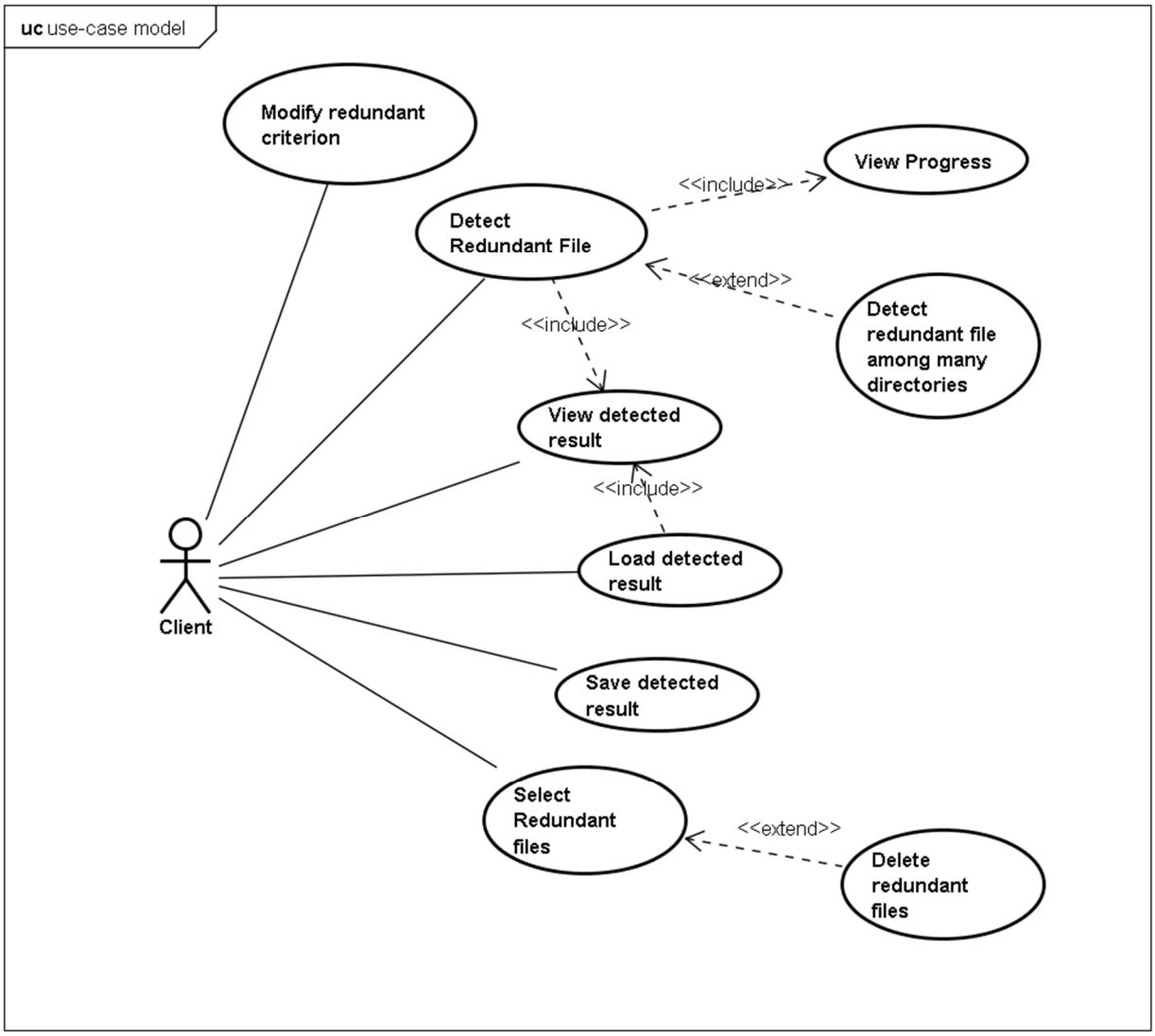
Requirement
I am the client:
I want a software that is able to detect redundant files in directory or among many directories (without cycle). A file is redundant when some files have identical name, size and/or contents (MD5 hash?). Of course, the result must be saved in order to use it in future, to modify redundant criteria or to select/delete redundant files.

Analysis Document
From the requirement above, I did the analysis processes and described here.

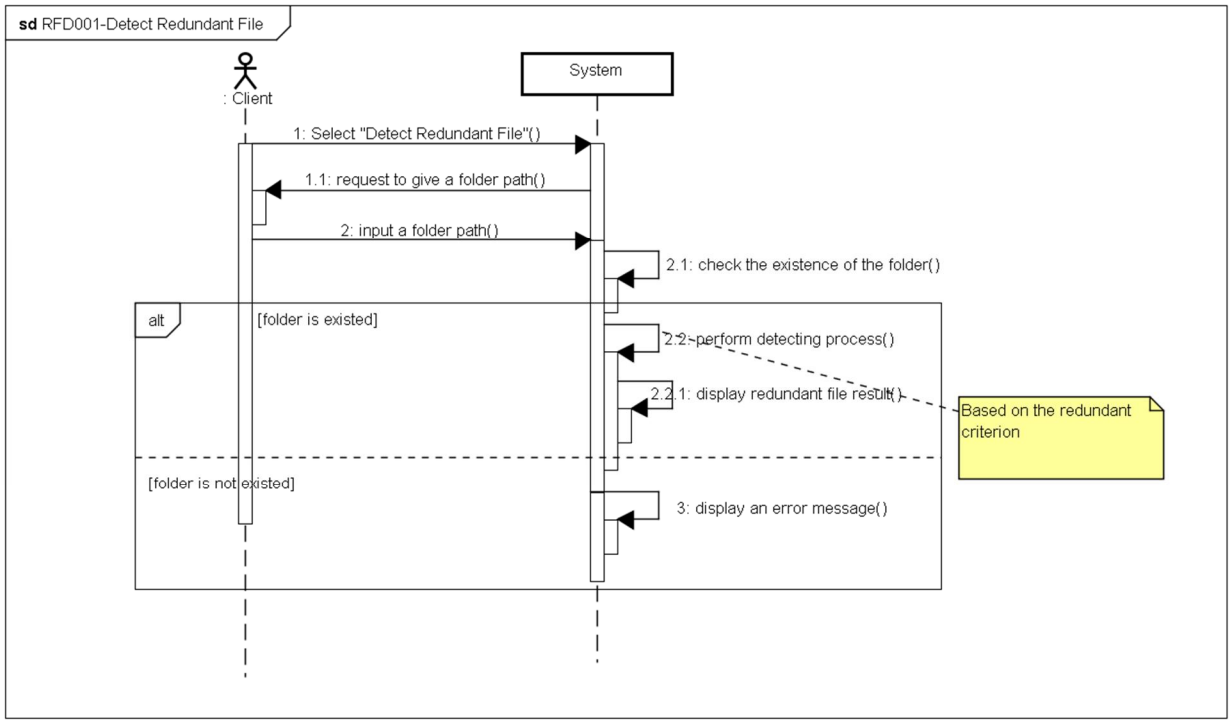
1. **Use-case diagram**



Following descriptions are the detail explanation of the cases on the diagram.
**Note: N/A is the shortcut of Not Available.*

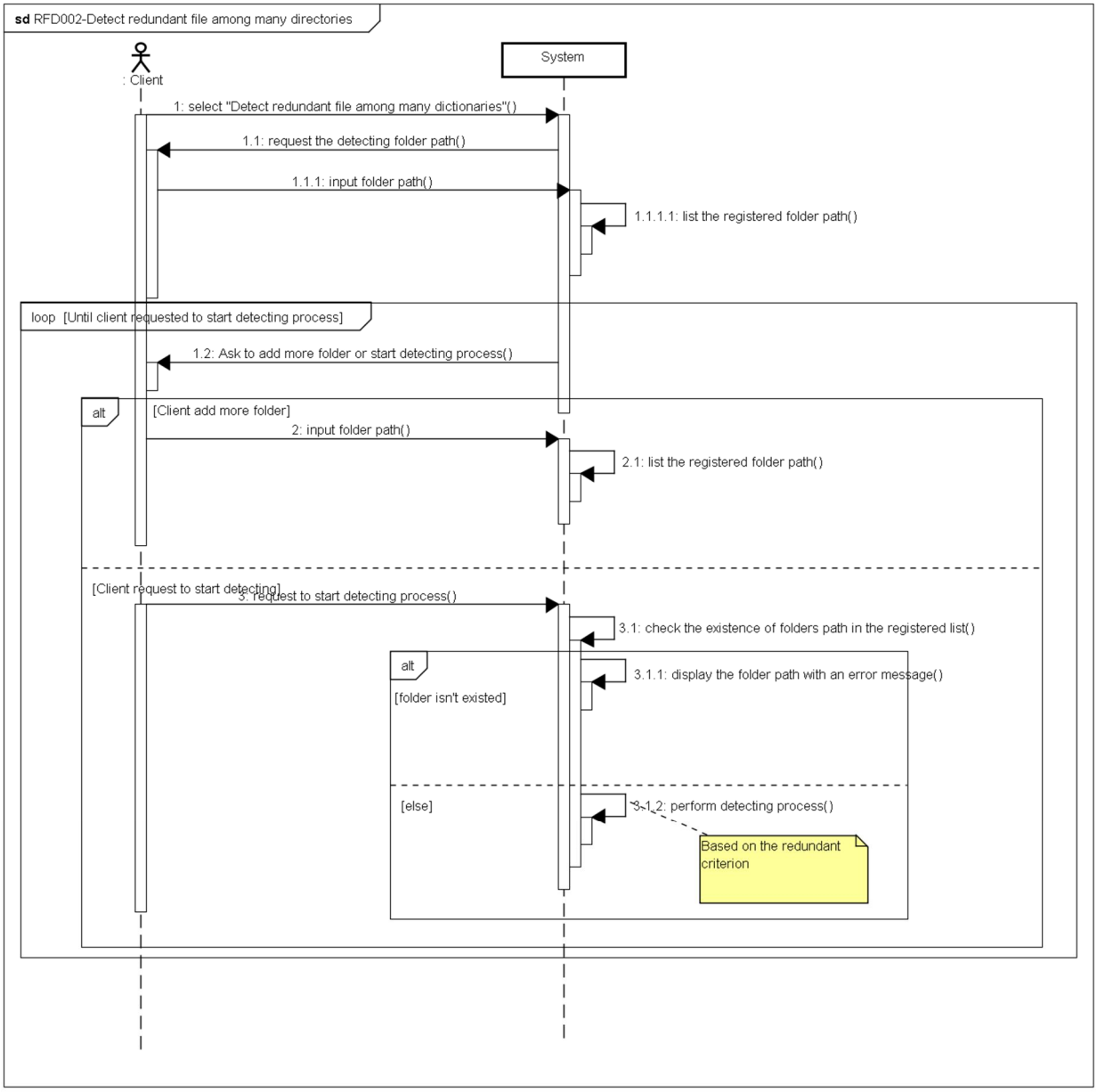
Use Case ID:	RFD-001		
Use Case Name:	Detect Redundant File		
Created By:	DuongTB	Last Updated By:	DuongTB
Date Created:	2014/11/25	Date Last Updated:	2014/11/25

Actor:	Client
Description:	Client request the system detecting all of redundant file in a given directory and its sub-directories according to the redundant criterion which are configured in advance.
Preconditions:	Client is working on the main screen
Post-conditions:	All of redundant file are listed.
Priority:	High
Frequency of Use:	Always
Normal Course of Events:	1. Client select the "Detect redundant file" on the menu. 2. System request the client giving the path to the expected checking directory. 3. Client input the directory path 4. System will check the existence of the directory 5. If the directory is existed, system will perform the detecting process. 6. Based on the redundant criterion which have been configured, the system will validate and display the result on the screen (refer to RFD-003 – View Progress)
Alternative Courses:	N/A
Exceptions:	1. Client select the "Detect redundant file" on the menu. 2. System request the client giving the path to the expected checking directory. 3. Client input the directory path 4. System will check the existence of the directory 5. If the directory is not existed or is not correct, system will display an error message.
Includes:	RFD-003 – View Progress RFD-005 – View detected result
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A



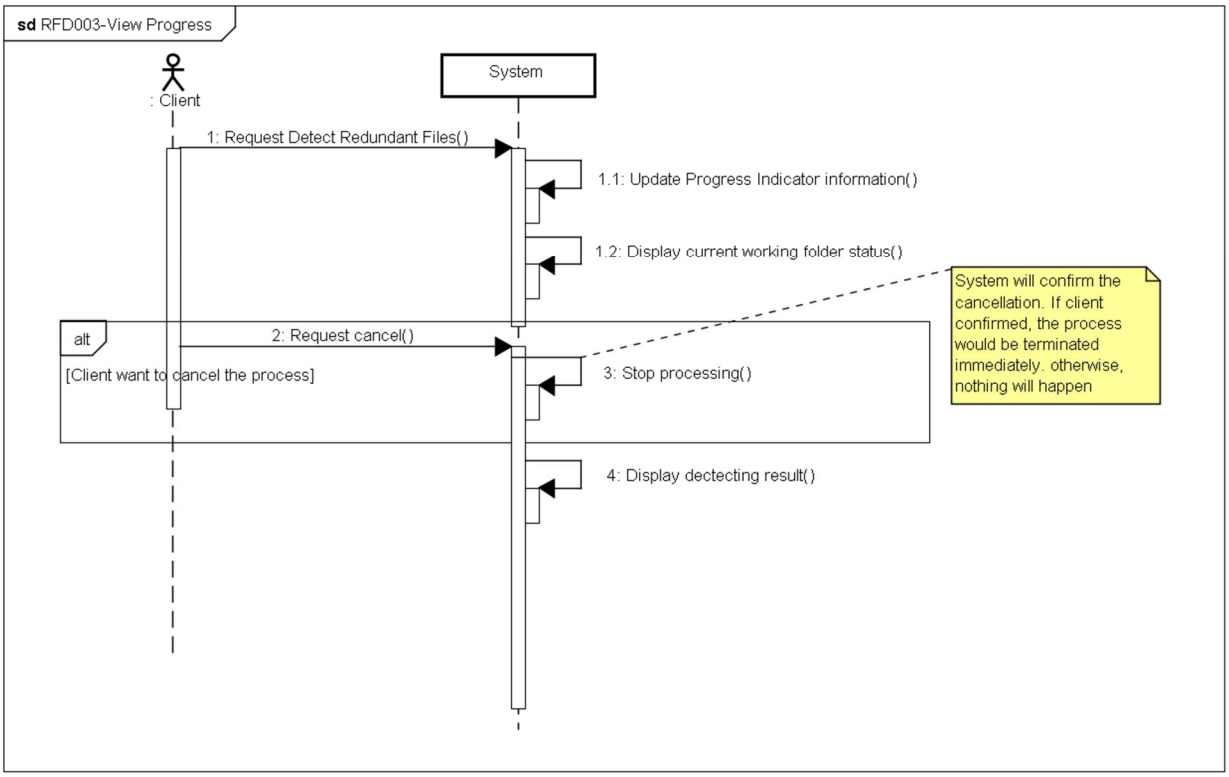
Use Case ID:	RFD-002		
Use Case Name:	Detect redundant file among many directories		
Created By:	DuongTB	Last Updated By:	DuongTB
Date Created:	2014/11/25	Date Last Updated:	2014/11/25
Actor:	Client		
Description:	Client request the system detecting all of redundant file in a list of directories and its sub-directories according to the redundant criterion which are configured in advance.		
Preconditions:	Client is working on the main screen		
Post-conditions:	All of redundant file are listed.		
Priority:	Normal		
Frequency of Use:	Often		
Normal Course of Events:	1. Client select the "Detect redundant file among many dictionaries" on the menu. 2. System request the client giving the path to the expected checking directory. 3. Client input the directory path 4. System lists the registered directories path, then asks client to add more directory or start detecting 5. If client want to start detecting. System will check the existence of the directories in the list 6. With each existed the directory, system will perform the detecting process. Otherwise, see the Exceptions case.		

	7. Based on the redundant criterion which have been configured, the system will validate and display the result on the screen (refer to RFD-003 – View Progress)
Alternative Courses:	1. Client select the "Detect redundant file among many directories" on the menu. 2. System request the client giving the path to the expected checking directory. 3. Client input the directory path 4. System asks client to add more directory or start detecting 5. If client want to add more directory, the step 2 to 4 will be repeated again. Otherwise, the main scenario is performed from step 5.
Exceptions:	1. With each not existed the directory, system will display an error message.
Includes:	RFD-003 – View Progress RFD-005 – View detected result
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A



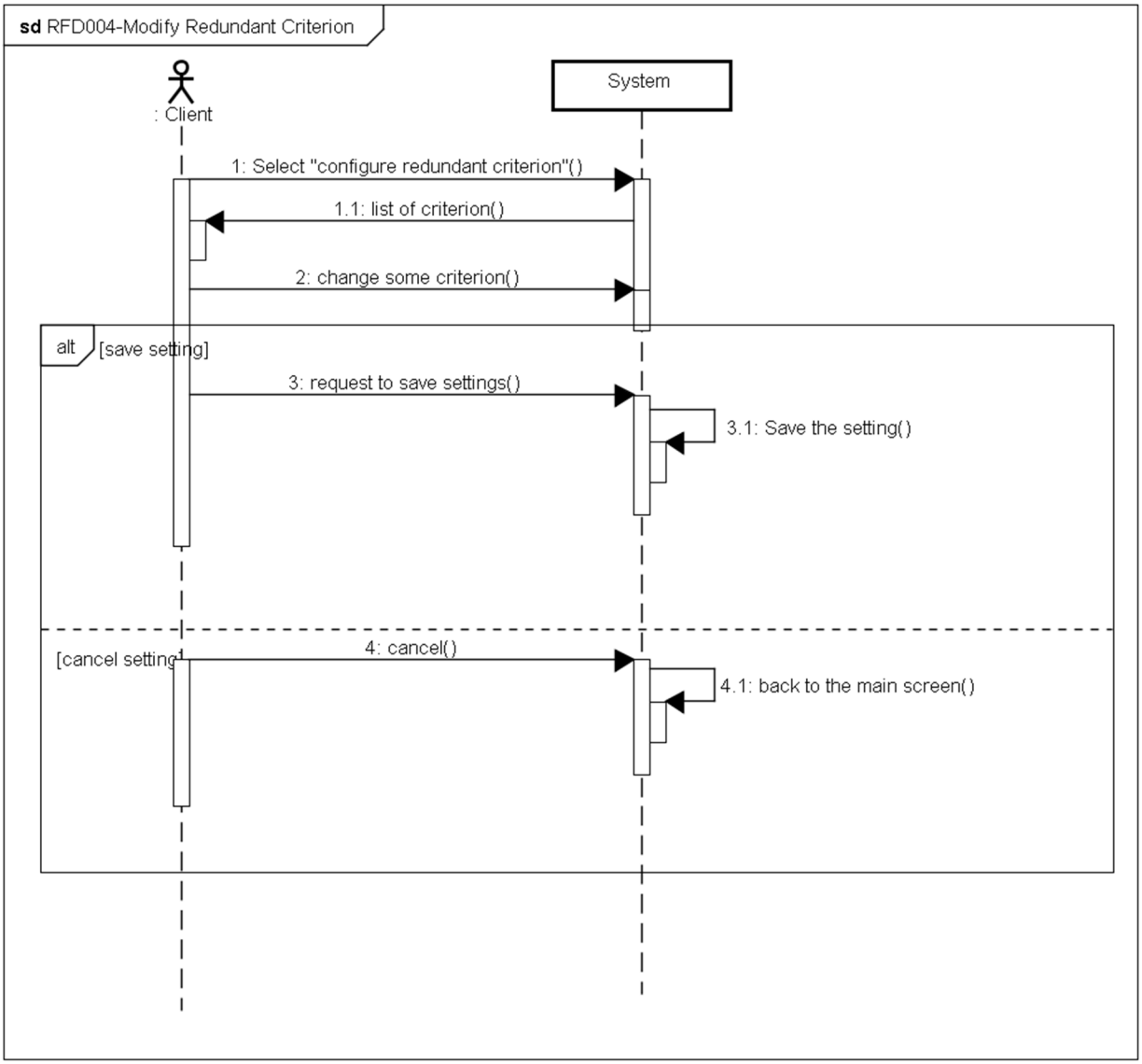
Use Case ID:	RFD-003		
Use Case Name:	View Progress		
Created By:	DuongTB	Last Updated By:	DuongTB
Date Created:	2014/11/25	Date Last Updated:	2014/11/25
Actor:	Client		
Description:	Client can see the progress of detecting process of the system.		
Preconditions:	Client chose detecting redundant files in a directory or in many directories		
Post-conditions:	The detecting progress is displayed and all redundant files		
Priority:	Normal		
Frequency of Use:	Always		

Normal Course of Events:	1. While system is processing, there is a progress indicator displayed on the screen. 2. Each time a redundant file is found, its full path will be listed on the screen. 3. In case, the multi directories detection mode had been chosen, the system will display the current working-in directory on the screen 4. If there is no more files to detect, user will be moved to RFD-005 – view detected result scenario.
Alternative Courses:	N/A
Exceptions:	1. If client cancel the process while it is processing, a terminate confirmation will be displayed to ask client confirm the cancellation. 2. If Client confirmed, the process will be terminated. 3. Then the system will display the detected result on the screen. (refer to RFD-005 – view detected result) 4. If Client didn't confirm, the process will be continued.
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A



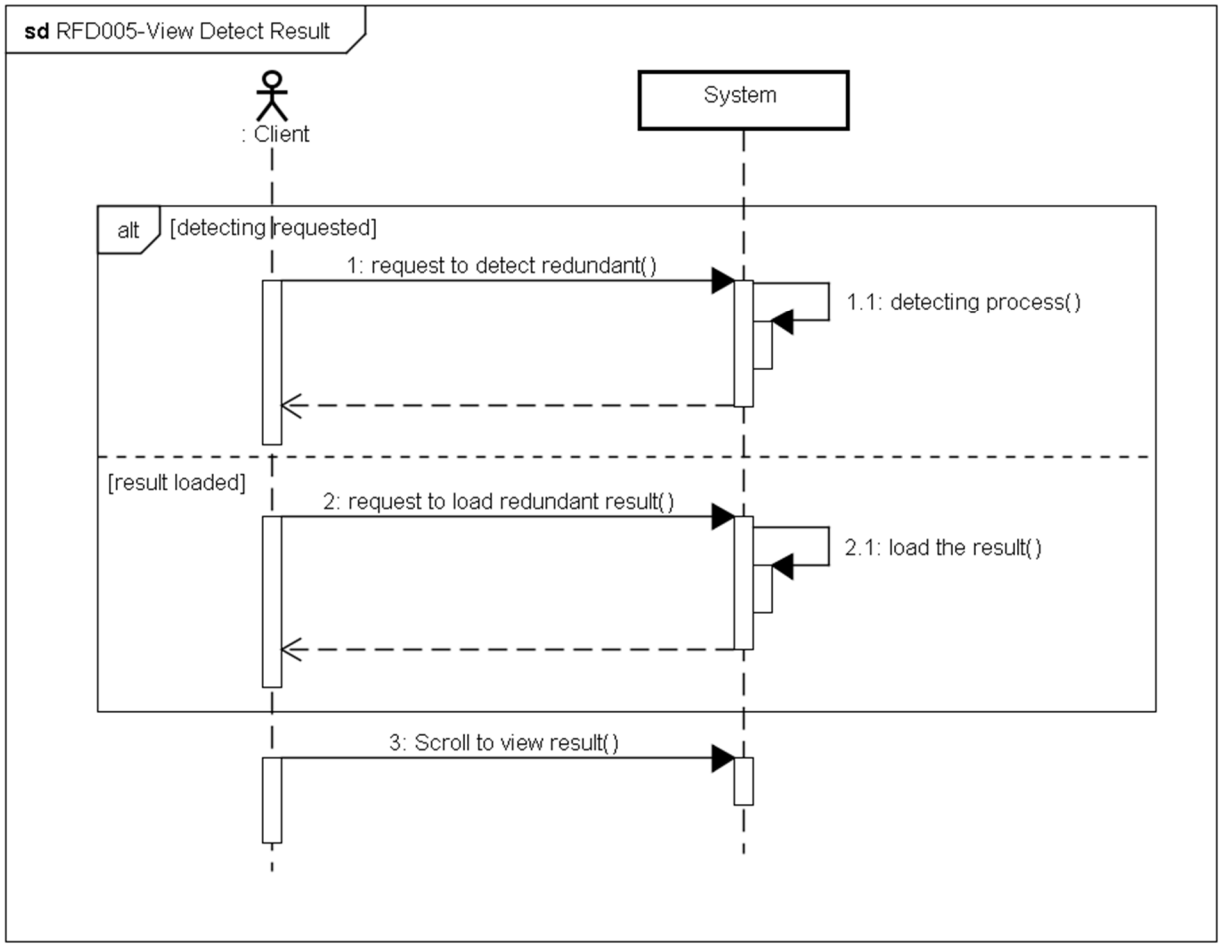
Use Case ID:	RFD-004

Use Case Name:	Modify redundant criterion		
Created By:	DuongTB	Last Updated By:	DuongTB
Date Created:	2014/11/25	Date Last Updated:	2014/11/25
Actor:	Client		
Description:	Client modify redundant criterion which will be used in redundancy detecting process.		
Preconditions:	System has been initialized successfully		
Post-conditions:	The client setting has been saved and it is apply immediately in the next detecting process.		
Priority:	Normal		
Frequency of Use:	Sometime		
Normal Course of Events:	1. Client selects on “configure redundant criterion” function. 2. System lists all criterion on the screen. 3. Client changes the setting value of each criteria which they want. 4. Client request to save the setting. 5. System will saved the updated setting and return to the main screen.		
Alternative Courses:	1. Client selects on “configure redundant criterion” function. 2. System lists all criterion on the screen. 3. Client changes the setting value of each criteria which they want. 4. Client return to the main screen by closing the configuration screen. 5. System ask client confirming whether to save the new setting or not 6. Client confirm to exit without save the setting. 7. System return to the main screen.		
Exceptions:	In case, the setting saving process raised exception, the system will notify by corresponding error messages.		
Includes:	N/A		
Special Requirements:	N/A		
Assumptions:	N/A		
Notes and Issues:	N/A		



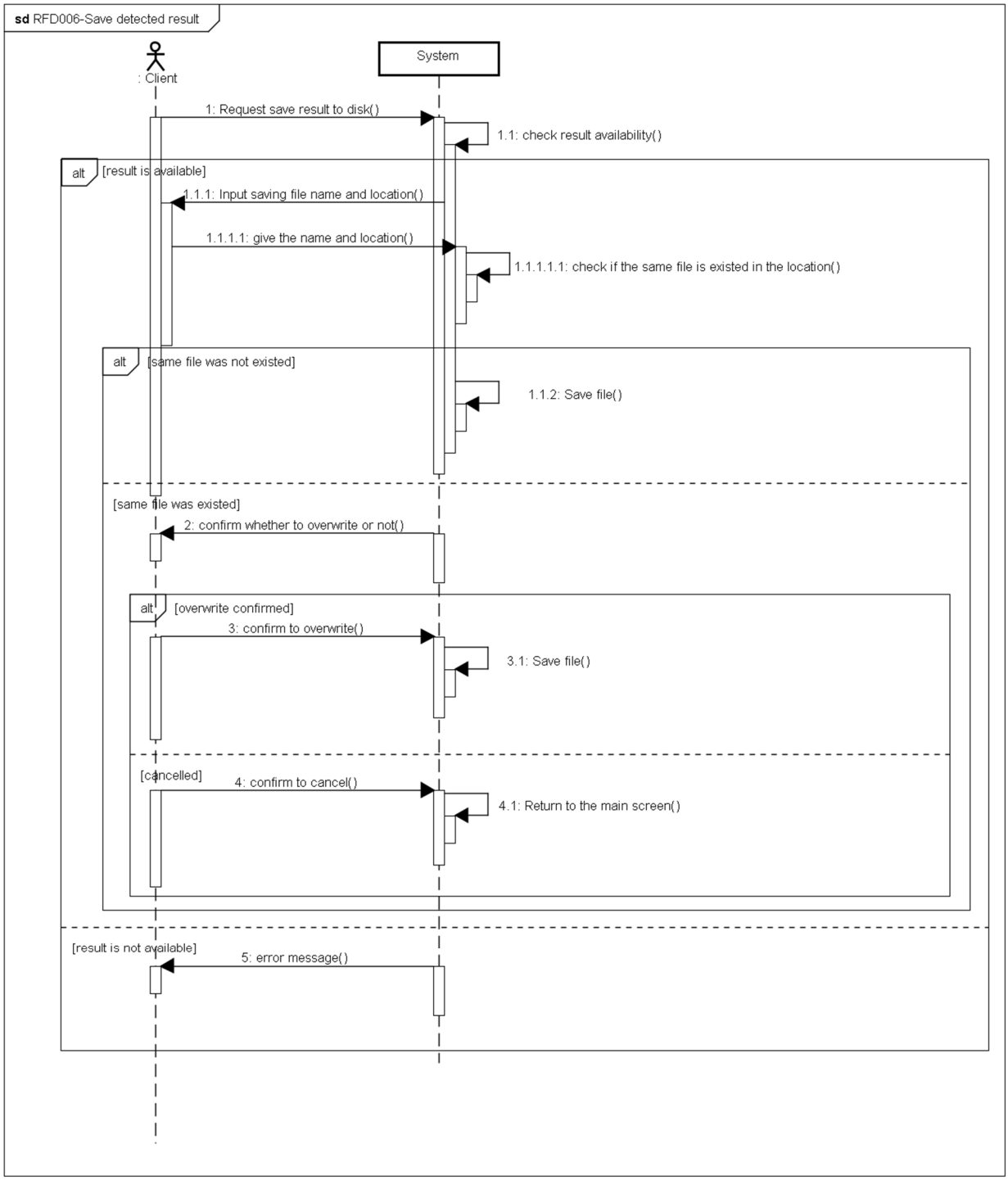
Use Case ID:	RFD-005		
Use Case Name:	View detected result		
Created By:	DuongTB	Last Updated By:	DuongTB
Date Created:	2014/11/25	Date Last Updated:	2014/11/25
Actor:	Client		
Description:	Client can view the detected result either right after detecting process completion or after it was loaded.		
Preconditions:	Client started detecting process or requested system to load the saved result.		
Post-conditions:	The result is displayed on the screen		
Priority:	Normal		
Frequency of Use:	Sometime		
Normal Course of Events:	1. System extract the data in the memory and display it on the screen		

	2. Client may use keyboard or mouse to scroll the result view to see it.
Alternative Courses:	N/A
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	



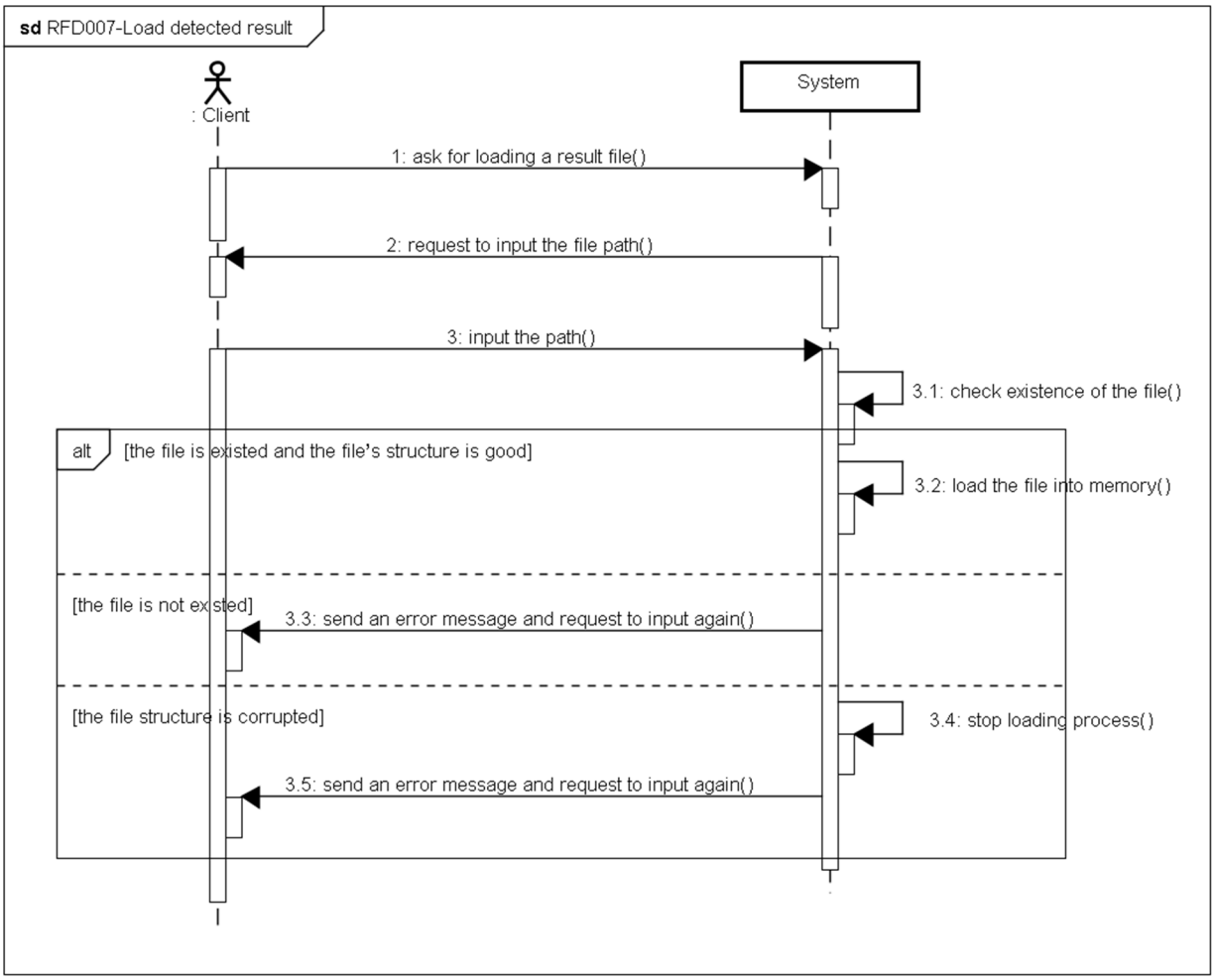
Use Case ID:	RFD-006		
Use Case Name:	Save detected result		
Created By:	DuongTB	Last Updated By:	DuongTB
Date Created:	2014/11/25	Date Last Updated:	2014/11/25
Actor:	Client		
Description:	Client can save the detected result to use it later.		
Preconditions:	Client started a detecting process or loaded a result data		
Post-conditions:	The memory-stored result is saved successfully.		
Priority:	Normal		
Frequency of Use:	Sometime		

Normal Course of Events:	1. Client request to save the current result in memory to disk. 2. System check the availability of the result in memory. 3. If the result is in the memory, System will request user inputting the location and the name of the saving file. 4. If the same file was not existed, the result would be saved into the file.
Alternative Courses:	1. If the result was not available in the memory, System would show an error message to notice the situation. 2. If the same file was existed, System would confirm to overwrite it or cancel the process with the client. 3. If Client accepted to overwrite the file, the result would be saved. 4. If the client cancel the process, the system will stop and display a notice message.
Exceptions:	It there is not enough space to save the result, the system will display a corresponding error message.
Includes:	N/A
Special Requirements:	The structure of the result file must be designed in the way which supports for version upgrading.
Assumptions:	N/A
Notes and Issues:	N/A



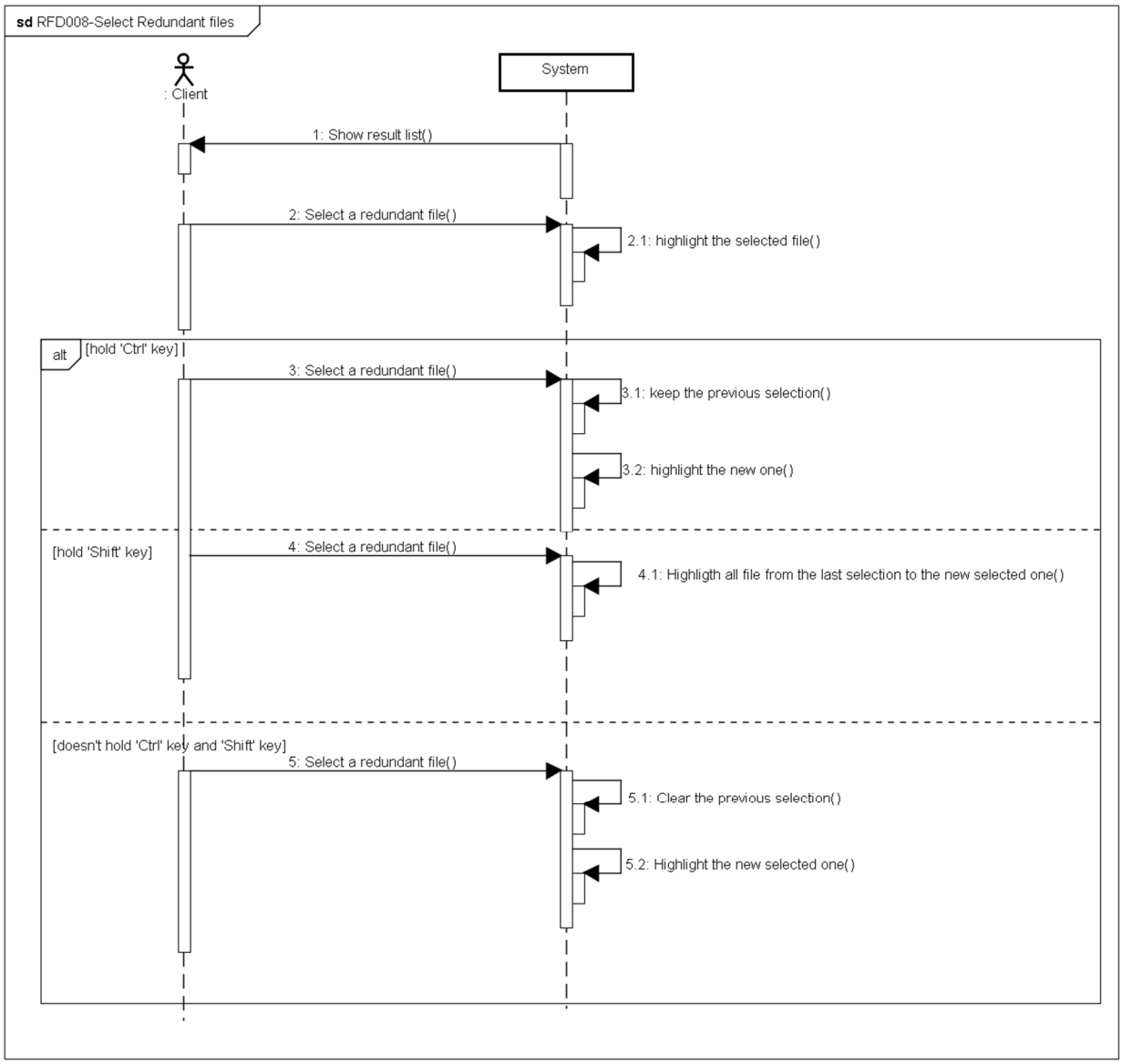
Use Case ID:	RFD-007		
Use Case Name:	Load detected result		
Created By:	DuongTB	Last Updated By:	DuongTB
Date Created:	2014/11/25	Date Last Updated:	2014/11/25
Actor:	Client		
Description:	Client load a saved result from disk to continue view and delete		

	redundant files.
Preconditions:	There is a saved result to load
Post-conditions:	The result is stored in memory so that the client can view the loaded result by performing Use-case RFD-005
Priority:	Normal
Frequency of Use:	Sometime
Normal Course of Events:	1. Client asks for loading a result file 2. System request the client input the path to that file. 3. Client inputs the path or browses to the file 4. System checks the existence of the file. 5. If the file is existed and the file's structure is good, system will load it into the memory.
Alternative Courses:	6. If the file is not existed, system will display a warning message and request client inputting another path.
Exceptions:	7. If the file's structure is corrupted, system will display an error message and the current loading process will be stopped. 8. Then system will request client inputting another path.
Includes:	N/A
Special Requirements:	System can load any saved result from the previous version of the system (backward compatible)
Assumptions:	N/A
Notes and Issues:	N/A



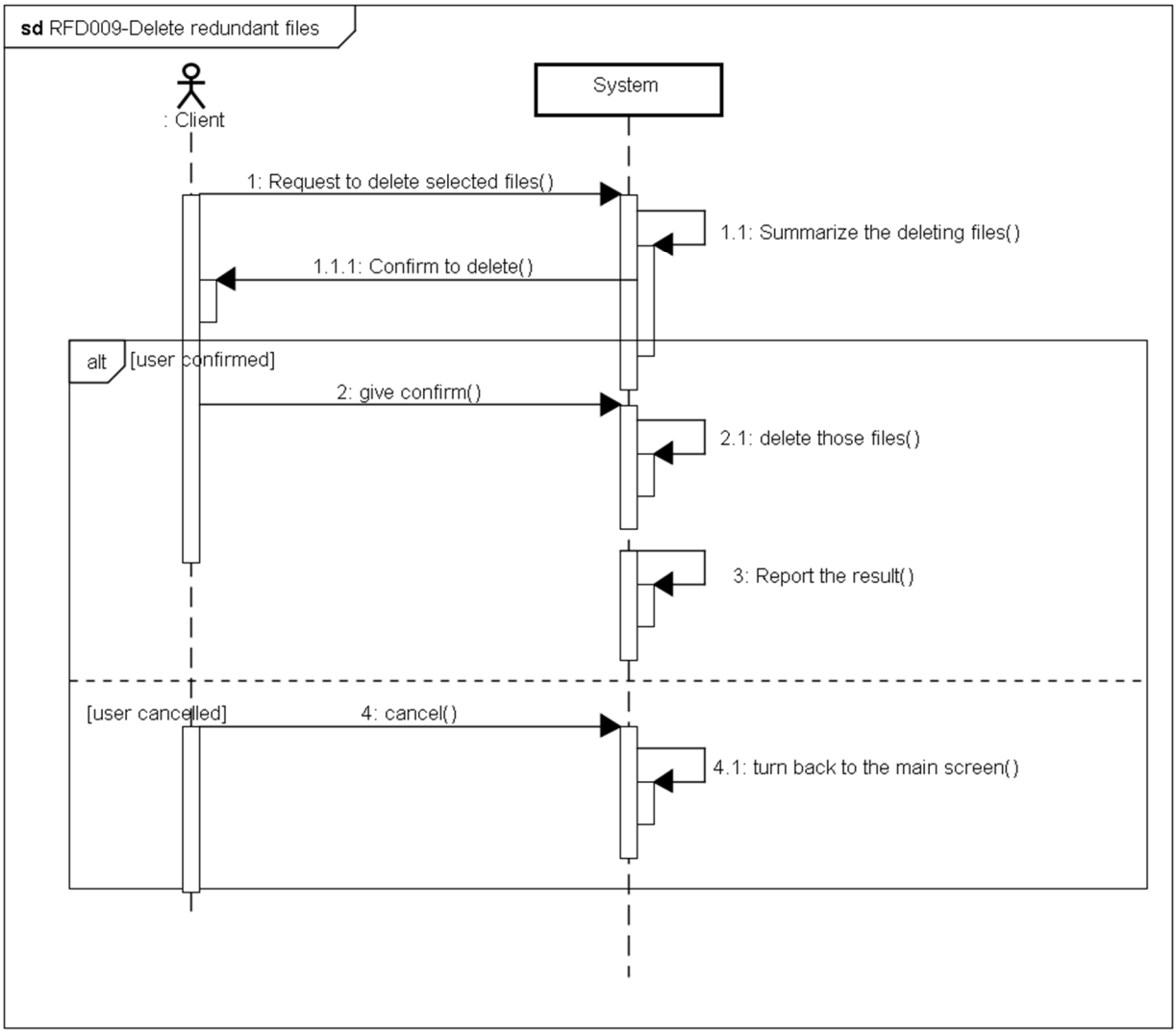
Use Case ID:	RFD-008		
Use Case Name:	Select Redundant files		
Created By:	DuongTB	Last Updated By:	DuongTB
Date Created:	2014/11/25	Date Last Updated:	2014/11/25
Actor:	Client		
Description:	Client can select one or many redundant files.		
Preconditions:	There is a redundant result in memory (the last redundant result of a detecting process or the result has been loaded into the memory)		
Post-conditions:	Selected files must be highlighted		
Priority:	Normal		
Frequency of Use:	Sometime		
Normal Course of Events:	1. Client select a redundant file in the result list. 2. System highlight the file. 3. Client hold 'Ctrl' key and select another file 4. The previous selection is kept highlight and the file will be highlighted too.		
Alternative Courses:	5. If Client doesn't hold 'Ctrl' key and select another file.		

	6. The previous selection will be clear and the file will be highlighted 7. If Client hold 'Shift' key while selecting files, All files in the range from the last selected point to new one will be highlighted
Exceptions:	There isn't any redundant files in the list, The client can't select anything.
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A



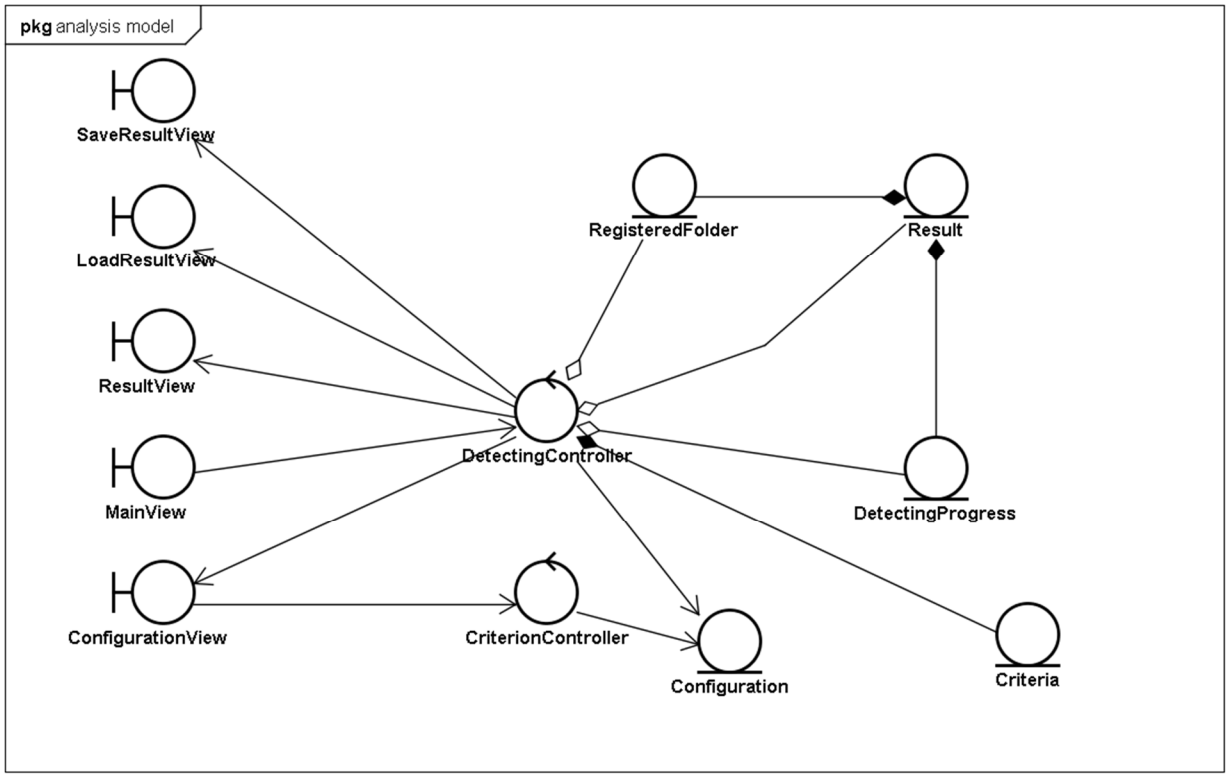
Use Case ID:	RFD-009

Use Case Name:	Delete redundant files		
Created By:	DuongTB	Last Updated By:	DuongTB
Date Created:	2014/11/25	Date Last Updated:	2014/11/25
Actor:	Client		
Description:	Client can delete redundant files		
Preconditions:	There is at least one file selected.		
Post-conditions:	The selected file(s) has been removed from the list and from the disk.		
Priority:	Normal		
Frequency of Use:	Sometime		
Normal Course of Events:	1. Client requests to delete the selected redundant files 2. System summarizes the number of files which client would like to delete and confirms one again before deleting them. 3. If the Client confirms, the system will delete those files.		
Alternative Courses:	If the Client cancels the work, the system will stop the process.		
Exceptions:	If there is any exception occurred during the deletion, the system will notify to client to confirm. There are 3 options in the confirmation, Cancel, Ignore or Try Again.		
Includes:	N/A		
Special Requirements:	N/A		
Assumptions:	N/A		
Notes and Issues:	N/A		



**Note: N/A is the shortcut of Not Available.*

2. Analysis Class Diagram



3. Class Responsibility and Collaboration

MainView	
Responsibilities	Collaboration
+ displays menu option for user interaction + handles detecting folders registering + knows DetectingController	Detecting Controller

ConfigurationView	
Responsibilities	Collaboration
+ displays configuration settings + handles setting change request + knows DetectingController	Detecting Controller

ResultView	
Responsibilities	Collaboration
+ displays last result + handles selecting redundant file request + handles deleting redundant file request + knows DetectingController	Detecting Controller

LoadResultView

Responsibilities	Collaboration
+ handles browse saved result request + displays loading progress + handles request back to MainView + knows Detecting Controller	Detecting Controller

SaveResultView	
Responsibilities	Collaboration
+ handles browse folder to save result request + displays saving progress + handles request back to MainView + knows Detecting Controller	Detecting Controller

CriterionController	
Responsibilities	Collaboration
+ knows Configuration + knows configuration view + handles get configuration information + handles set configuration information	Configuration ConfigurationView

DetectingController	
Responsibilities	Collaboration
+ knows main screen + knows detection result displaying screen + knows detection result saving screen + knows detection result loading screen + knows configuration screen + knows registered detecting folders + produces detecting result + knows detecting criterion + knows detecting progress + handles event transferred from views + updates the configuration criterion + updates the progress information on main screen	MainView ResultView SaveResultView LoadResultView ConfigurationView RegisteredFolders Result Criteria DetectingProgress MainView, ResultView, SaveResultView, LoadResultView, ConfigurationView Criteria MainView, DetectionProgress

RegisteredFolder	
Responsibilities	Collaboration
+ Stores registered folder information + Stores proceed status	

Result

Responsibilities	Collaboration
+ Stores the list of redundant files structure + Stores the detected progress when the system create the result + Stores the list of registered folder when the system is requested to detect + Stores the redundant criterion at the detecting time	

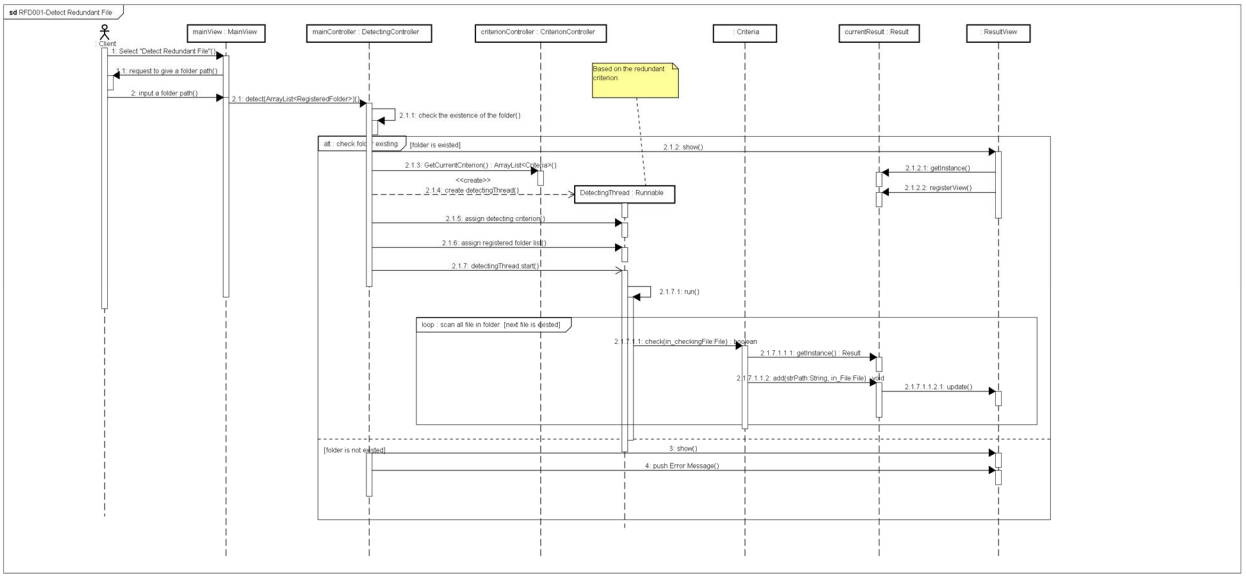
Configuration	
Responsibilities	Collaboration
+ Stores a configuration setting	

Criteria	
Responsibilities	Collaboration
+ Process checking according to a criteria	

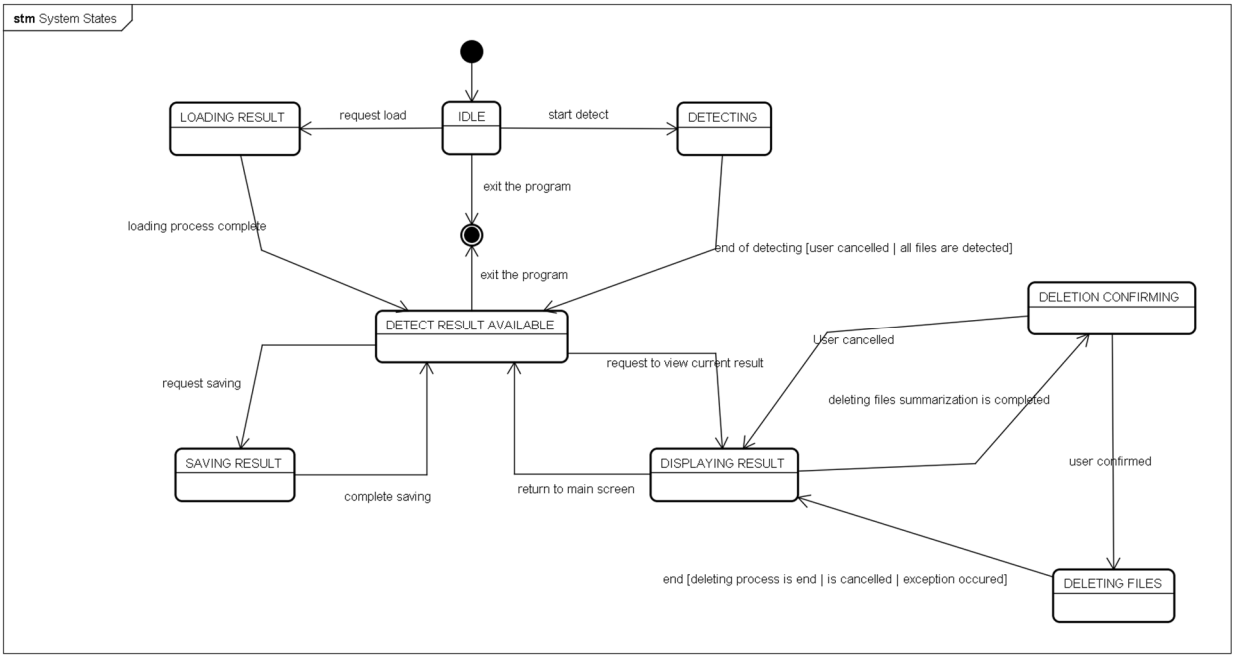
DetectingProgress	
Responsibilities	Collaboration
+ Stores progress data + provides progress data	

4. Interactive Diagram

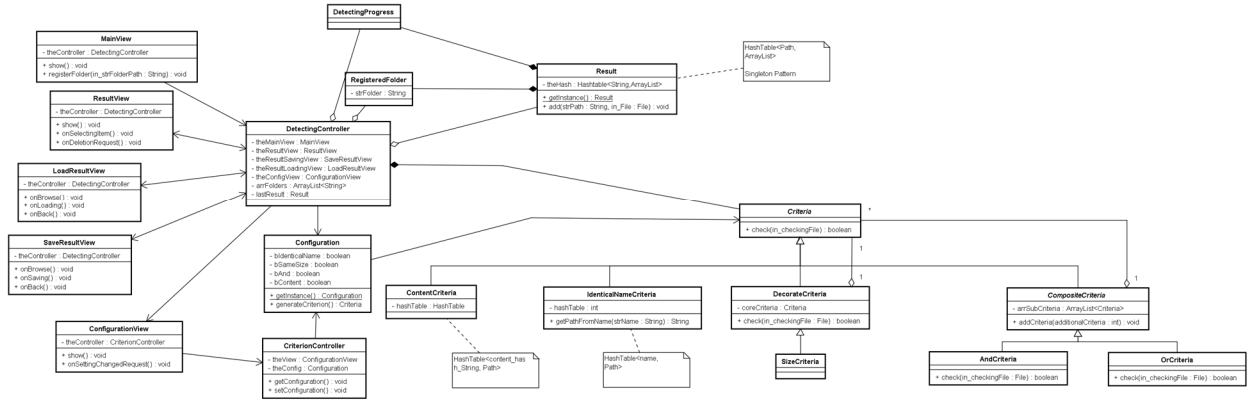
a. RFD001-Detect Redundant File



5. State-Machine diagram



6. Class Specification



MainView			
Attributes			
Name	Type	Initialize	Description
theController	DetectingController		Reference to the controller
Methods			
Method Name	show	Scope	public
Return	Return Type	Description	
N/A	None	displays menu option for user interaction	
Parameter	Type	Default Value	
N/A	-		
Method Name	registerFolder	Scope	public
Return	Return Type	Description	
N/A	None	handles detecting folders registering	
Parameter	Type	Default	

		Value	
strFolderPath	String		The registered folder path

ConfigurationView			
Attributes			
Name	Type	Initialize	Description
theController	DetectingController		Reference to the controller
Methods			
Method Name	show	Scope	public
Return	Return Type	Description	
N/A	None	displays configuration settings	
Parameter	Type	Default Value	
N/A	-		
Method Name	onSettingChangedRequest	Scope	public
Return	Return Type	Description	
N/A	None	handles setting change request	
Parameter	Type	Default Value	
N/A			

ResultView			
Attributes			
Name	Type	Initialize	Description
theController	DetectingController		Reference to the controller
Methods			
Method Name	show	Scope	public
Return	Return Type	Description	
N/A	None	displays last result	
Parameter	Type	Default Value	
N/A	-		
Method Name	onSelectingItem	Scope	public
Return	Return Type	Description	
N/A	None	handles selecting redundant file request	
Parameter	Type	Default Value	
N/A			
Method Name	onDeletionRequest	Scope	public
Return	Return Type	Description	
N/A	None	handles deleting redundant file request	
Parameter	Type	Default Value	
N/A			

LoadResultView

Attributes			
Name	Type	Initialize	Description
theController	DetectingController		Reference to the controller
Methods			
Method Name	onBrowse	Scope	public
Return	Return Type	Description	
N/A	None	handles browse saved result request	
Parameter	Type	Default Value	
N/A	-		
Method Name	onLoading	Scope	public
Return	Return Type	Description	
N/A	None	displays loading progress	
Parameter	Type	Default Value	
N/A			
Method Name	onBack	Scope	public
Return	Return Type	Description	
N/A	None	handles request back to MainView	
Parameter	Type	Default Value	
N/A			

SaveResultView			
Attributes			
Name	Type	Initialize	Description
theController	DetectingController		Reference to the controller
Methods			
Method Name	onBrowse	Scope	public
Return	Return Type	Description	
N/A	None	handles browse folder to save result request	
Parameter	Type	Default Value	
N/A	-		
Method Name	onSaving	Scope	public
Return	Return Type	Description	
N/A	None	displays saving progress	
Parameter	Type	Default Value	
N/A			
Method Name	onBack	Scope	public
Return	Return Type	Description	
N/A	None	handles request back to MainView	
Parameter	Type	Default Value	

N/A			

CriterionController			
Attributes			
Name	Type	Initialize	Description
theView	ConfigurationView		Reference to the View
theConfig	Configuration		Reference to the Configuration
Methods			
Method Name	getConfiguration	Scope	public
Return	Return Type	Description	
N/A	None	handles get configuration information	
Parameter	Type	Default Value	
N/A	-		
Method Name	setConfiguration	Scope	public
Return	Return Type	Description	
N/A	None	handles set configuration information	
Parameter	Type	Default Value	
N/A			

RegisteredFolder			
Attributes			
Name	Type	Initialize	Description
theFolderPath	String		Stores registered folder information
isDetected	boolean		Stores the proceed status

Result			
Attributes			
Name	Type	Initialize	Description
theHash	Hashtable<String, ArrayList>		Stores the list of redundant files structure
theDetectedProgress	Progress		Stores the detected progress when the system create the result
theRegisteredFolder	ArrayList<RegisteredFolder>		Stores the list of registered folder when the system is requested to detect
theCriterionConfiguration	Configuration		Stores the redundant criterion at the

			detecting time
Methods			
Method Name	getConfiguration	Scope	public
Return	Return Type	Description	
N/A	None	handles get configuration information	
Parameter	Type	Default Value	
N/A	-		
Method Name	setConfiguration	Scope	public
Return	Return Type	Description	
N/A	None	handles set configuration information	
Parameter	Type	Default Value	
N/A			

Criteria			
Attributes			
Name	Type	Initialize	Description
Methods			
Method Name	check	Scope	public
Return	Return Type	Description	
N/A	None	Process checking according to a criteria	
Parameter	Type	Default Value	
N/A	-		

DetectingProgress			
Attributes			
Name	Type	Initialize	Description
numberOfScannedFiles	Integer		The number of scanned files
numberOfRedundantFiles	Integer		The number of redundant files
percentage	Integer		The progress in percentage
Methods			
Method Name	setXYZ	Scope	public
Return	Return Type	Description	
XYZ attribute value	Integer	Stores progress data	
Parameter	Type	Default Value	
N/A	-		

Method Name	getXYZ	Scope	public
Return	Return Type	Description	
XYZ attribute value	Integer	provides progress data	
Parameter	Type	Default Value	
N/A			

DetectingController			
Attributes			
Name	Type	Initialize	Description
theMainView	MainView		Reference to main screen
theResultView	ResultView		Reference to detection result displaying screen
theResultSavingView	SaveResultView		Reference to detection result saving screen
theResultLoadingView	LoadResultView		Reference to detection result loading screen
theConfigView	ConfigurationView		Reference to configuration screen
arrFolders	ArrayList<String>		Registered detecting folders collection
lastResult	Result		Reference to the last proceed result
currentCriterion	Criteria		Reference to the current detecting criterion
progress	DetectingProgress		Reference to the current progress
Methods			
Method Name	detect	Scope	public
Return	Return Type	Description	
N/A	None	produces detecting result	
Parameter	Type	Default Value	
N/A	-		
Method Name	onXYZ	Scope	public
Return	Return Type	Description	
N/A	None	handles event transferred from views	
Parameter	Type	Default Value	
N/A			
Method Name	updateProgress	Scope	public
Return	Return Type	Description	
N/A	None	updates the progress information on main screen	
Parameter	Type	Default Value	
N/A			
Method Name	updateCriterion	Scope	public

Return	Return Type	Description	
N/A	None	updates the configuration criterion	
Parameter	Type	Default Value	
N/A			