the branch being registered	Yes	String
BranchAddress	Mailing address of the branch	Yes	String
BranchCity	City of the branch	Yes	String
BranchState	State of the branch	Yes	String
BranchZipCode	Zip Code of the branch	Yes	String
BranchPhoneNumber	Phone number of the branch	No	String
BranchFax	FAX number of the branch	No	String
ContactEmail	Contact e-mail address for the branch	No	String

3.5.3 Security

Only users with System Administrator and Company Administrator roles are able to create branches.

3.6 Create System Administrator User

System Administrator users are a system wide entity and have the ability to manage users, companies, branches, escrow, recorders and Investors. It is the highest role level in the system. However Sys Admin is not allowed to manage closing packages in the system.

3.6.1 Behavior

1. Log in the system as Sys Admin. Only Sys Admins can create other System Administrator users.
2. System Administrators manage all system entities but they are not allowed to manage closing packages.
3. When the values are submitted the system should validate:
 - a. All the required fields
 - b. Email, phone format.
4. When the user clicks the save button the new System Administrator is created and it is ready to be used. An email with new credentials is sent to the user to notify about new account.

3.6.2 Fields

These fields are used in the System Administrator user creation:

Field	Description	Required	Type
UserRole	Assigned automatically by the system.	Yes	Integer

	SYS ADMIN		
UserLogin	Login entered by the user; must be an email address; must be unique.	Yes	String
Password	Password entered by user	Yes	String
User First Name	User's first name	Yes	String
User Last Name	User's last name	Yes	String
UserPhoneNumber	Phone number of the user	No	String
UserEmail	E-mail address for the user.	No	String

3.6.3 Security

Only users with Sys Admin role are able to create other Sys Admin users.

3.7 Create Company Admin User

Creates a Company Admin.

3.7.1 Behaviors

1. The user logs in as Sys Admin or Company Admin
2. When the values are submitted the system should validate:
 - a. All the required fields
 - b. Email, phone format.
3. When the user clicks the save button the new Company Administrator is created and it is ready to be used. An email with new credentials is sent to the user to notify about new account.

3.7.2 Fields

These fields are used in the Company Admin creation:

Field	Description	Required	Type
CompanyID	ID of the selected company	Yes	Integer
UserRole	Assigned automatically by the system. COMPANY ADMIN	Yes	Integer
UserLogin	Login entered by the user; must be an email address; must be unique.	Yes	String
Password	Password entered by user	Yes	String
User First Name	User's first name	Yes	String
User Last Name	User's last name	Yes	String
UserPhoneNumber	Phone number of the user	Yes	String
UserEmail	Email address for the user	Yes	String

3.7.3 Security

Only users with Sys Admin or Company Admin role are able to create a Company Admin.

3.8 Create Funder

Creates a Funder. A Funder may have access to packages across all branches that belong to the same company. A funder is used to review loans for funding.

3.8.1 Behaviors

4. The user logs in as Sys Admin or Company Admin
5. When the values are submitted the system should validate:
 - a. All the required fields
 - b. Email, phone format.
6. When the user clicks the save button the new Funder is created and it is ready to be used. An email with new credentials is sent to the user to notify about new account.

3.8.2 Fields

These fields are used in the Funder creation:

Field	Description	Required	Type
CompanyID	ID of the selected company	Yes	Integer
UserRole	Assigned automatically by the system. FUNDER	Yes	Integer
UserLogin	Login entered by the user; must be an email address; must be unique.	Yes	String
Password	Password entered by user	Yes	String
User First Name	User's first name	Yes	String
User Last Name	User's last name	Yes	String
UserPhoneNumber	Phone number of the user	Yes	String
UserEmail	Email address for the user	Yes	String

3.8.3 Security

Only users with Sys Admin or Company Admin role are able to create a Company Admin.

3.9 Create Branch User

Creates a Branch User.

3.9.1 Behaviors

1. The user logs in as Sys Admin , Company Admin or Branch Admin
2. All required fields should be validated and well formats for phone, email and login.
3. If UserType is admin, the branch user can only create other users, else the regular user can only manage eClosing packages.
4. When the user clicks the save button the new Branch User is created and it is ready to be used. An email with new credentials is sent to the user to notify about new account.

3.9.2 Fields

These fields are used in the Branch User creation:

Field	Description	Required	Type
CompanyID	ID of the selected company	Yes	Integer
BranchID	ID of the selected branch	Yes	Integer
UserRole	Assigned automatically by the system. BRANCH USER	Yes	Integer
UserLogin	Login entered by the user. This login must be unique in the system; however a branch user may be assigned to more than one branch under the same company.	Yes	String
Password	Password entered by user	Yes	String
User First Name	User's first name	Yes	String
User Last Name	User's last name	Yes	String
UserPhoneNumber	Phone number of the user	Yes	String
UserEmail	Email address for the user	Yes	String
UserType	A branch user can be an admin or just a regular user	Yes	String
ReceiveSavePackageNotification	Flag that tells the system if user wants to receive notification that a new package was created using his/her login	Yes	Boolean
ReceiveAutomatedReports	Flag that tells the system if user wants to receive daily status reports by e-mail	Yes	Boolean

3.9.3 Security

Only users with Sys Admin, Company Admin and Branch Admin roles/types are able to create Branch users.

3.10 Create Investor

Creates an Investor. An Investor is an entity (like company) with only one contact information.

3.10.1 Behaviors

1. The user logs in as Sys Admin , Company Admin or Branch User
2. All required fields should be validated as well formats for phone, email and login.
3. When the user clicks the save button the new Investor is created and it is ready to be used. An email with new credentials is sent to the user to notify about new account.

3.10.2 Fields

These fields are used in the Investor creation:

Field	Description	Required	Type
InvestorID	ID of the new Investor	yes	Integer
InvestorName	Name of the Investor	Yes	String
UserLogin	Login entered by the user. It should be an email and unique in the system	Yes	String
Password	Password entered by user	Yes	String
InvestorFirstName	Investor's first name	Yes	String
InvestorLastName	Investor's last name	Yes	String
InvestorAddress	Mailing address of the investor	Yes	String
InvestorCity	City of the investor	Yes	String
InvestorState	State of the investor	Yes	String
InvestorZipCode	Zip Code of the investor	Yes	String
InvestorPhoneNumber	Phone number of the Investor	No	String
InvestorFaxNumber	Fax number of the Investor	No	String
UserEmail	Email address for the Investor	Yes	String
FannieMaeCompliant	Flag that describes if investor is in compliance with Fannie Mae's requirements for purchase of, and lender's servicing of, eMortgage loans.	Yes	Boolean

3.10.3 Security

Sys Admin, Company Admin and Branch Users are able to create Investors.

3.11 Create Escrow

Creates an Escrow company in the system. An Escrow company can have many contact users.

3.11.1 Behaviors

1. The user logs in as Sys Admin , Company Admin or Branch User
2. All required fields should be validated as well formats for phone, email and login.
3. When the user clicks the save button the new escrow company is created and it is ready to be used.

3.11.2 Fields

These fields are used in the Escrow creation:

Field	Description	Required	Type
EscrowCompanyName	Name of the escrow company being registered	Yes	String
EscrowCompanyAddress	Mailing address of the escrow company	No	String
EscrowCompanyCity	City of the escrow company	No	String
EscrowCompanyState	State of the escrow company	No	String
EscrowCompanyZipCode	Zip Code of the escrow company	No	String
EscrowCompanyPhoneNumber	Phone number of the escrow company	No	String

3.11.3 Security

Sys Admin, Company Admin, and Branch Users are able to create an escrow company.

3.12 Create Escrow Contacts

After an escrow company is created the administrator can add contacts (escrow users) in the escrow company.

3.12.1 Behaviors

1. In manage Escrow Companies. The user selects the escrow company from the list.
2. A list of contacts is shown in the screen. The user can:
 - a. Edit contact
 - b. Create new Title company user
3. In Create Escrow company enter the data. Information should be validated when saved.
 - a. Required fields
 - b. And fields format

3.12.2 Fields

Field	Description	Required	Type
EscrowCompanyID	ID of the selected company	Yes	Integer
EscrowUserLogin	Login entered by the user	Yes	String
EscrowPassword	Password entered by user	Yes	String
EscrowUser First Name	User's first name	Yes	String
EscrowUser Last Name	User's last name	Yes	String
EscrowUserPhoneNumber	Phone number of the user	No	String
EscrowUserEmail	Email address for the user	yes	String
ReceiveNotification	Flag that identifies if notification	Yes	Boolean

generated by the escrow acting in the package should be sent back to the escrow user.

3.12.3 Security

SysAdmin, CompanyAdmin, BranchUser and Escrow User

3.13 Create Recorder

Register a Recorder and sets up the protocol to communicate with that company in order to transfer documents to be recorded and receive recorded images back, to be stored in the system.

3.13.1 Behaviors

1. The user will log as Sys Admin, Company Admin or Branch User
2. The user will fill out the form.
3. Depending on the RecorderCommunicationProtocol value we should read additional fields.
4. The system needs to be able to define protocols and fields to read as well need a way to associate the recorder with the protocol. We are looking for a pluggable solution for the communication protocol
5. After additional fields are read the data should be validated and saved

3.13.2 Fields

There are common fields being read for the recorder and there are fields dependent on the communication protocol. For now we will show an example for SFTP.

Field	Description	Required	Type
RecorderCompanyID	Internal identification number of the recorder company	Yes	Integer
RecorderCompanyName	Name of the recorder company being registered	Yes	String
RecorderCompanyAddress	Mailing address of the recorder company	No	String
RecorderCompanyCity	City of the recorder company	No	String
RecorderCompanyState	State of the recorder company	No	String
RecorderCompanyZipCode	Zip Code of the recorder company	No	String
RecorderCompanyPhoneNumber	Phone number of the recorder company	No	String
RecorderCommunicationProtocol	Defines what kind of protocol the system will use to communicate		

with the recorder
company.

3.13.3 Protocols

- SFTP – should read server, login, password and initial folder fields
- Web services

3.13.4 Security

Sys Admin, Company Admin and Branch User.

3.13.5 Open issue

Each recorder will have to be studied to identify the required communication protocol.

3.14 Create eVaulting Companies

Register an eVaulting company and sets up the protocol to communicate with that company in order to transfer original documents to be stored in the vaulting system.

3.14.1 Behaviors

6. The user will log as Sys Admin, Company Admin, Branch User or Escrow user.
7. The user will fill out the form.
8. Depending on the eVaultingCommunicationProtocol value we should read additional fields.
9. The system needs to be able to define protocols and fields to read as well need a way to associate the eVaulting with the protocol. We are looking for a pluggable solution for the communication protocol
10. After additional fields are read the data should be validated and saved
11. Documents should be transferred.

3.14.2 Fields

There are common fields being read for the eVaulting and there are fields dependent on the communication protocol. For now we will show an example for SFTP.

Field	Description	Required	Type
EVaultingCompanyID	Internal identification number of the EVaulting company	Yes	Integer
EVaultingCompanyName	Name of the EVaulting company being registered	Yes	String
EVaultingCompanyAddress	Mailing address of the EVaulting company	No	String
EVaultingCompanyCity	City of the EVaulting company	No	String
EVaultingCompanyState	State of the EVaulting company	No	String

EVaultingCompanyZipCode	Zip Code of the EVaulting company	no	String
EVaultingCompanyPhoneNumber	Phone number of the EVaulting company	no	String
EVaultingCommunicationProtocol	Defines what kind of protocol the system will use to communicate with the eVaulting system	yes	String

3.14.3 Protocols

- SFTP – should read server, login, password and initial folder fields

3.14.4 Security

Sys Admin, Company Admin, Branch User and Escrow User.

3.15 Create Remote Signing Service Companies (RSSC)

Registers a Remote Signing Service Company in the system. This type of company will have temporary access to selected packages for online or remote signing.

3.15.1 Behaviors

1. The user will log as Sys Admin or Escrow User
2. The user will fill out the form.
3. Data must be validated
4. When save, a notification is sent to the company email with login information

3.15.2 Fields

Field	Description	Required	Type
RemoteCompanyID	Internal identification number of the RSSC company	Yes	Integer
RemoteCompanyName	Name of the RSSC company being registered	Yes	String
RemoteCompanyAddress	Mailing address of the RSSC company	No	String
RemoteCompanyCity	City of the RSSC company	No	String
RemoteCompanyState	State of the RSSC company	No	String
RemoteCompanyZipCode	Zip Code of the RSSC	No	String

	company		
RemoteCompanyPhoneNumber	Phone number of the RSSC company	No	String
RemoteEmail	Email used for company login	Yes	String
RemotePassword	String password used in the system	Yes	String

3.15.3 Security

Sys Admin and Escrow User will be able to create these types of Companies.

3.16 My Account

Show the current information for the logged user and allow information to be modified.

3.16.1 Behaviors

1. Show the current user information
2. Allow the user to modify the fields (Role, login are not allowed)
3. In order to save the modifications user will have to provide current password. Password should be checked and the system should save the information if they match.

3.17 Delete Company

Remove a company from the system

3.17.1 Behaviors

1. The user log as Sys Admin
2. The user selects a company from a list
3. The delete option is presented. If clicked the system will request the admin password (current logged user) in order to complete the action.
4. All the records related to that company should be removed from the system. There will be no way to revert the process.

3.17.2 Security

Only Sys Admins are allowed to perform this task

3.18 Delete Branch

Remove a Branch from the system

3.18.1 Behaviors

1. The user log as Sys Admin or Company Admin
2. The user selects a company from a list and also a branch
3. The delete option is presented. If clicked the system will request the admin password (current logged user) in order to complete the action.

4. We will flag the branch record as deleted. Data is not physically removed from the system. There will be option to query data from “deleted” branches.

3.18.2 Security

Only Sys Admins and Company Admin are allowed to perform this task

3.19 Delete User

Remove a User from the system

3.19.1 Behaviors

1. The user log as Sys Admin, Company Admin or Branch Admin
2. The user selects a company and or Branch. A list of user will be shown with delete option.
3. If delete is clicked the system will ask for confirmation in order to complete the action.
4. We will flag the user record as deleted. Data is not physically removed from the system.

3.19.2 Security

Only Sys Admins, Company Admin and Branch Admin are allowed to perform this task

3.20 Add Company Resource

The system allows administrator to customize some resources in the system. The table below describes the resources in the system.

Resources	Description
Banner	A banner is a flash file and should have dimension 120x360
Logo	It is a GIF or JPG file with dimensions 120x70
Email Template	<p>Templates used to create the notifications for the entities involved in the closing process. The following templates are available:</p> <ul style="list-style-type: none"> • Save Package notification • Send notes notification
Time To Archive	Integer that define after how many days a package should go to archive state

3.20.1 Behaviors

1. User logs as Sys Admin
2. Select the option for Resource Management
3. A list of companies and or branches is showed. The user selects one or both.
4. At this point the user can select the type of resource to add and click the New Resources button. Each resource will have its own interface.
5. Some resources only allow one in the system. If a resource already exists the user will have to remove or replace the resource data.

6. In this interface the user will also be able to delete resources.
7. In this interface the user will be able to add resource for a company or branch depend on the selection made.

4 Creating eClosing Packages

This function allows the Lender to create an eClosing package in the system. The create package is a wizard that will walk the user through the steps to create an eClosing package. The package creation will also be available through the web services on a programmatic way.

4.1 Step 1: Loan Information

4.1.1 Behaviors

1. The user logs as Branch User
2. The user selects create package. The Loan information page is displayed.
3. The funder, escrow, investor and recorder fields will contain a select button that will take the user to another screen that will allow escrow contact, funder, investor or recorder to be selected, or created and then selected. For more details refer to the current spec. In the automated process, only escrow contact may be informed; funder, investor and recorder will be manually selected later, by a branch or escrow user.
4. If release to escrow is selected:
 - a. Notification is sent to escrow contact
 - b. Escrow has access to the package when created.
5. If Release to borrower is selected:
 - a. Notification is sent to borrower
 - b. Borrower has access to package when created.
6. Validation:
 - a. Required fields need to be validated
 - b. Format for email, zip
 - c. Field size
7. The user will click next to go to the Add Borrower Step

4.1.2 Fields

Field	Description	Required	Type
CreationDate	Date that the package was created	Yes	Datetime
LoanNumber	Number assigned for the loan (unique by company)	Yes	String
EscrowNumber	Number assign to the loan at escrow company	No	String
PropertyAddress	Address of the property	Yes	String
PropertyCity	City where the property is located	Yes	String
PropertyState	State where the property is located	Yes	String

PropertyZipCode	Zip code for the property	Yes	String
Funder	Email for the funder in the Lender company	No	String
EscrowContact	Email for the escrow contact in the escrow company	No	String
ReleaseToEscrow	Flag that informs if package should be accessible to escrow after creation	Yes	Boolean
ReleaseToBorrower	Flag that informs if package should be accessible to Borrower after creation	Yes	Boolean
Recorder	Id for the recorder entity registered in the system	No	Integer
Investor	Id for the investor entity registered in the system	No	Integer

4.1.3 Security

Branch User is the only one with access to create package.

4.2 Step 2: Add Borrowers

This wizard step will allow the user to identify the borrowers in the eClosing process.

4.2.1 Behaviors

1. The user should click the Add Borrowers button in order to enter information
2. User enters data and click Save Borrower button.
 - a. The data should be validated.
 - b. The borrower should be saved and added to the borrower list.
 - c. The user can repeat this steps for additional borrowers.
3. At any time in the process the user can press NEXT to go to: Add Documents.
4. At any time in the process the user can press Previous to return to: Property Information.
5. At any time in the process the user can press Cancel. All the data about this package should be erased.
6. The borrower list will contain options to Edit Borrower and delete borrower.

4.2.2 Fields

Field	Description	Required	Type
BorrowerLastName	Last Name for the Borrower	Yes	String
BorrowerFirstName	First Name for the Borrower	Yes	String
BorrowerEmail	Email for the Borrower	Yes	String
Last4Social	Last 4 digits of social security	Yes	Integer
Address	Borrower's Address	Yes	String
City	Borrower's City	Yes	String
Sate	Borrower's State	Yes	String
ZipCode	Borrower's Zip Code	Yes	String

4.2.3 Security

Branch User is the only one with access to create package.

4.3 Step 3: Add Documents

This step lets the user to add documents to the package.

4.3.1 Behaviors

1. In the add documents step wizard a control will be presented. This control allows the user to upload several documents at a time.
2. PDF and SMART documents are the only format supported. Safedocs will provide to escrow companies a desktop application to convert different input format to PDF.
3. If a document format is not supported the system should show a message. Other supported documents should be uploaded.
4. Flags assigned to the document (see fields bellow) should be applied to each document being uploaded.
5. After upload, the documents will show on a list. This list will allow the user to remove and change flags for documents previously uploaded.
6. The user can click Finish to save the package or click Next to go to Add Notes step.

4.3.2 Fields

Field	Description	Required	Type
LegalDocument	This flag helps to group the document based on legal type	Yes	Boolean
NeedsNotarization	Signal the system if the document can be signed under borrower login.	Yes	Boolean

4.3.3 Security

Branch User is the only one with access to create package.

4.4 Step 4: Add Notes

This step allows the user to send to notes to entities in the closing process.

4.4.1 Behaviors

1. The user can add a note to be added to the notifications sent after save package.
2. The user can click finish and the package will be saved
3. The user can click cancel to end up the package creation.

4.4.2 Fields

Field	Description	Required	Type
Note	Note to be added to all notifications sent after	No	String

4.4.3 Security

Branch User is the only one with access to create package.

4.5 Finishing Create eClosing Package

Although the final step happens when the user click finish on the Add Documents or Add Notes step, we decided to detail the behavior separately.

4.5.1 Behaviors

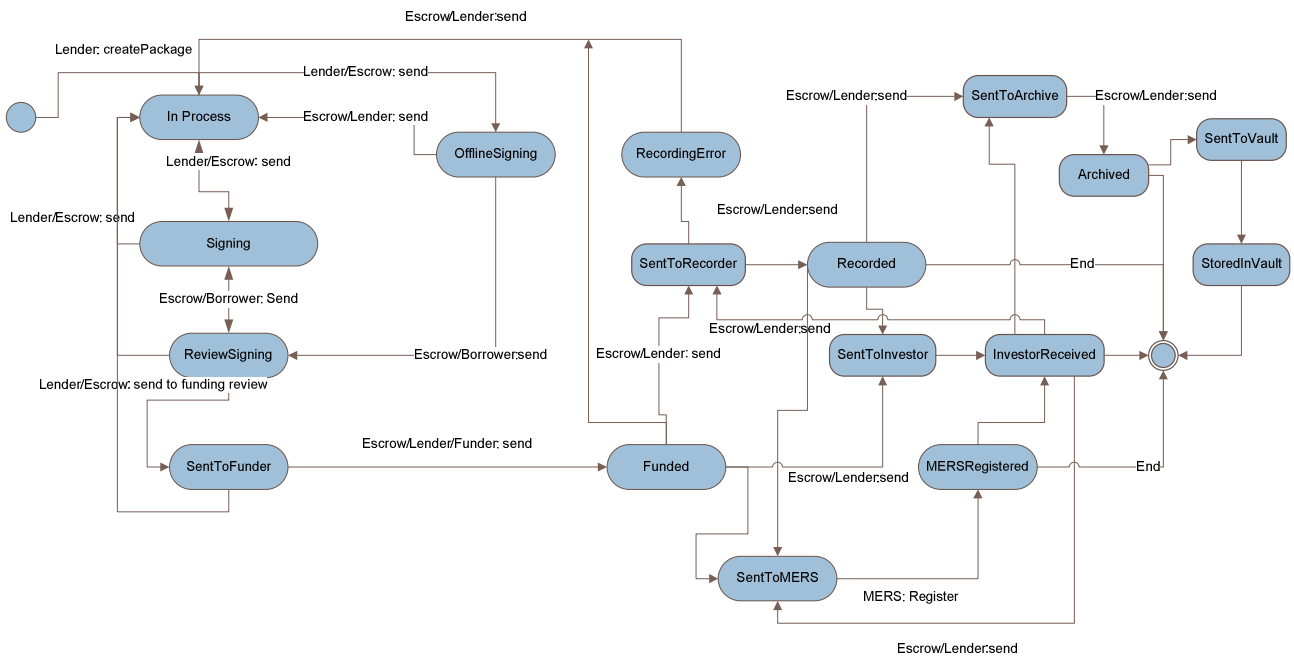
1. User clicks finish on the Add Documents step.
2. The system saves all the information about the package.
3. At this point, if no error occurred, the package is said to be in **Pending** state.
4. If the SendToEscrow flag is checked:
 - a. An email notification generated for the Escrow contact. At this point the escrow has access to the package.
5. If the SendToBorrower flag is checked:
 - a. An email notification is generated for the borrower. At this point the borrower has access to the package.

5 Package Management

The entry point for package management is the My Packages screen. This section will describe the actions accessible for this screens as well the workflow for the eClosing process.

5.1 Workflow

The workflow section defines all possible package states, accessibility and actions that can be taken by the entities involved in the eClosing process. We will start defining all the states and then the transitions.



5.1.1 States

At all stages the parts involved will be able to send notifications to other parts involved during the state transitions. Notifications are sent to all parts or can be routed to specific parts selected by the sender.

State	Description	Allowed Entity
In Process	The package assumes this state as soon as it is created. At this point depending on the creation flags (ReleaseToEscrow and ReleaseToBorrower) the package will be available for perusal. The possible transitions include: Canceled, Signing or OfflineSigning.	If flagged <ul style="list-style-type: none"> Escrow with full access to the package including send notes (Add documents, remove documents, notes, change package information and status). Borrower will be able to view, print and send notes only.
Signing	This state tells the system that the package is waiting for the borrowers to sign the documents. The possible transitions include: Canceled, In Process, OfflineSigning and ReviewSigning.	Lender has full access to the package Escrow <ul style="list-style-type: none"> Can login on behalf of Borrower so he/she can sign. Can change the package state (full access). Documents may be notarized Documents can be edited

		<p>Borrower</p> <ul style="list-style-type: none"> • Can use the email sent in order to log on the system and sign the documents. • Can signal that all the documents were signed
OfflineSigning	<p>Same as signing but the package is locked. This means nobody in the system can modify the package while in this state. The package can still be viewed. The possible transitions include: Canceled, In Process, Signing and ReviewSigning. The remote signing user will be notified to download the package to a laptop. After all documents are signed offline, the remote signing user will connect to the server and return the signed documents to the system.</p>	<p>Lender has full access</p> <p>Escrow</p> <ul style="list-style-type: none"> • Can send someone with Offline package • Can cancel the package • Can upload offline documents back in the system • Can change the package status to ReviewSigning
ReviewSigning	<p>This state tells the system that borrower signed the required documents and all documents are ready to be reviewed by Lender or Escrow. The possible transitions include: Canceled, In Process, Signing and SentToFunder.</p>	<p>Lender/Escrow</p> <ul style="list-style-type: none"> • Has full control and can change back to any state. • Can send the package status to SentToRecorder or SentToFunder
SentToFunder	<p>This state tells the system that signed documents were reviewed and the loan is ready to be funded. When the package transitions to this state the Funder will be notified he has access to the documents. The possible transitions include: Canceled, In Process, Signing and Funded.</p>	<p>Lender/Escrow</p> <ul style="list-style-type: none"> • Has full control and can change back to any state. • Will change the package state to Funded when the funds received. <p>Funder</p> <ul style="list-style-type: none"> • Can view the documents in the package • Send notifications
Funded	<p>This state tells the system that the loan was funded. When the package</p>	<p>Lender/Escrow</p>

	<p>transitions to this state the entities involved in the process will be notified. The Funder will no longer see that package. The possible transitions include: Canceled, In Process, Signing, ReviewForFunding, SentToInvestor and SentToRecorder.</p>	<ul style="list-style-type: none"> • Has full control and can change back to any state. • Can send the package to Recorder or Investor (SentToRecorder State)
SentToRecorder	<p>The package is sent to recorded state to be recorded. Some documents will be processed by the recorder and be sent back to be stored in addition to the current documents. The result for this process will transition the package to Recorded or RecordingError. The system may use a centralized repository for sending and receiving information from the Recorder (HUB model).</p>	<p>Lender/Escrow</p> <ul style="list-style-type: none"> • Has full control and can change back to any state. • Can send the package <ul style="list-style-type: none"> ○ RecordingError ○ Recorded
RecordingError	<p>Signals that the system was unable to access the Recorder system. The user can resubmit or send the package to any other previous state.</p>	<p>Lender/Escrow</p> <ul style="list-style-type: none"> • Has full control and can change back to any state. • Can resubmit the package or cancel
Recorded	<p>The Recorded signaled the system that the documents were recorded.</p>	<p>Lender/Escrow</p> <ul style="list-style-type: none"> • Has full control and can change back to any state. • Can send the package to: <ul style="list-style-type: none"> ○ SentToArchive ○ SentToMERS ○ SentToInvestor
SentToInvestor	<p>The Package is sent to investor. A result for this process will transition the package to InvestorReceived. If investor is flagged as FannieMae the system we will comply with their data format and communication protocol.</p>	<p>Lender/Escrow</p> <ul style="list-style-type: none"> • Has full control and can change back to any state. • Can Send the Package to: <ul style="list-style-type: none"> ○ InvestorReceived

InvestorReceived	Signals that the package with selected documents was received by the investor	<p>Lender/Escrow</p> <ul style="list-style-type: none"> • Has full control and can change back to any state. • Can send the package to: <ul style="list-style-type: none"> ○ SentToMERS ○ SentToArchive ○ SentToRecorder ○ End the process.
SentToMERS	Send the loan registration information to MERS eRegistry. It will follow MERS standards. The system will create an ePackage or re-use the one sent at creation time.	<p>Lender/Escrow</p> <ul style="list-style-type: none"> • Has full control and can change back to any state. • Can send the package to: <ul style="list-style-type: none"> ○ MERSRegistered
MERSRegistered	MERS registry will acknowledge the package and the system will modify the package status	<p>Lender/Escrow</p> <ul style="list-style-type: none"> • Has full control and can change back to any state. • Can send the package to: <ul style="list-style-type: none"> ○ SentToInvestor ○ SentToArchived ○ End The Process
SentToAchive	The Package is archived locally or using a third party company.	<p>Lender/Escrow</p> <ul style="list-style-type: none"> • Has full control and can change back to any state. • Can send the package to: <ul style="list-style-type: none"> ○ Archived
Archived	The package is archived in the system.	<p>Lender/Escrow</p> <ul style="list-style-type: none"> • Has full control and can change back to any state. • Can send the package to: <ul style="list-style-type: none"> ○ End workflow • The package can no longer be changed
SentToVault	Original documents are sent to any eVault in compliance with MISMO	<p>Lender/Escrow</p> <ul style="list-style-type: none"> • Has full control and can change back to

StoredInVault	The eVault system has accepted the documents	Lender/Escrow <ul style="list-style-type: none"> • Has full control and can change back to any state <ul style="list-style-type: none"> ○ End Workflow
---------------	--	---

5.2 Packages Module

Main module that allow management of the packages in the system

5.2.1 Behaviors

1. This screen is accessible when a branch or escrow user logs in the system
2. The my package functionality will allow the user:
 - a. Advance Search Packages
 - b. List Archived, troubled and in Process Packages
 - c. Edit Package
 - d. Delete Package
 - e. View Package Content
 - f. Manage Package Closing
 - g. Manage document accessibility
 - h. Cancel Package
3. At login by default to have a list of in Process Packages.
4. In addition to the actions listed above the user can use a shortcut “More Actions” that can be applied to each package. The additional actions in the package will be dependent on the actual package state as well who is logged in the system.

5.3 Offline signing

Offline signing is a complicated process that deserves its own explanation. This section will describe the Offline Signing process:

1. Escrow selects documents in a package and a RSSC to have access to package. Package status is changed to OfflineSign and become locked to any other transaction besides viewing.
2. A notification is sent to the remote signing service contact address. Notification should contain access information, borrowers’ contact information and instructions on the remote signing process.
3. The RSSC will log in to the system and assign the package to one of its contacts. If the desired contact person is not in the system, a new contact can be added by entering First and Last names, email, login and password.

4. The system will notify this person that the package and the offline signing application can be downloaded. Login and borrower contact information is also sent at this time.
5. The agent goes to the system and downloads the signing client (if necessary) and the package. The package is a protected Zip file that was compressed with a key that can only be generated with information from the Borrower. That way we guarantee that only the Borrower will be able to access the package.
6. The agent takes the application to the road and proceeds with the regular signing process.
7. After the signing process the agent can notarize the package with its own signature.
8. The package is then sealed again.
9. The agent connects back with eClosing system and uploads the package. Package is unlocked and documents are validated.
10. The new documents will replace the old ones and the package will change state to notify the system that the signature is done.
11. Notification is sent to branch user, escrow contact and borrower.

5.3.1 Fields

Field	Description	Required	Type
RemoteLastName	Last Name for the Remote Contact	Yes	String
RemoteFirstName	First Name for the Remote Contact	Yes	String
RemoteEmail	Email for the Remote Contact	Yes	String
RemoteLogin	Login for the Remote Contact	Yes	String
RemotePassword	Password for the Remote Contact	Yes	String

6 Borrower Package Management

The module gives the borrower the ability to examine and sign documents in the package.

6.1 Borrower Login

This is the initial screen in the Borrower module. Usually the Borrower will be able to access this screen through the email notification received from the Lender/Escrow after Borrower is given permission to access the package. Depending on the package status, the signing feature may or may not be available.

6.1.1 Behaviors

1. The user will log in the system using the passkey issued in the email, keycode (verify existence of Adobe Reader) as well last 4 digits of social security.
2. After clicking Login the system will verify the data against the package and will redirect the user to the main package Viewer.

6.1.2 Fields

Field	Description	Required	Type
Passkey	System generate key,	Yes	String

	different for each borrower		
LastDigitsSocial	Last 4 digits of Borrowers social security number	Yes	Integer

6.1.3 Security

Only Borrowers with the information described above will have access to the package.

6.2 Borrower Viewer

This screen is the entry point for viewing and signing documents, depending on the status of the package. Only documents that do not require notarization can be signed in the borrower viewer.

6.2.1 Behaviors

1. The Borrower will see a list of documents on the left side of the screen. Documents that needs signature should be flagged with some visual cue. Documents that require notarization can be viewed, but not signed.
2. The user can click any document from the list. The selected document will show in the document viewer pane on the right side.
3. If document is a SMART Doc the system should validate it and display a visual cue that shows the conformance of the document with MISMO standards.
4. At any time the Borrower can Save Documents. This functionality should compress all the documents in a zip file and save in the user's local disk.
5. At any time the Borrower can print all documents or just the current document.
6. If the selected document has signature fields, an option should be shown allowing the Borrower to sign the document. When clicked:
 - a. The Borrower will see a signature term explaining the signature process. The Borrower should accept or not.
 - b. If accepted the system will sign all the signature fields associated with the Borrower.

6.2.2 Signature Types

Although the viewer will be the same, the type of signature that will be used can be different depending on the type of document and type of signature field. The table below defines the proposed signatures to be supported and when to use them.

Type	Type of Document	Notes
Click2Sign	PDF	<ul style="list-style-type: none"> • Borrower selects document • A term is shown. Borrower can accept or not • If accepted the system identify all signature belonging to Borrower and apply the signature to the signature

		<ul style="list-style-type: none"> fields in the selected document. • Not a valid method for notarization.
PKI (digital signature)	PDF, SMART Doc	<ul style="list-style-type: none"> • Requires the signer to have a PKI
Electronic Signature (live)	PDF	<ul style="list-style-type: none"> • Requires signing pad • Signature saved applied as image, so no useful biometric data for validation • Not a valid method for notarization.
Biometric (handwritten signature, fingerprint)	PDF, SMART Doc	<ul style="list-style-type: none"> • Requires signature product that captures and stores biometric information. • Signature pad and/or fingerprint sensor

The signature process can also be accomplished remotely using an online meeting service where the Borrower will allow a third person to sign documents on his/her behalf. This method presents some challenges that will be circumvented by using audio and video recording that needs to be stored with the package. Here is how the scenario would look like:

1. The Lender calls the Borrower to notify that closing documents are ready to be signed
2. The Lender invites the Borrower to join an online meeting session and explains that the session will be recorded (audio and video).
3. The Borrower gives the Lender power to sign on his/her behalf
4. The Lender shows each document and signature field in the eClosing system and asks if Borrower agrees with signing the fields.
5. When the process finishes the Lender will upload in the audio/video file to the package. That way the whole transaction is recorded and saved with the package.

6.2.3 Open Issues

1. PDF: We can propose signature ids "B1Sig1", "B1Sig2", "B2Sig1", "B2Sig2", etc. to identify all signatures belonging to each Borrower. When signing, we can allow borrower to select Sign All or sign each signature field individually.

6.3 Notarization

Some documents may require notarization. Documents that require notarization cannot be signed under borrower login.

6.3.1 Behaviors

1. A notary (escrow contact or RSSC) would log in to the systems in the presence of the borrower.
2. Borrower would sign documents.
3. Notary would notarize (digital signature) document.

4. If notary has Enjoa by National Notary Association system, the notary can use an option available on the signing interface to open Enjoa.
5. After finishing using Enjoa and closing that application, notary would be back in the signing interface.
6. After all documents are signed and notarized, notary should use an option to inform that package has been completely signed.
7. Application should request confirmation and then close signing interface.

7 Interfaces to External Systems – Web Services

Web Services make up a crucial component of the eClosing architecture since they are needed to support eClosing Package creation as well communication (B2B) with third party components of the system. For that reason it is imperative that our designs provide loosely coupled services that can be used by any service consumer; consequently we follow the principles of SOAP and XML RPC. Package creation will be implemented using XML RC, as per DocMagic’s request.

The web services will fall into the categories named SOAP Web Services and XML RPC. All the responses will be done using Xml to represent the results.

7.1 Versioning & Extensibility

Two types of versioning are common in web services: contract and message. Contract versioning relates to versioning the WSDL and contract information used to describe the service [7]. XML message versioning relates to the versioning of schemas used to describe messages processed by the service [7].

Message versioning applies to both XML and JSON data returned. We may encounter these scenarios when updating our messages [6]:

1. New concepts are added
2. Existing concepts are changed
3. Existing concepts are deprecated
4. Existing concepts are removed

7.2 XML Message Versioning

There are a few ways to deal with the message update scenarios using schemas [6, 8, 9]. However, validation of documents with respect to extensibility and versioning is very cumbersome [10]. We compromise by targeting an approach that fits the expected usage of our data (XPath addressing) and follows many of the best practices of [6, 8, 9]. These best practices are summarized in “Versioning XML Languages” by the W3C [5]:

DEFINITION: “Language” in the following requirements refers to the set elements and attributes of our data along with their defined constraints.

1. Allow Extensibility: Languages designers SHOULD create extensible languages.

2. Any Namespace: The language SHOULD provide for extension in any namespace.
3. Full Extensibility: All XML Elements that can allow attributes, i.e. ComplexTypes in XML Schema, SHOULD allow any attributes and any elements in their content models.
4. Namespace identifies compatible names: The namespace name identifies names that are compatible within the same namespace name.
5. Identify specific version with version attribute: The specific version of a set of names within a given namespace may be identified with a version attribute to differentiate between the compatible versions.
6. Provide Processing Model: Languages MUST provide a processing model for dealing with extensions.
7. Must Ignore: Receivers MUST ignore any XML attributes or elements that they do not recognize in a valid XML document.
8. Must Ignore All: The Must Ignore rule applies to unrecognized elements and their descendents.
9. Must Ignore Container: The Must Ignore rule applies only to unrecognized elements
10. Re-use Namespace Names and Element Names: If a backwards or forwards compatible change is made to an element definition by the owner of the element's namespace, then the old namespace name and element names SHOULD be used in conjunction with the extensibility model.
11. Only Namespace Owners Change Namespace: The namespace name owner is the only entity that is allowed to change the meaning of names in a namespace.
12. Provide Must Understand: Container languages MUST provide a "must understand" model for dealing with optionality of extension elements that override the must ignore good practice.
13. Use Must Understand: When a language provides a must understand model, extensions MUST use it when the extension is required.
14. Must Understand for elements: Container languages should provide a Must Understand model for elements and not attributes
15. Promote Must Understand: If a must understand option is not provided inside a particular extension element, the extension that must be understood SHOULD be promoted to the first container that does provide a must understand option.
16. Multiple Extensions: Containers should specify the expected behavior of multiple extensions.

We follow best practices 1, 2, 3, 4, 5, 10, 11, and 16 as described in “Web Service Data Requirements” below. We require best practices 6, 7, and 8 as described in “Client Processing Requirements” below. Best practice 9 does not apply because it is mutually exclusive with 8. We do not follow best practices 12, 13, 14, 15 because they apply only to container languages.

7.3 Web Service Data Requirements

The following requirements apply to XML web service output:

- Messages MUST be well-formed.
- Messages MUST use a namespace to describe major versions and a version attribute to describe the minor version, e.g. <Programs xmlns="http://tv.msn.com/ExactSearch/" version="0">
- New elements and attributes compatible with an existing namespace MUST be added to an existing namespace.
- New elements and attributes incompatible with an existing namespace MUST be added to a new namespace.

7.4 Client Processing Requirements

The following requirements apply to client processing of XML:

- Clients MUST process data in a “Must Ignore All” fashion described in [5].
- Clients MUST reject data from an unknown namespace.
- Clients MUST NOT distinguish order of attributes.
- Clients MUST NOT distinguish order of sibling elements.

7.5 References

Name	Comments
1. REST Web Services	Describes a REST Web Service
2. Yahoo! REST Web Services	A sample of Yahoo!’s REST Web Services
3. SOA	Service-Oriented Architecture
4. Introducing JSON	Describes the JSON object format
5. Versioning XML Languages	Describes versioning of XML and XML web services
6. Designing Extensible, Versionable, XML formats	
7. Principles of Service Design: Service Versioning	
8. W3C XML Schema Design Patterns - Dealing With Change	
9. Versioning XML Vocabularies	

The next sections will define the methods in the web service.

7.6 Safedocs eClosing Partner Web Service

This web service provides an interface to automate the eClosing Package creation in the Safedocs eClosing system. Following is the list of methods exposed by this web service. The creation of DocMagic client accounts in the eClosing platform will be “organic.” Accounts will be created as clients use the system. DocMagic does not intend to export the complete client base into SafeDocs (“Fork Lift”).

The eClosing product will provide web services that define the APIs to:

- **CreateClosingPackage:** Checks if entities exist and if not, create Company, Branch, Branch user, Escrow company, Escrow contact person. Create a closing package using package information (XML manifest, MISMO standard tags). Passwords for users will not be required in the manifest and system will define the passwords randomly.

- **EditClosingPackage:** Edit the package replacing information and documents. Documents will be replaced with ones in the xml being sent.
- **GetClosingPackageInformation:** Finds a package in the system and returns xml information about the package.

8 Auditing

Safedocs eClosing system will maintain track of all the actions performed on packages in the system. The information collect will basically be:

- When the task was performed
- Who performed the task
- Which task was performed

In order to support auditing the system will provide several reports. These reports will be described in the report section of this document.

9 Reports

The Safedocs eClosing system will provide some reports that can be seen on the screen or printed. This section will describe the reports. At this point, only Sys Admin and Lender (users) can access the documents.

Reports	Description
Activity	Shows total transactions in a period, by company or branch. It shows totals by package status.
Transaction History	Shows detailed information about selected package(s). The information includes package header information, borrower(s), documents, accesses, transactions and statuses.
Package Status	Shows summarized information about selected packages, including only current status. User may select a
Daily	Automatic or manual. Shows information about new packages or change in status.
Accesses to System	Shows who (and when) accessed the system in the defined period.

9.1.1 Open Issues

1. Safedocs will review the list of reports after analyzing lender's needs reporting information.

10 External eClosing Tools

This section defines external tools provided by Safedocs eClosing system.

- Signature and Text Enabler. This application allows the client to prep and PDF with signature fields as well insert some text areas (identified for notary information in signing).
- Offline Signing System. This is basically an offline Borrower Viewer. It enables a remote signature agency to go to the Borrower in order to collect signatures.
- PDF Rendering Tool. This helper tool will be used to convert supported document format to PDF.

11 Document References

The following documents and products were sources of this version of the FSD:

- a. Requirements notes collected during Safedocs and DocMagic's meeting in Los Angeles, May 13-14, 2008
- b. Safedocs's Finance Suite of products
- c. MISMO eMortgage Guide, 2006
- d. MISMO eMortgage Vaulting Guide, Version 3.0, 2008
- e. MISMO eMortgage Packaging Specification, Version 2.4, 2007
- f. FannieMae's eMortgage Technical Requirements, Version 1.0, 2007
- g. FannieMae's Guide to Delivering eMortgage Loan to Fannie Mae, Version 2.5, 2007
- h. MISMO SMART Document Specification, 2006
- i. Appendix A – Notes about Workflow Modules
- j. <http://www.nationalnotary.org>

12 Appendix A

I. Administration

- a. This interface will create the company and its branches and users(Sig field client)
- b. SDS will manage emails, branding, 3P firms (Escrow/Recorders/Investors/Remote)
- c. DM will maintain this for their clients
- d. Option: Company Admin to manage themselves

II. Module I: CREATION OF PACKAGE

- a. DM will send XML Manifest and documents to SDS to create package place documents
 - i. Status is Created Pending
- b. **Functions:**
 - i. Package will forward notifications based on directions within the Manifest (Notifications will contain passwords)
 - 1. Package created: STOP
 - 2. Lender to Borrower: STOP
 - 3. Lender to Escrow & Borrower: STOP
 - 4. Lender to Escrow; Escrow to Borrower: STOP
 - ii. **Lender** may manually add and prep documents then forward to Borrower and or Escrow
 - 1. Any type of document may be added then converted to PDF
 - iii. **Borrower** may communicate back with Lender, view, save, print documents
- c. **Requirements**
 - i. **XML Manifest**
 - ii. **Documents are identified:**
 - 1. Individual separate documents
 - 2. PDF/SMART
 - 3. Lending documents; Non Lending documents; Notarized(Borrower can't sign by themselves)
 - 4. Signing and Text fields created on documents

III. **Module II: ESCROW**

- a. **Creating Escrow company on fly**
 - i. Manifest will create and or modify Escrow company/branch/user
 - 1. Option: Manual process to ADMIN Escrow companies
 - ii. Create Notice of password for Escrow
- b. **Escrow Functions (A):**
 - i. View, add, delete and prep documents; (Any type converted to PDF)
 - 1. Sig field creator/Local
 - ii. Communicate to Lender and Borrower
 - iii. Manually Forward notice to Borrower to ONLY view documents
 - iv. Package may be processed multiple times (Redraws-Communications sent back to Doc Drawer)
 - v. No Changing after Recording, but may add documents until Achieved
- c. **Escrow Functions (B):**
 - i. Status changed from Pending to Signing
 - ii. **Forward Documents to Borrowers for signing**
 - 1. Specific Invitation
 - 2. Click sign features available
 - iii. **Sign Borrowers in Escrow office (Full/Hybrid)**
 - iv. **Facilitate Remote Closing(Module IIa):**

1. SDS will Admin Signing companies
 2. SDS will forward connection password information
 3. Agents may sign remotely:
 - a. On Line (Full/Hybrid)
 - b. Off Line
 - i. Down loading client
 - v. **Signing technology (Modular IIb)**
 1. Multi capacity: Click, PKI, Bio, Live signature
 2. Ability to add text to Notary documents
 3. Access to Notary Journal
 - vi. **Video Closing (Modular IIc)**
 1. Process can be performed with 3P program to record wave & video actions during closing
 2. Store Wave & Video files within the SDS
- d. **Escrow Functions (C):**
- i. Notice forwarded to Lender's **FUNDER**
 - ii. *Status changed to Funded*
 - iii. **Escrow can forward recordable documents to County Recorder (Module IId)**
 1. SDS Administrator regarding Hubs for County Recorder
 2. Recordable copies created from system originals in PDF or TIFF
 3. SDS delivers and retrieves documents from Recorder's Hub
 4. SDS notifies both Lender and Escrow of returned documents
 - a. Errors/Open: rejected documents/transmission
 - b. Resubmit corrections
 - iv. Escrow may only forward documents to system up to Archiving status
 - v. *Status Changed to Recorded*
- IV. **Lender Module III:**
- a. **Register (MERS) Optional**
 - i. **Function:**
 1. User- Lender/Escrow submits
 2. Escrow may submit
 3. SDS provides window displaying MERS information
 - ii. *Status Change to Registered*
- V. **Lender Module IV:**
- a. **Authoritative Copy: Optional**
 - i. **Function:**
 - ii. SDS allows for Lender user to forward Authoritative copy to internal/3P eValut/storage
 - iii. SDS will have ADMIN to create customized Manifest
- VI. **Lender Module V:**
- a. **Investor docs to Custodian: Optional**
 - i. Lender User to send required investment docs (upload)
- VII. **Lender Module VI:**

a. Archiving/Storage

i. Functions:

1. Lender User forward documents/package to Long Term Storage
2. SDS will create client specific Manifest that will contain History/documents to seamlessly integrate with archiving/imaging systems
3. SDS will have ADMIN to create customized Manifest
4. System will automatically purge to long term storage after X days

ii. Status Change to Archived

VIII. General:

a. List of Status'

- i. Creation/Pending; Signing; funded; Recorded; Registered ;Achieved

b. List of Document types

- i. Lending Documents; Non Lending Documents; Notarized Documents

c. SDS Technology- Java

d. SDS Security