

DR Reporting Application Architecture Document

Revision History

Date	Version	Description	Modified by
24-11-2023	1.0	DR Reporting Application Architecture – Initial draft	P Narayan
08-12-2023	1.1	Re-formatted document as per MDIForm menu form order	Sneha K
12-12-2023	1.2	Completed re formatting of document. In-corporated John's comments	Sneha K
14-12-2023	1.3	Updates of Review comments from John	Sneha K
18-12-2023	1.4	Updated latest diagrams with readable font, orientation	Sahil D
19-12-2023	1.41	Added App.config file section	P Narayana

Table of Contents

1 Deployment View	7
1.1 System Specification/ Pre-Requisite	7
1.2 Architecture	8
1.2.1 Deployment Architecture	8
1.2.2 Technical Architecture	9
1.2.3 High Level Flow Diagram for DRReporting Application	10
2 Development	11
2.1 Solution Design	11
2.1.1 Solution – JMMB DRReporting Application.....	11
3 Application Menu	13
3.1 Reports	13
3.2 REIT (Real Estate Investment Trust).....	13
3.3 Maintenance	14
4 Technical Design Details	16
4.1 DR Reporting Application - Form Authentication	16
Code level details for Authentication(frmAuth)	16
4.1.1 FrmAuth - Ok button click.....	16
4.2 DR Reporting Application – MDI Form Parent Menu.....	19
Code level details for MDIParent Form.....	20
4.3 DR Reporting Application – Statements Report.....	26
Code level details for Statements report	26
4.3.1 Statement Report – Form Load	27
4.3.2 Statement Report – Build button click.....	29
4.3.3 Statement Report – Email button click	31
4.3.4 Statement Report – Preview button click.....	35
4.3.5 Client Statement Report – Build button click	36
4.3.6 Client Statement Report – Preview button click	37
4.4 DR Reporting Application – Deal Slips Report.....	42
Code level details for Deal Slips.....	42
4.4.1 Deal Slips – Form Load.....	43
4.4.2 Deal Slips – Build button click	45
4.4.3 Deal Slips – Email button click.....	47

4.4.4 Deal Slips – Preview button click.....	49
4.4.5 Deal Slips – GetTransaction button click	50
4.4.6 DomRepDealSlip report flow – Build button click.....	52
4.5 DR Reporting Application – Regulatory report	58
Code level details for Regulatory Reports	58
4.5.1 Regulatory Report – Form Load.....	58
4.5.2 Regulatory Report – Build button click	61
4.5.3 Regulatory Report – Preview button click	64
4.5.4 Regulatory Report – Workbook button click	66
4.5.5 InformacionDiariaDePublicar Report – Build button click	69
4.5.7 InformacionDiariaDePublicar Report – Workbook button click	71
4.6 DR Reporting Application – Balances report.....	74
Code level details for Balances report	74
4.6.1 Balances Report – Form Load.....	75
4.6.2 Balances Report – Build button click.....	77
4.6.3 Balances Report – Email button click	80
4.6.4 Balances Report – Preview button click	82
4.6.5 Balances (FondoMutuoRentabilidad) Report – Build button click.....	84
4.6.6 Balances (FondoMutuoRentabilidad) Report – Preview button click.....	85
4.7 DR Reporting Application – AFP Statements.....	88
Code level details for AFP Statements	88
4.7.1 AFP Statements – Form Load Action	89
4.7.2 AFP Statements – Build button click.....	92
4.7.3 AFP Statements – Email button click	94
4.7.4 AFP Statements – Preview button click.....	97
4.7.5 AFP Statement Report – Build button click	98
4.7.6 AFP Statement Report – Preview button click	99
4.8 DR Reporting Application – REIT Upload Tool.....	102
Code level details for REIT Upload Tool	102
4.8.1 REIT Upload Tool – Load Form	102
4.8.2 REIT Upload Tool- Import Button Click.....	104
4.8.3 REIT – Send File button click.....	106
4.8.4 REIT – Clear Batch button click.....	108
4.9 DR Reporting Application – View Log.....	111
Code level details for View Log.....	111
4.9.1 ViewLog – Form Load action	112
4.9.2 ViewLog – Refresh button click.....	113

5 DR Reporting Application – Email Services.....	116
Code Level Details for Email Services	116
6 Data flow design.....	119
6.1 ER Diagrams	119
6.1.2 T24ReportingDB ER Diagram.....	121
7 Reports	122
7.1 Client Statement	122
7.1.1 The SSRS data diagram of the Client Statement report.....	123
7.1.2 The SSRS Code diagram of the Client Statement report.....	125
7.2 Fondo Mutuo Rentabilidad	125
7.2.1 The SSRS data diagram of the Fondo Mutuo Rentabilidad report.....	126
7.2.2 The SSRS Code diagram of the Fondo Mutuo Rentabilidad report.....	128
7.3 Fondo Plazo90 Statement	129
7.3.1 The SSRS data diagram of the Fondo Plazo90 Statement report.....	129
7.3.2 The SSRS Code diagram of the Fondo Plazo90 Statement report	131
7.4 AFP Statement.....	133
7.5 InformacionDiariaDePublicar	135
7.5.1 The SSRS data diagram of the InformacionDiariaDePublicar report	135
7.5.2 The SSRS Code diagram of the InformacionDiariaDePublicar.....	137
7.6 Limitede Participaciony Relacionados.....	137
7.6.1 The SSRS data diagram of the Limitede Participaciony Relacionados report.....	137
7.6.2 The SSRS Code diagram of the Limitede Participaciony Relacionados report	138
7.7 Statement.....	139
7.7.1 The SSRS data diagram of the Statement report	139
The SSRS data diagram of the Statement (FundoMercadoDineroStatement.rdl) report describes datasets, parameters, images and datasource used by the report	139
7.7.2 The SSRS Code diagram of the Statement	141
7.8 Dealslips	142
7.8.1 The SSRS data diagram of the Dealslips report.....	142
7.8.2 The SSRS Code diagram of the Dealslips	144
7.9 Regulatory	144
7.9.1 The SSRS data diagram of the Regulatory report.....	144
7.9.2 The SSRS Code diagram of the Regulatory report.....	146
7.10 PortafolioDeInversiones.....	146
7.10.1 The SSRS data diagram of the PortafolioDeInversiones report	146
7.11 ReporteDeLiquidez.....	148
7.11.1 The SSRS data diagram of the ReporteDeLiquidez report:	148
7.11.2 The SSRS Code diagram of the ReporteDeLiquidez	150

7.12 SuscripcionRescatesCuotas	150
7.12.1 The SSRS data diagram of the SuscripcionRescatesCuotas report	150
7.12.2 The SSRS Code diagram of the SuscripcionRescatesCuotas.....	151
7.13 DOMRepDealSlip	151
7.13.1 The SSRS data diagram of the DOMRepDealSlip report	151
7.13.2 The SSRS Code diagram of the DOMRepDealSlip.....	153
9 Exception Handling.....	153
10 Assumptions.....	153
11 Exception Handling:.....	153
12 Exclusion	153
13 Libraries	154
13.1 Internal Libraries	154
13.2 External Libraries.....	154
13.3 .Net Libraries	154
14. App.Config file	157
14.1 Configurations	157

1 Deployment View

1.1 System Specification/ Pre-Requisite

System Specifications and prerequisites needed for the DRReporting application are:

1. Dotnet framework (Version 4.5)
2. SQL Server (version 14.0.3436)
3. RDP Server (version Microsoft Windows Server 2012)
4. SSRS Reporting Server (version 11.0.7507)
5. Internet Information Server
6. SMTP Server

1.2 Architecture

1.2.1 Deployment Architecture

Deployment Architecture depicts the mapping of a logical architecture to a physical environment. The physical environment includes the computing nodes in an intranet or Internet environment, CPUs, memory, storage devices, and other hardware and network devices. The below diagram represents the Deployment Architecture of the application.

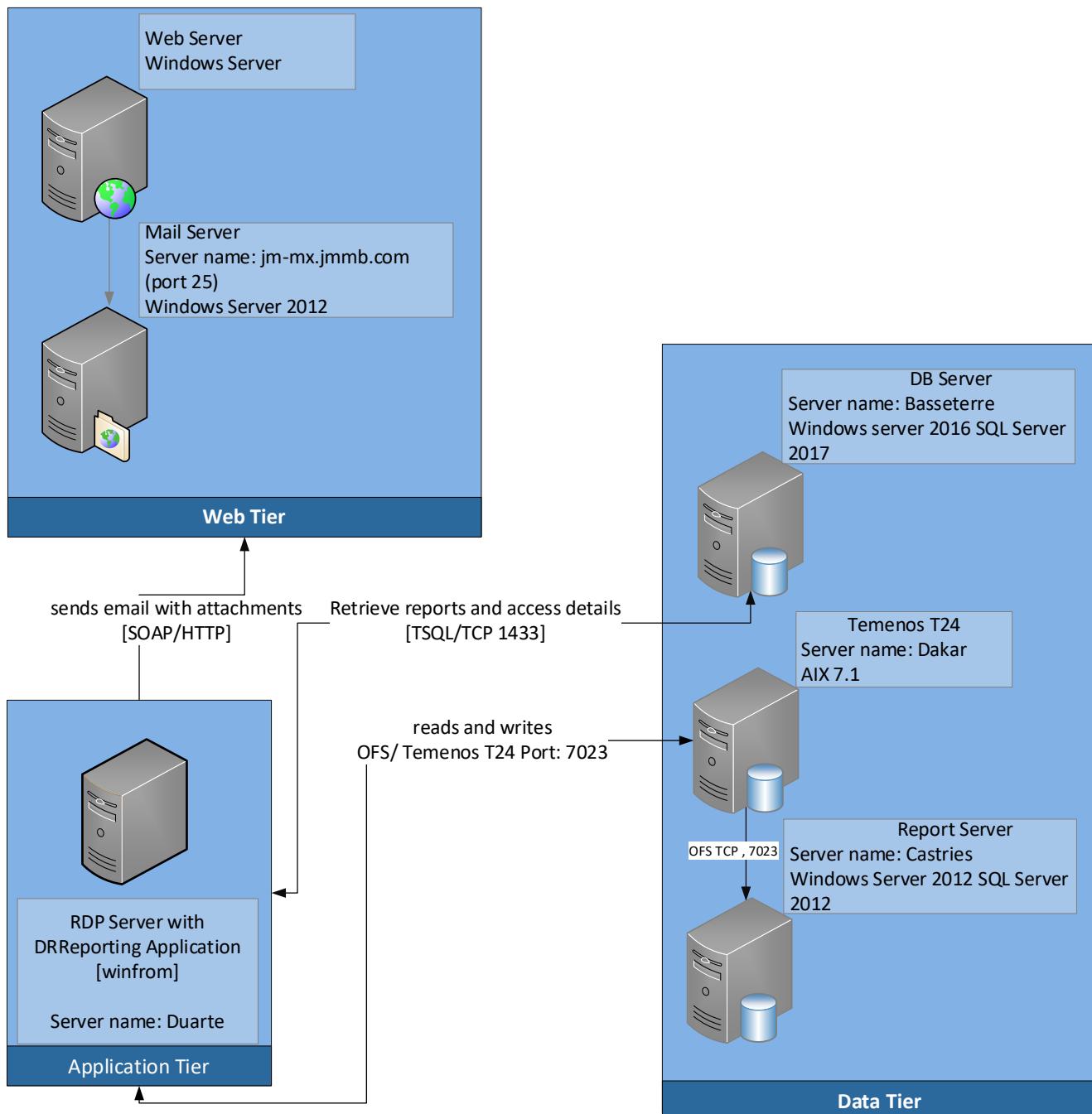


Figure 1: DRReporting - Deployment Architecture diagram

1.2.2 Technical Architecture

Technical Architecture refers to the design of the application itself. A Technical Architecture diagram provides a brief of the infrastructure, and it also illustrates how components in a system interact with each other on a larger scale. The below diagram explains the technical architecture of the DRReporting application:

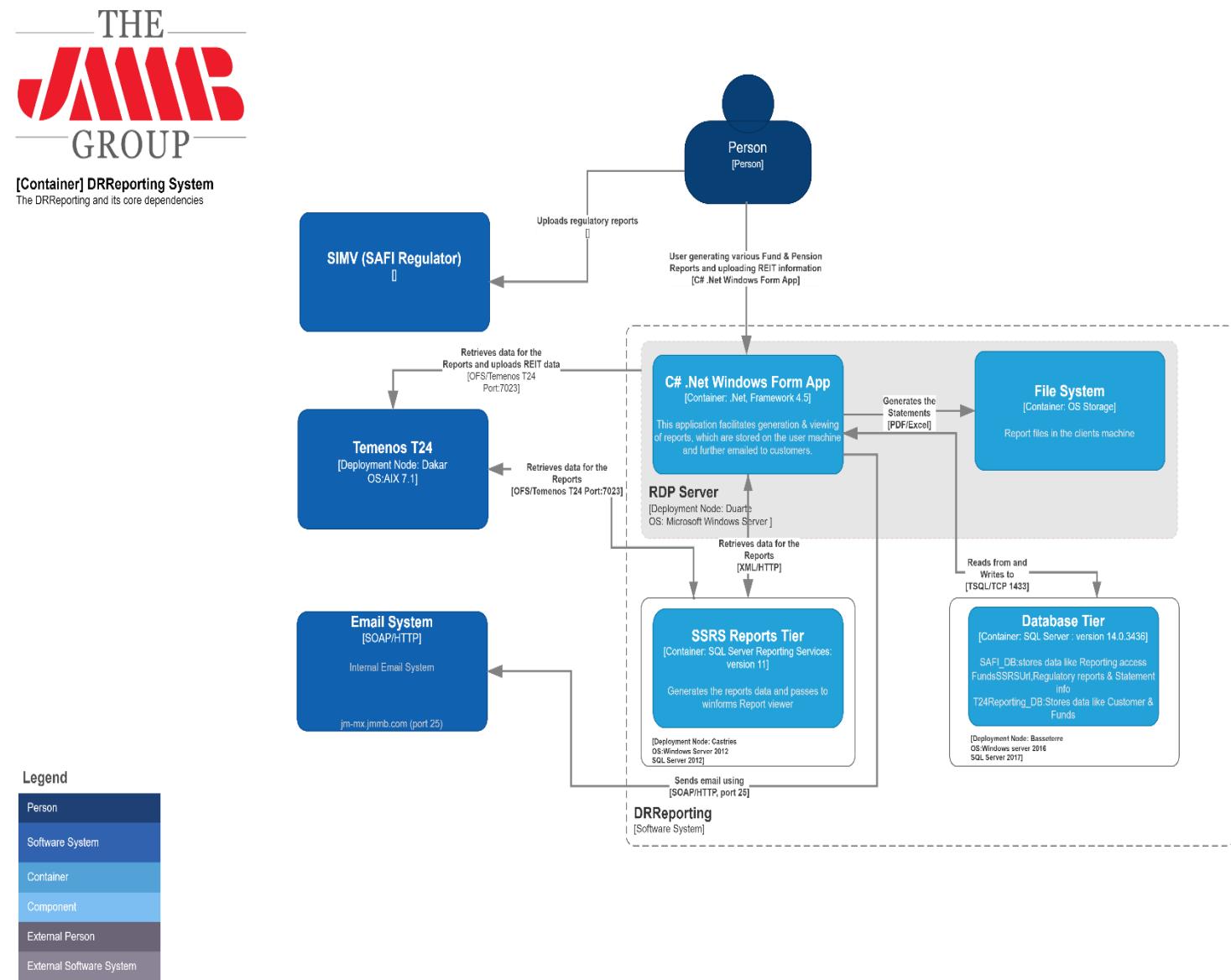


Figure 1: DRReporting - Technical Architecture (C4Level2) diagram

1.2.3 High Level Flow Diagram for DRReporting Application

This Diagram gives a high-level view of the application used for DR Reporting. It also displays the system components being used for each form.



Figure 2: DRReporting: Forms representation

2 Development

2.1 Solution Design

2.1.1 Solution – JMMB DRReporting Application

JMMB DRReporting is designed to run on Microsoft Windows operating systems. Windows applications typically have a graphical user interface (GUI) and can take advantage of features specific to the Windows platform. JMMB DRReporting Windows Application is built using C# .NET (Framework 4.5).

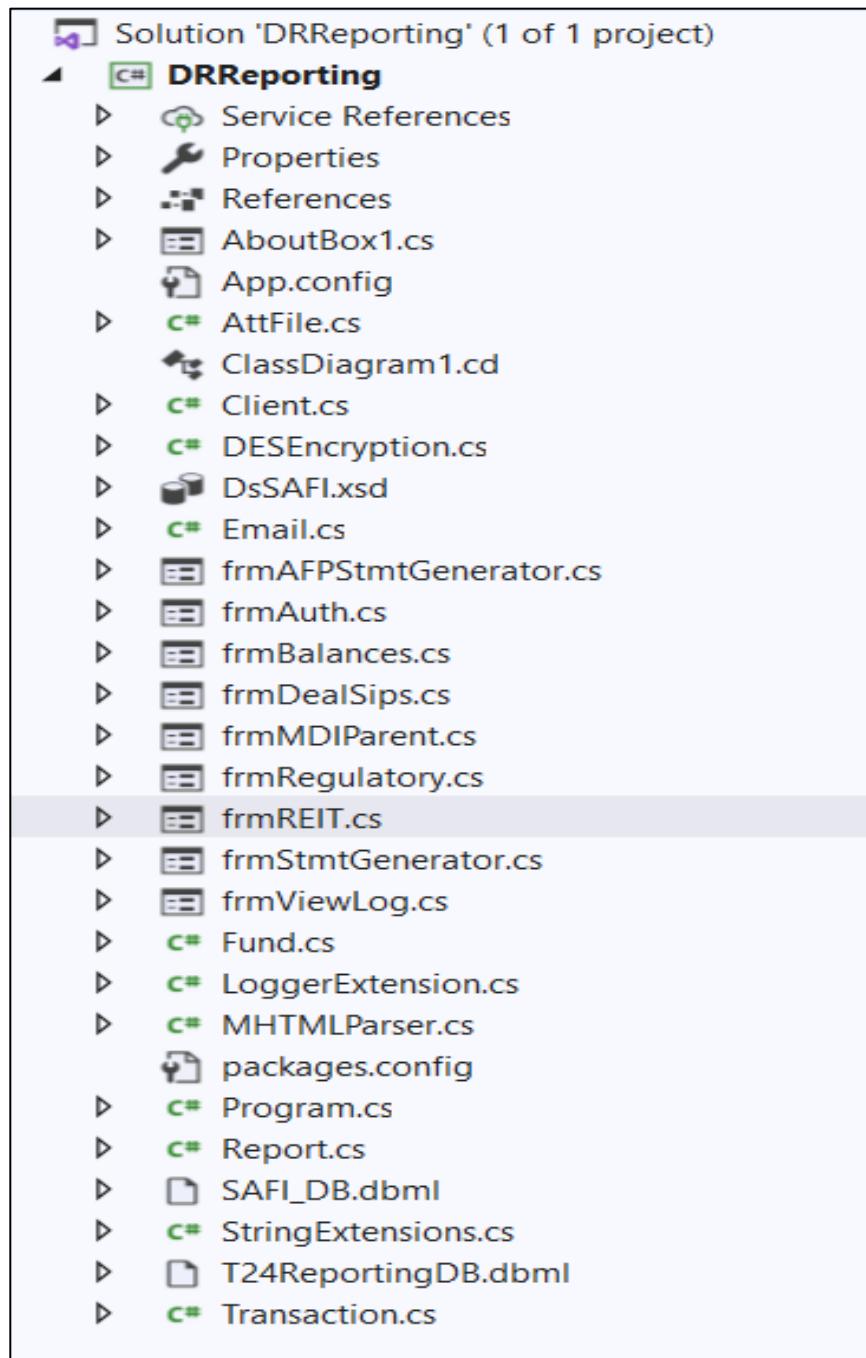


Figure 3: DRReporting - Solution view

DR Reporting application Solution and Project representation-

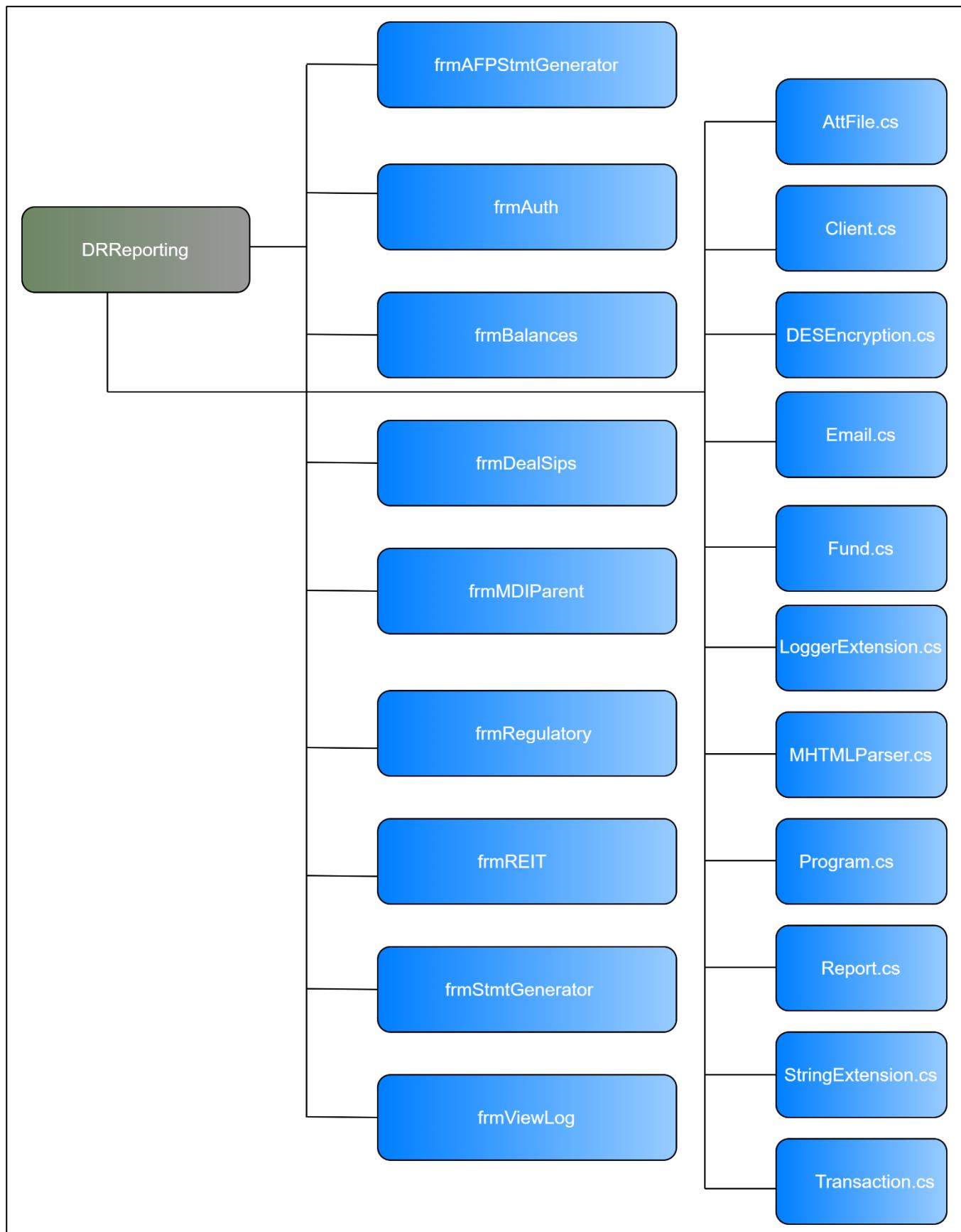


Figure 4: DRReporting - Solution and Project Hierarchical representation

3 Application Menu

Given below is the windows view of Parent Form (MDIParentForm) containing Reports menu in DR Reporting application.

The reports menu has below Sub menu-

- Statements
- Deal Slips
- Regulatory Reports
- Balances
- AFP Statements

3.1 Reports

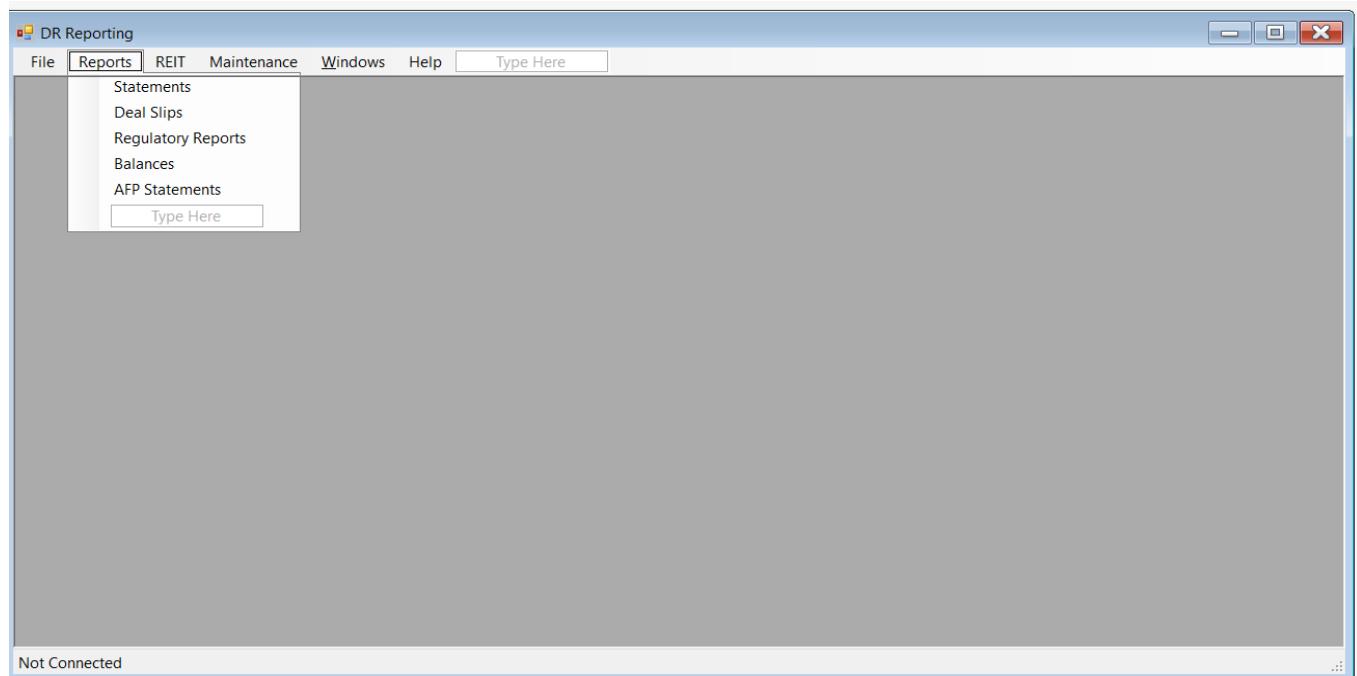


Figure 5: DRReporting - Reports menu

3.2 REIT (Real Estate Investment Trust)

The REIT menu has below Sub menu-

- REIT Upload Tool

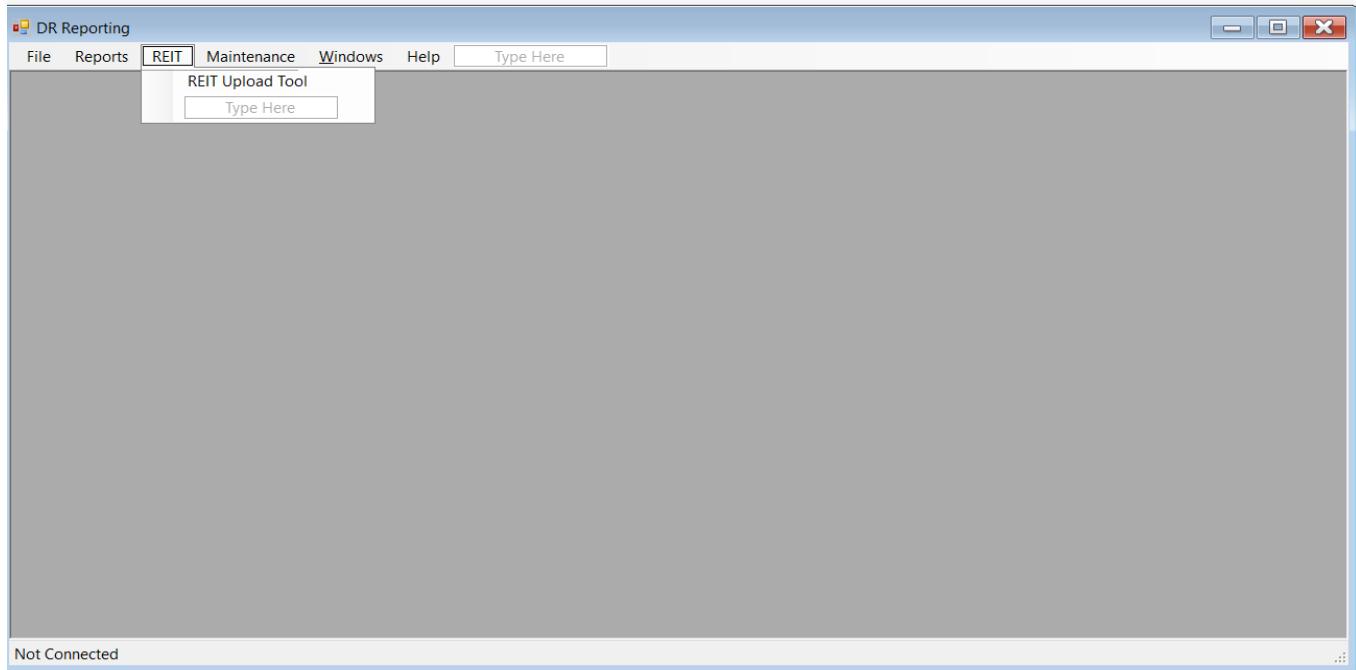


Figure 6: DRReporting - REIT menu

3.3 Maintenance

The Maintenance menu has below Sub menu-

- View Log

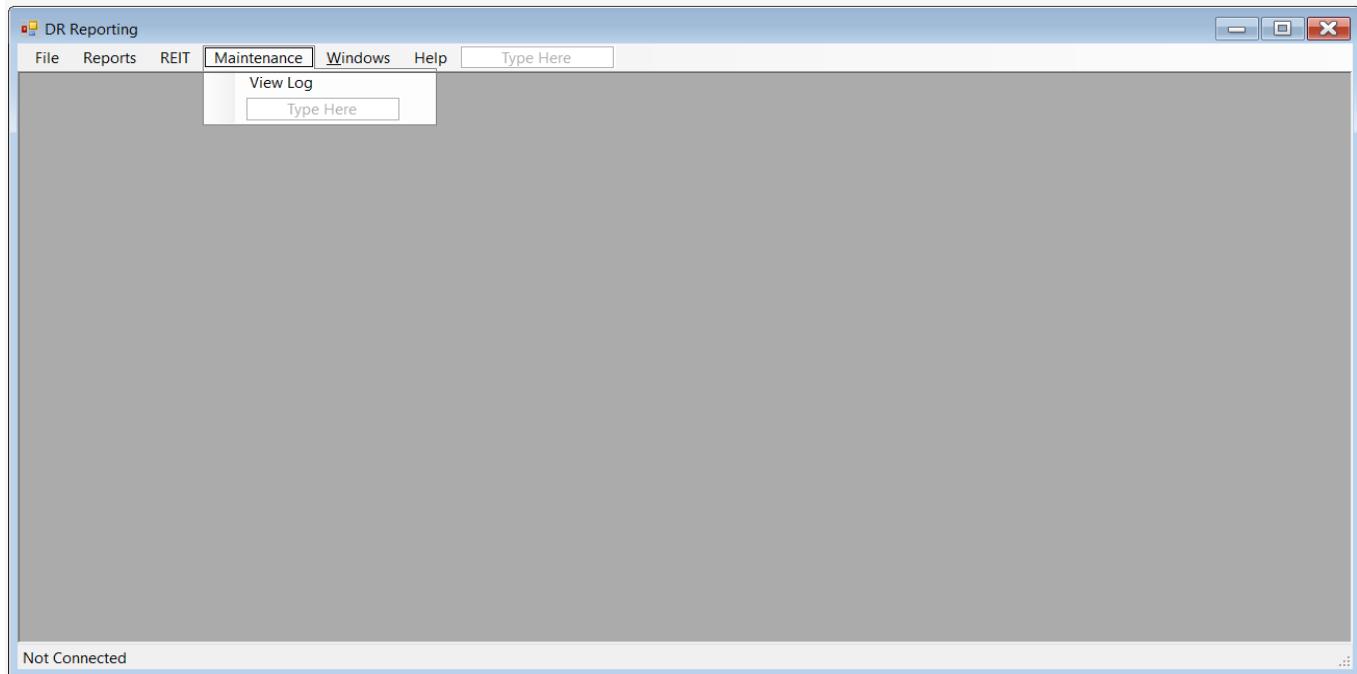


Figure 7: DRReporting - Maintenance menu

4 Technical Design Details

4.1 DR Reporting Application - Form Authentication

Code level details for Authentication(frmAuth)

The Authentication form opens when the application is accessed. Shown below is the login form that appears when an application is accessed for first time use, the user must provide authentication details like username and password. The credentials entered by the user are validated against the details (username/password) present in T24 Temenos database.

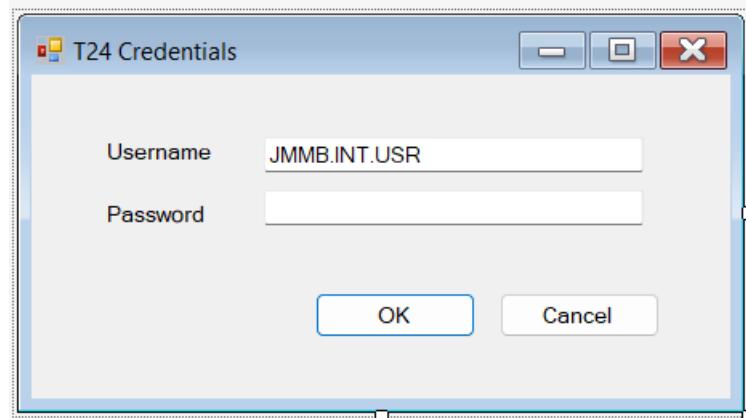


Figure 8: DRReporting - Login Window

The Authentication form(frmAuth) has one action method:

- OK Button Click

4.1.1 FrmAuth - Ok button click.

The login details entered by the user are sent to Temenos T24 and the response returned is evaluated. If authentication is successful, the user is navigated to frmMDIParent(Dashboard of DRReporting application).

Process Flow Diagram for FrmAuth 'OK Button click'-

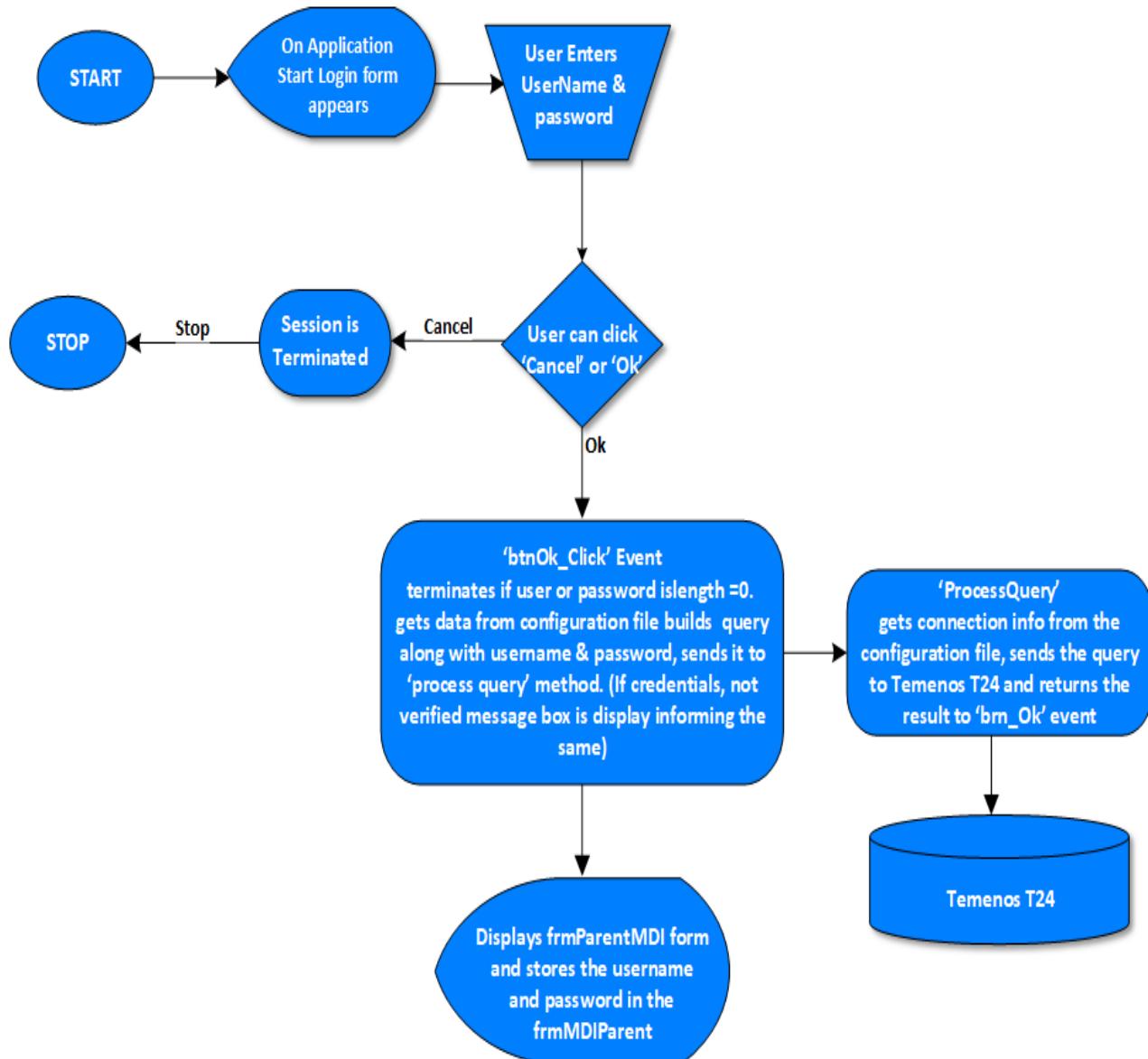
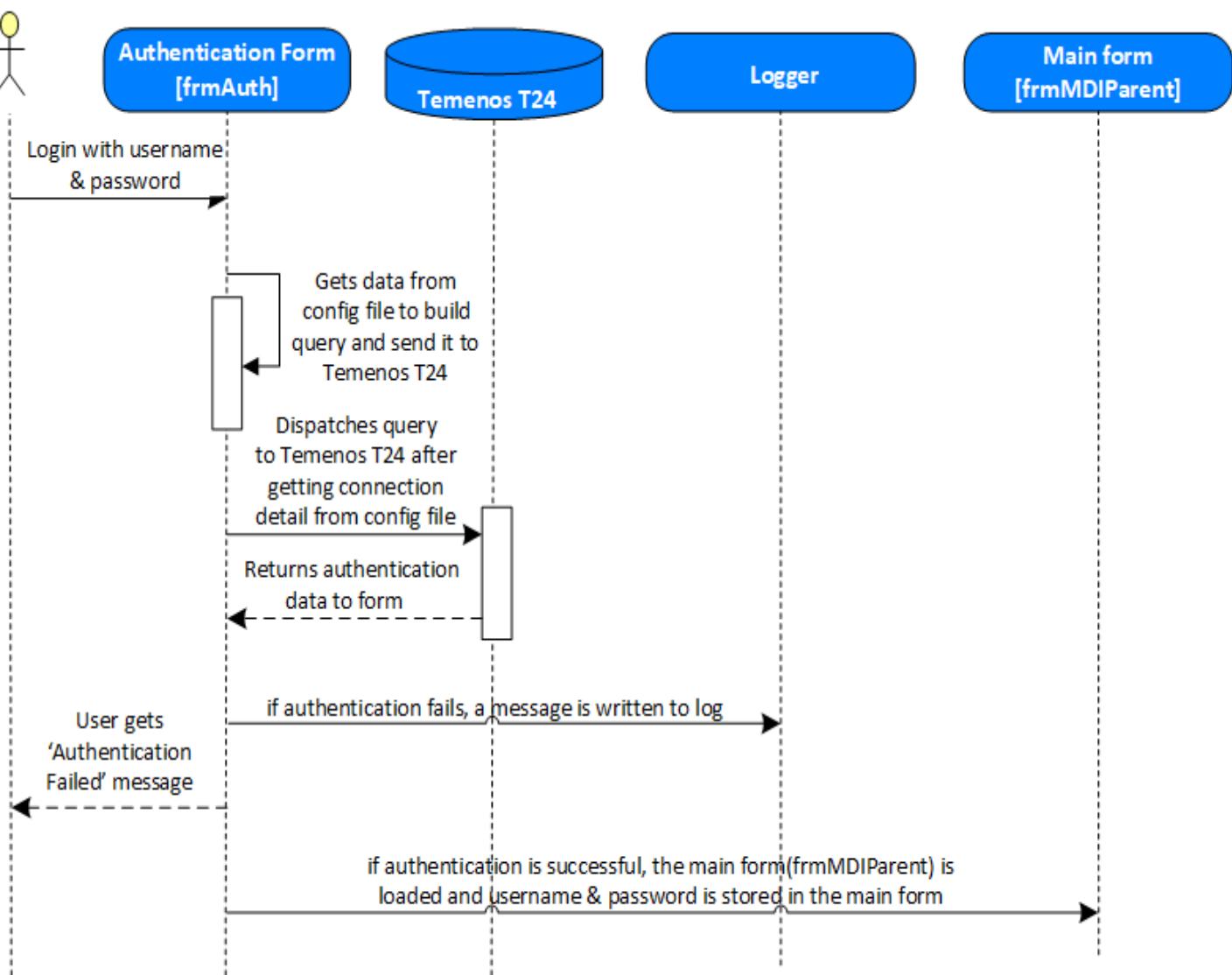


Figure 9: FrmAuth – 'Ok Button click' – Process flow diagram

Sequence Diagram for FrmAuth 'OK button click'-



Following is UML class diagram details of Authentication(frmAuth) form-

UML Class diagram for Authentication (frmAuth)-

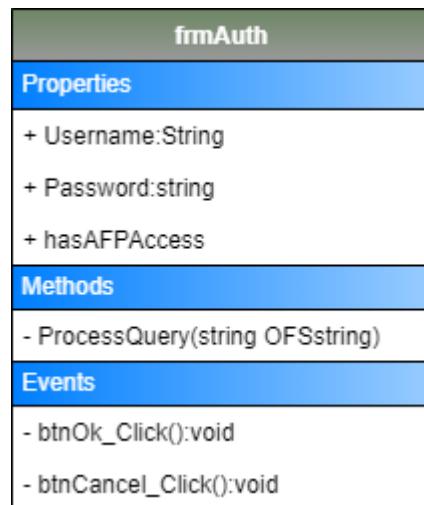


Figure 11: FrmAuth - UML Class diagram

ER Diagram for Authentication(frmAuth) Form-

One of this data structure is based on the company type

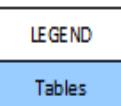
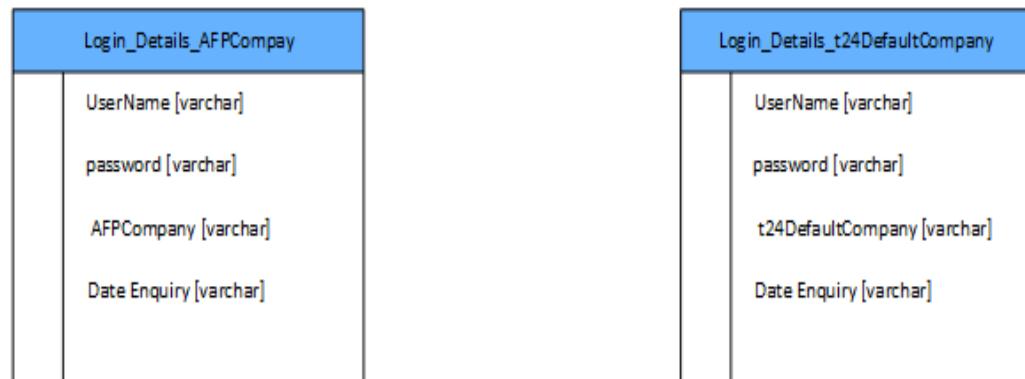


Figure 12: FrmAuth - ER Diagram

4.2 DR Reporting Application – MDI Form Parent Menu

Code level details for MDIParent Form

This is the main menu form, where users are directed to, upon successful login to the DR Reporting application depending on their permission access. In the MDIParent form Load event, the access list is retrieved from the SAFI_DB table named DRReportingAccess. Groups/Users permissions are added/removed from the Access Table data i.e 'Groups','Users', which determines the role based login (GrantAccess). The column data from this table is added to the AccessList, which is then passed to the CheckAndGrantAccess method.

Windows Security Groups in JMMB consists of **SDG_DR_T24Reports,SDG_BusinessSupport**. Within the CheckAndGrantAccess method, access is determined based on Active Directory information. For each entry in the AccessList, conditions such as 'Check Groups' and 'Users' are evaluated. If the conditions are met, access is granted using a SwitchCase structure.

The following AccessKey values and corresponding forms are given access:

AccessKey	Forms
BALANCE	Sets visibility ToolStripMenuItem
STATEMENT	Sets visibility to fundSystemGeneratorToolStripMenuItem
DealSlip	Sets visibility to dealSlipsToolStripMenuItem
REIT	Sets visibility to REITToolStripMenuItem
REGULATORY	Sets visibility to regulatoryReportsToolStripMenuItem
AFPStatement	Sets visibility to AFPStatementsToolStripMenuItem

Given below is the view of Parent (MDIParentForm) containing Reports menu in DR Reporting application

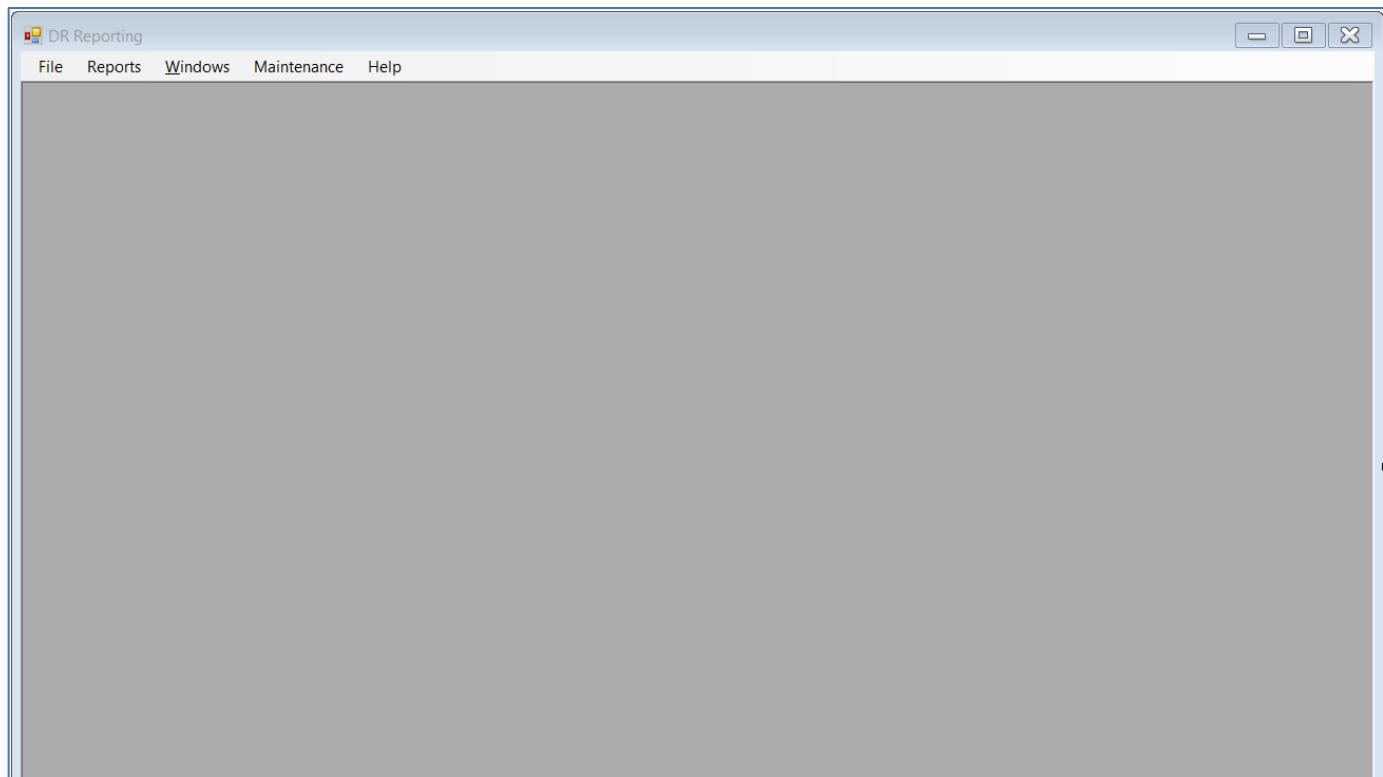


Figure 13: MDIParent Form - Application Screen view

The MDIParentForm contains Menu and Menu Items as Below:

Nr.	Menu	Sub-Menu
1	File	Exit
2	Reports	Statements DealSlips Regulatory Reports Balances AFP Statements
3	REIT	REIT Upload Tool
4	Windows	Cascade Tile Vertical Title Horizontal Close All
5	Arrange IconMaintainence	View Log
6	Help	About

Table 1: DRReporting - MDIParentForm (dashboard)

Process Flow Diagram for MDIParent form-

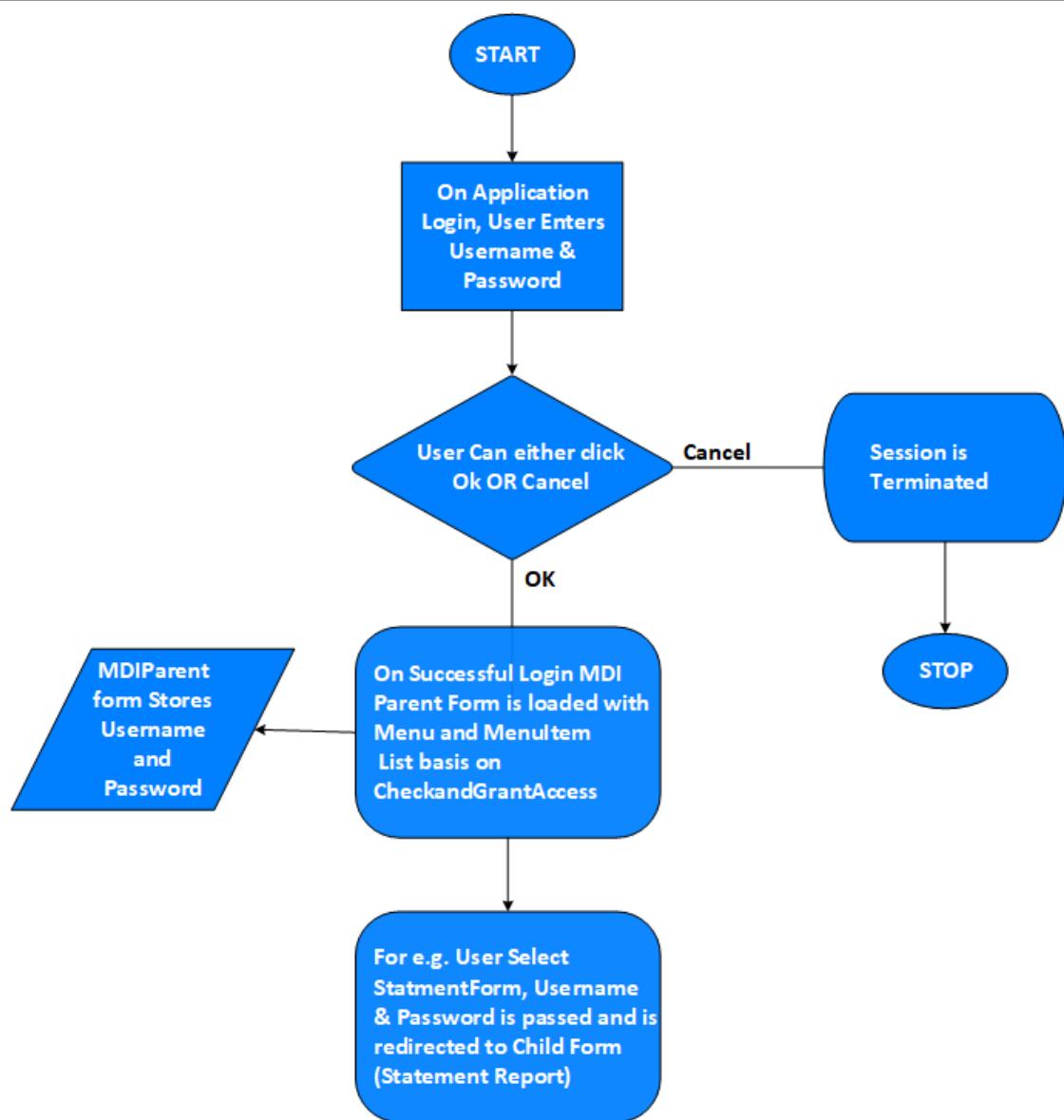


Figure 14: MDIParent (DashBoard) Process Flow Diagram

Sequence Diagram for MDIParent form-

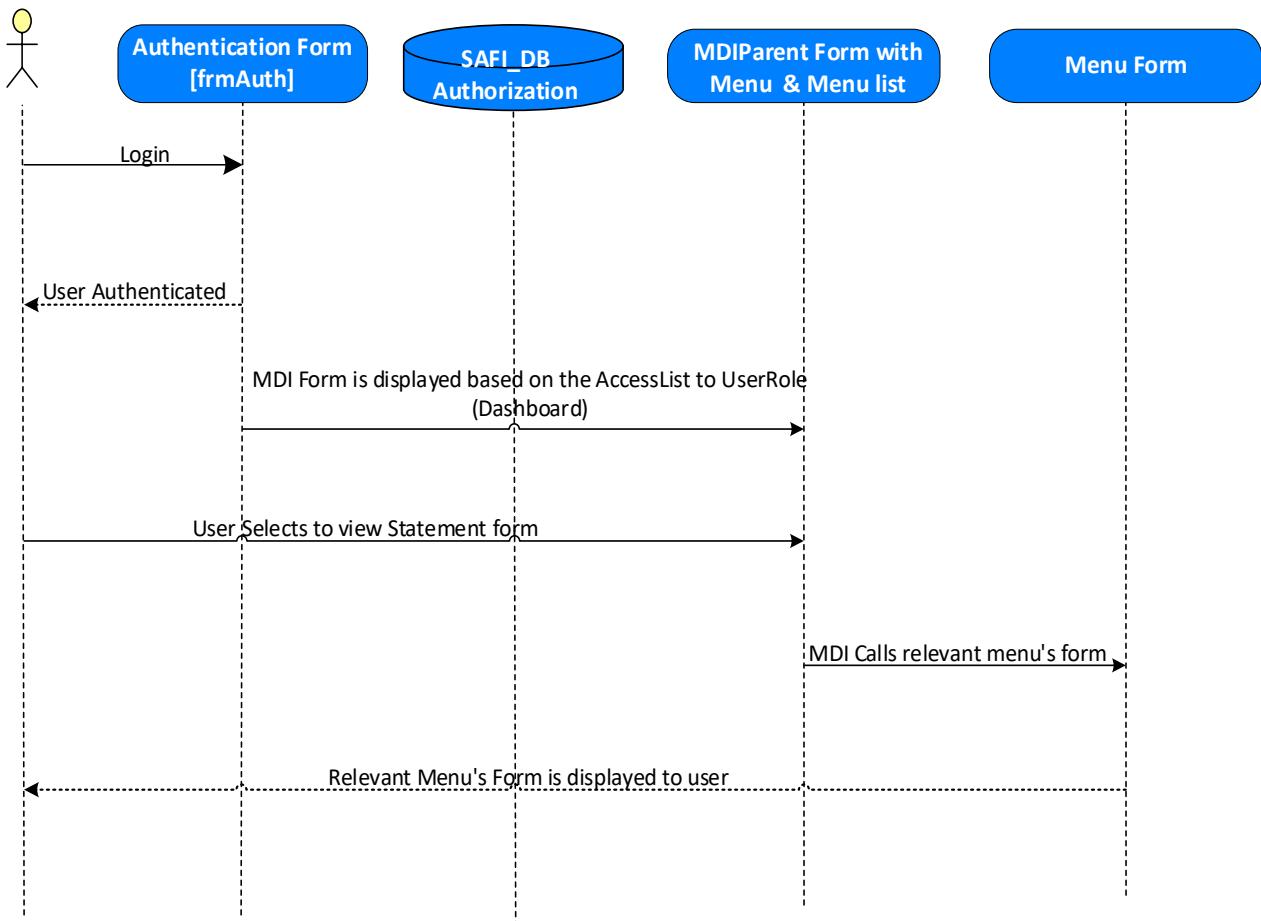


Figure 15: MDI Parent (DashBoard) Sequence Diagram

UML Class Level Diagram details for MDIParent Form-

frmMDIParent	
Fields	
- aFPStatementsToolStripMenuItem	
- arrangelconsToolStripMenuItem	
- balancesToolStripMenuItem	
- closeAllToolStripMenuItem	
- dealSlipsToolStripMenuItem	
- exitToolStripMenuItem	
- fileToolStripMenuItem	
- fundSystemGeneratorToolStripMenuItem	
- helpToolStripMenuItem	
- mainMenuStrip	
- maintenanceToolStripMenuItem	
- regulatoryReportsToolStripMenuItem	
- rEITUploadToolToolStripMenuItem	
- reportsToolStripMenuItem	
Properties	
+ AccessPermissions (string)	
+ username (string)	
+ password (string)	
+ Status (string)	
Methods	
- aboutToolStripMenuItem_Click():void	
- aFPStatementsToolStripMenuItem_Click():void	
- balancesToolStripMenuItem_Click():void	
- CascadeToolStripMenuItem_Click():void	
+ CheckAndGrantAccess(List<DRReportingAccess> accessList):void	
- dealSlipsToolStripMenuItem_Click():void	
- ExitToolsStripMenuItem_Click():void	
+ frmMDIParent()	
- frmMDIParent_Load():void	
- maintenanceToolStripMenuItem_Click():void	
- LoadNewChildForm():void	
- regulatoryReportsToolStripMenuItem_Click():void	
- rEITUploadToolToolStripMenuItem_Click():void	
+ SetAccess(string)	
- StatusBarToolStripMenuItem_Click():void	
- ToolBarToolStripMenuItem_Click():void	
- viewLogToolStripMenuItem_Click():void	

frmRegulatory_ClassDiagram	
' - '	Private field
' + '	Public field

Figure 16: MDIParent Form – UML Class diagram

ER Diagram for MDIParent form-

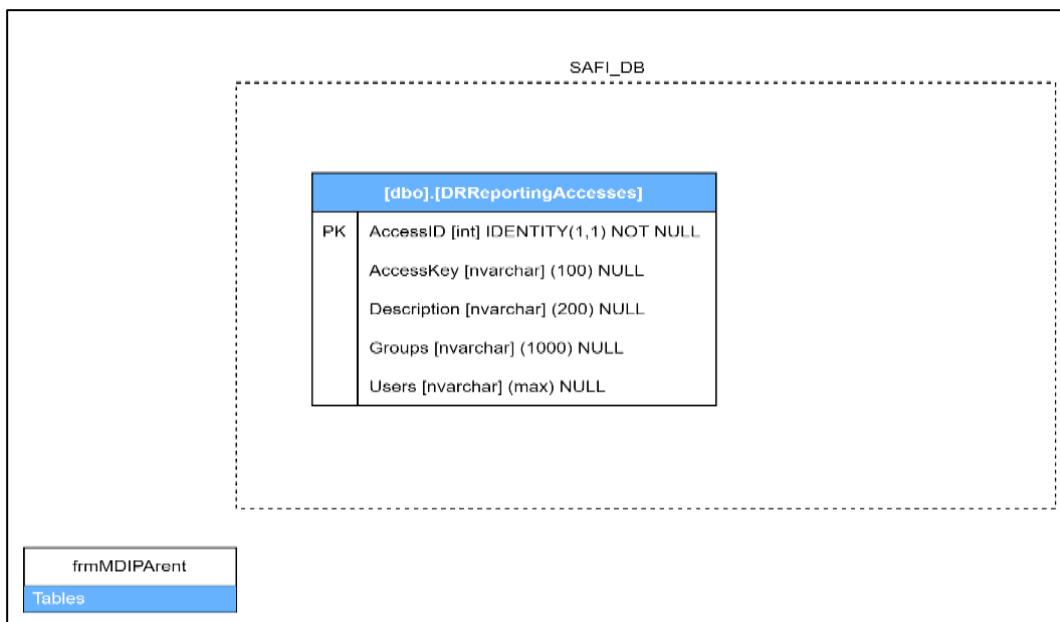


Figure 17: MDIParent Form (DashBoard) – ER Diagram

4.3 DR Reporting Application – Statements Report

Code level details for Statements report

The statements report can be generated by logging into DR application form MDIform, where user can select the required statement from the Report Menu. When user clicks on preview after selecting Fund, Client, start date and end date, config data required for report is fetched from database and the returned report is viewed in the application form.

Form frmStmtGenerator does the following:

- Allows building of reports after getting config data from database and fetching the reports from the Report server using config data and saves the reports to file system(disk).
- Once the reports are generated to the file system in the following path c:\DRStatements\{Selected fundCoCode}\Statements\{endDate}, the user validates the reports, then copies to the Email subfolder. The user may also add attachments (eg Quarterly Reports) in the OtherAttachments subfolder inside Email folder. Then the user clicks on the Email button to send an email to each client whose statements are in the email folder along with any attachments.

The statements generator form view is below:

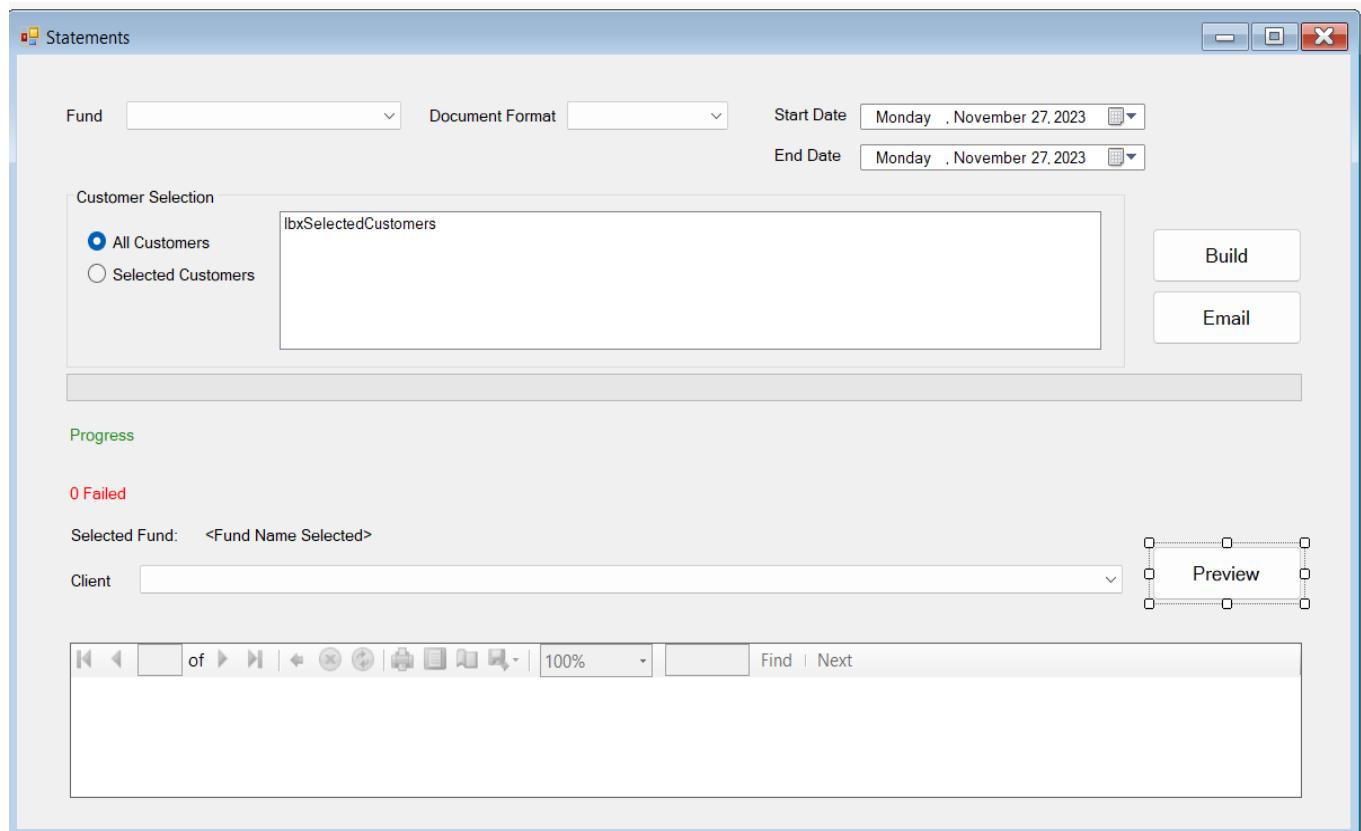


Figure 18: Statement Generation form – Application Screen

The Statements generation form has four action methods:

-
1. Form Load
 2. Build button click
 3. Email button click
 4. Preview button click

4.3.1 Statement Report – Form Load

During the form load process, the application creates the two ‘Linq2Sql dB’ contexts:

- SAFI_DB and
- T24ReportingDB

In the form load event of statement generation, a background process is initiated which fetches the client and funds data context entities and loads the data into client and funds combo-box respectively.

Process flow diagram for Statement Generation ‘Form load action’-

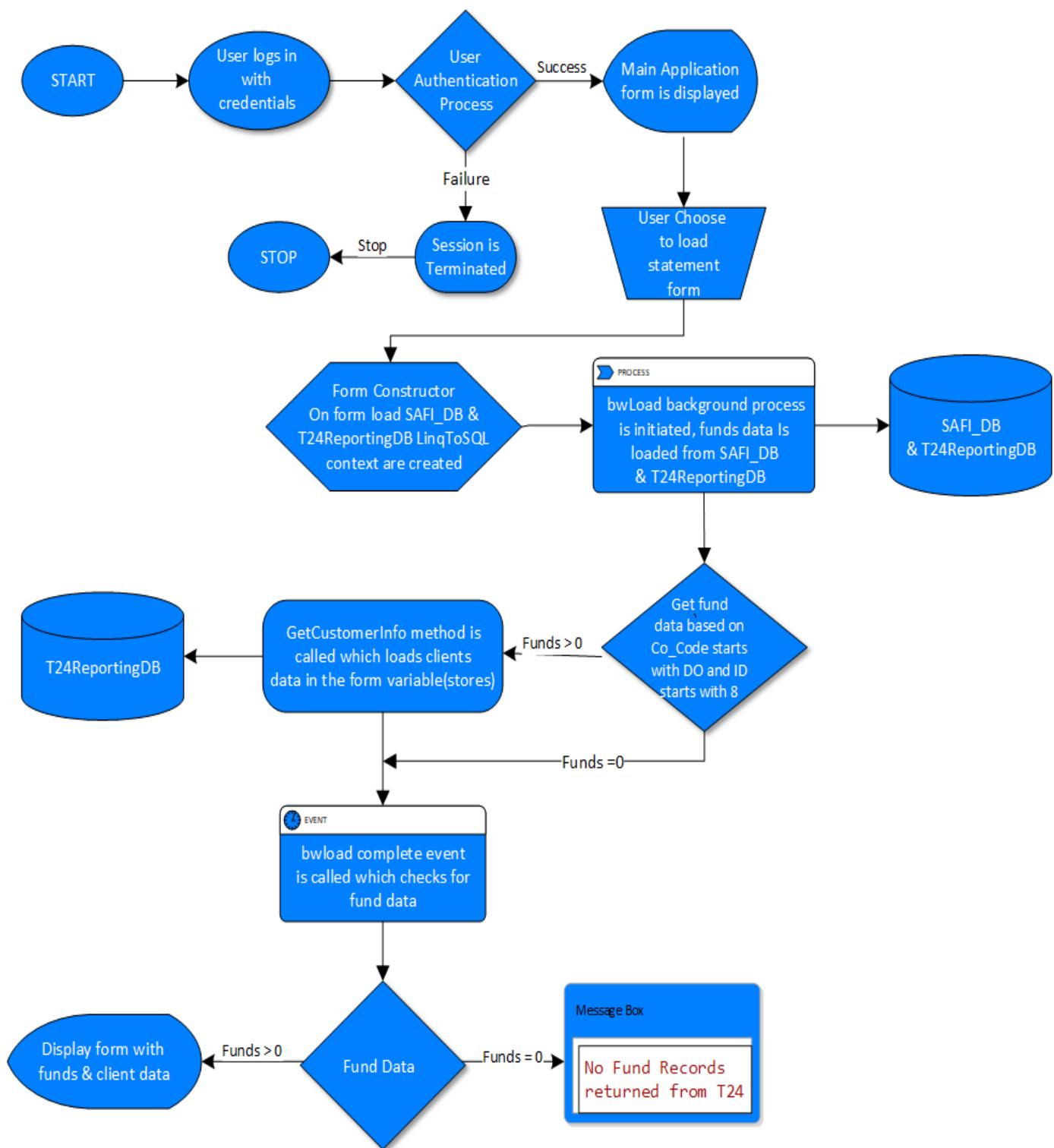


Figure 19: Statement Generation 'Form load action' – Process Flow Diagram

Sequence Diagram for Statement Generation 'Form load action' –

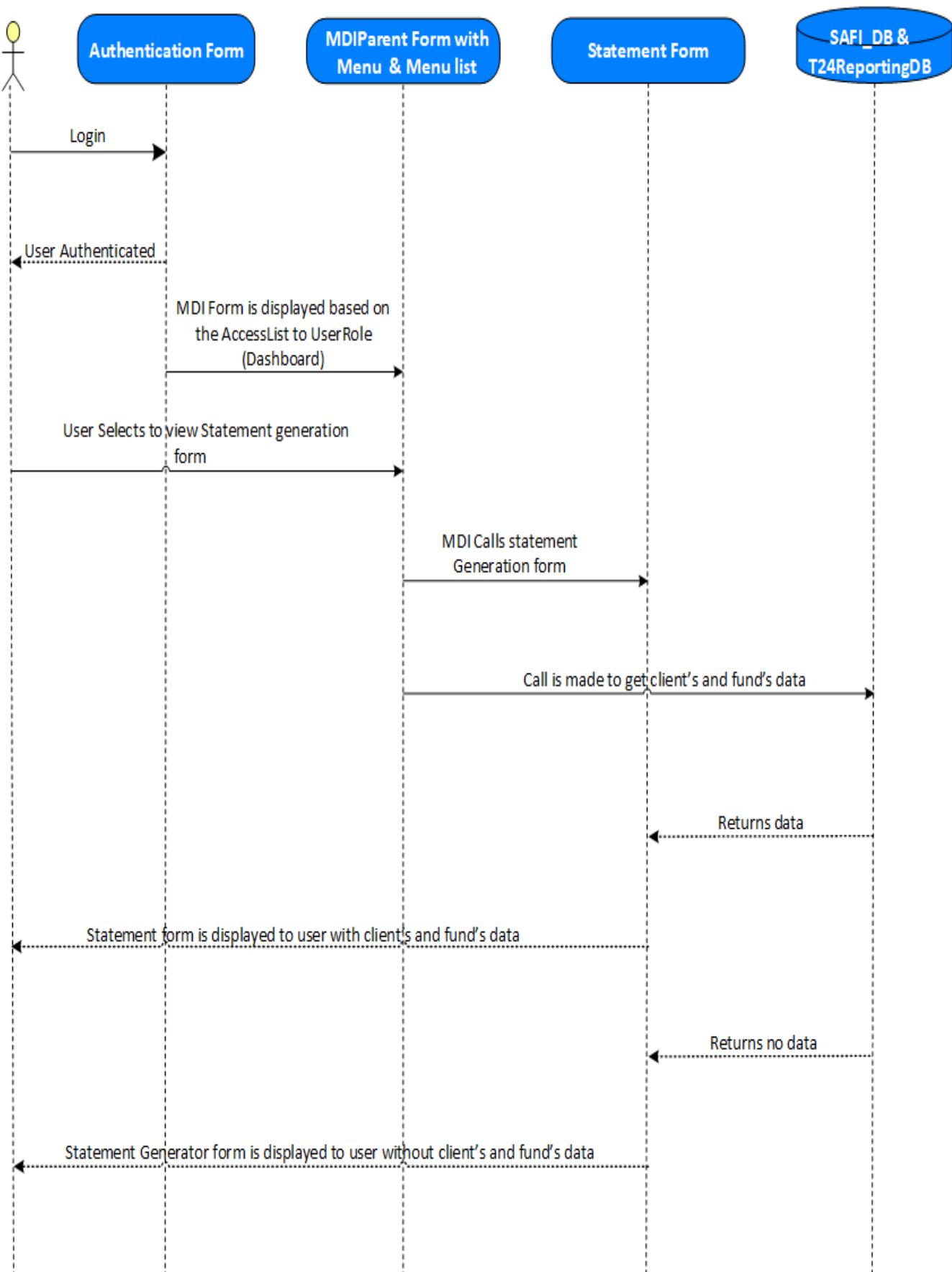


Figure 20: Statement Generation 'Form load action' – Sequence flow diagram

4.3.2 Statement Report – Build button click

To start the Statement Report, the end user must select a client and fund in the combo boxes, then set the start date and end date in the date fields. After this when the user clicks on 'Build', he is greeted with confirmation message box on selecting 'yes'. The creation of folders with different paths is done by checking if they exist. The data stored during form load is used to get URL for fetching data and the same is stored in the local path created earlier, in the .pdf format.

Once the background process involved in building the pdf reports is completed, it raises a complete event which displays the folder path of PDF files being generated.

Process flow diagram for Statement Report 'Build button click'-

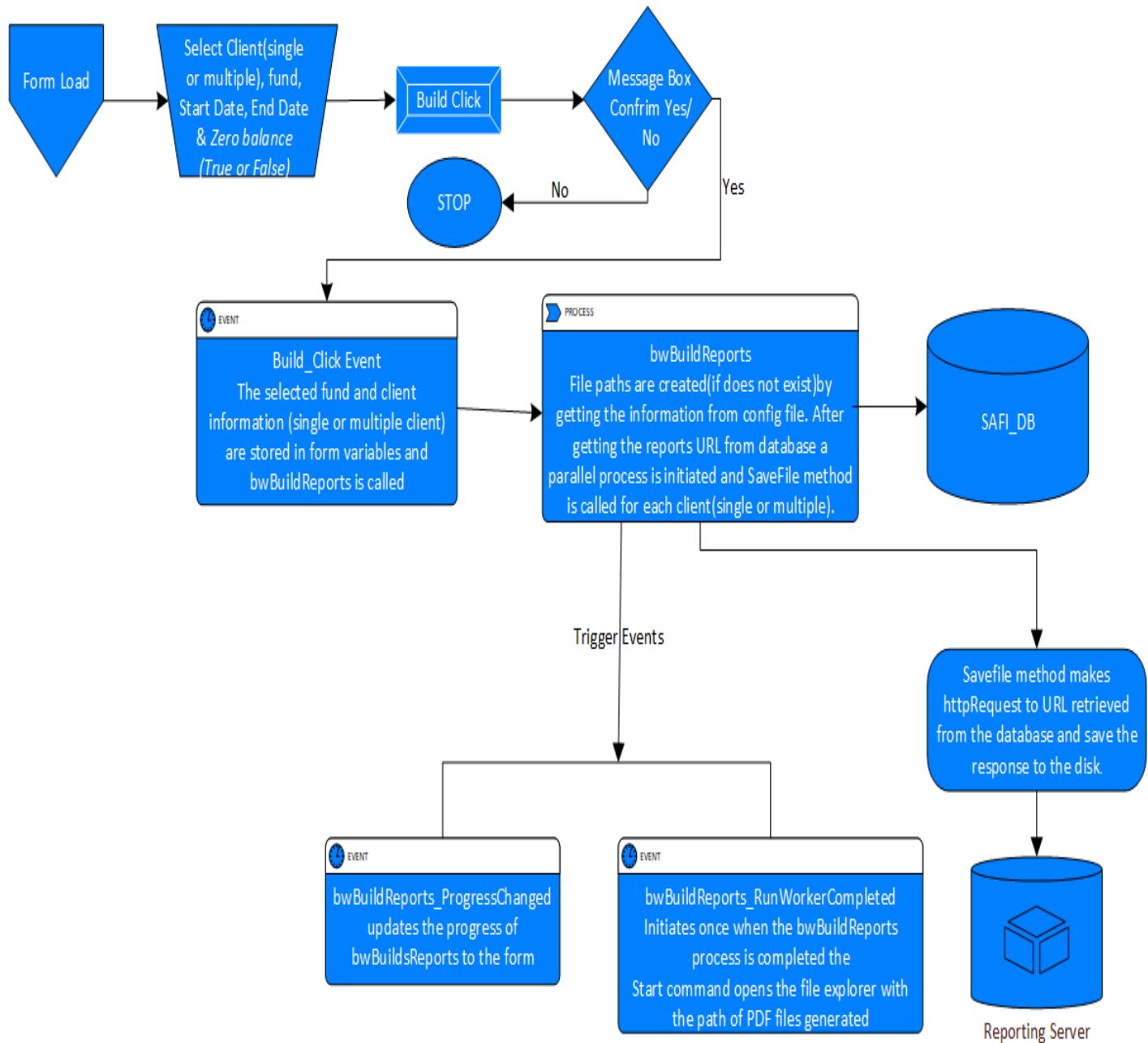


Figure 21: Statement Generation form 'Build Button Click' – Process Flow Diagram

Sequence diagram for Statement Report 'Build button click'-

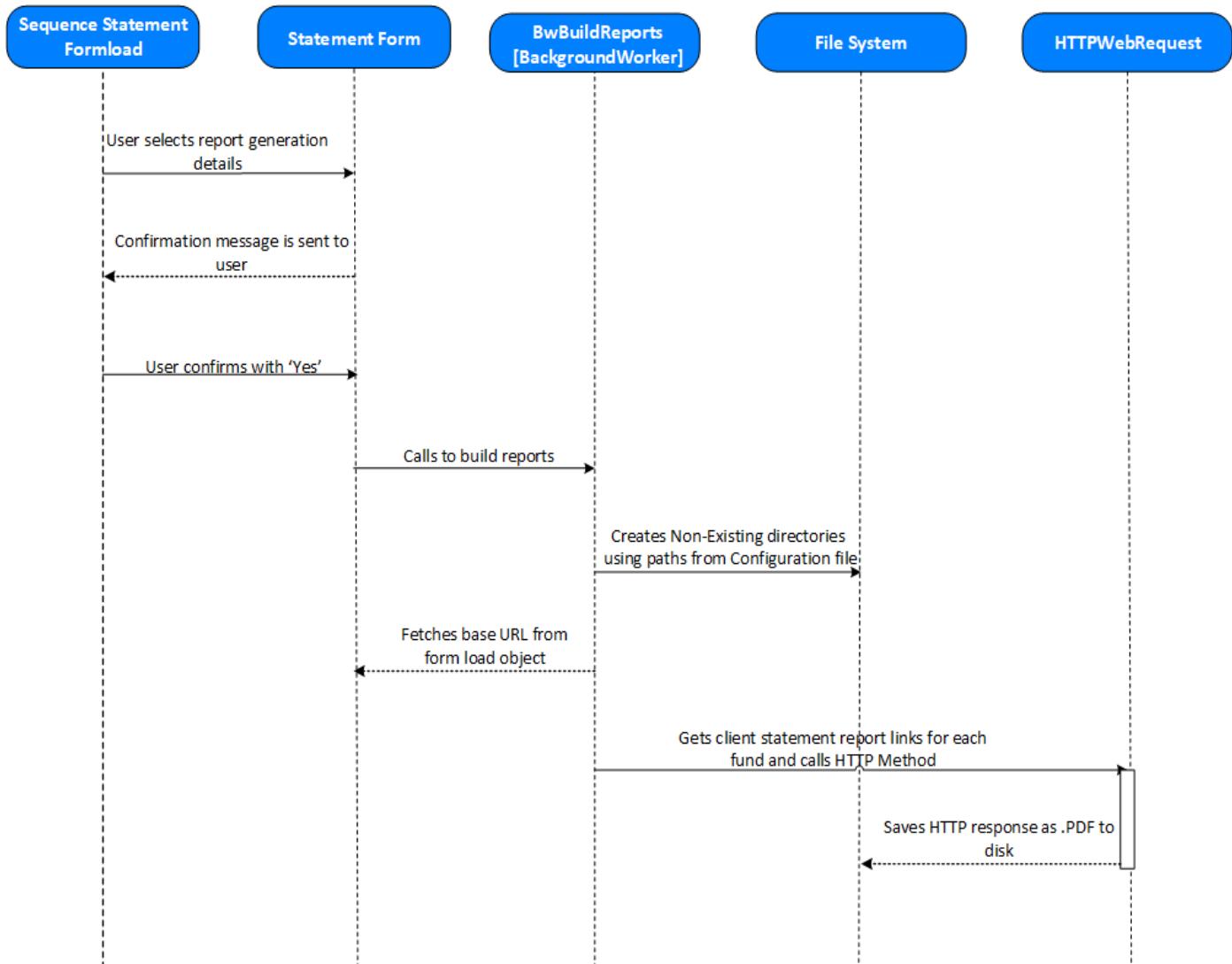


Figure 22: Statement Generation form 'Build Button Click' – Sequence Flow Diagram

4.3.3 Statement Report – Email button click

When the user clicks on email and gives confirmation, the email process checks for existence of folder defined in the configuration file, thereafter it checks for the reports (pdf files) generated during the build process. Then a background process is spawned.

An email template from the database is used to create the email that is sent, after the tags have been replaced and email service is called to send email to respective clients.

The following Tags are replaced with data from the application before email Is sent

Tags	Data from application
Replaces in the body & subject of the template.	
<Nombre>	Full Name
<Nombre>	If fullname=null then Cliente
<day>	endDate.ToString("dd")
<month>	months[endDate.ToString("MM")]
<year>	endDate.ToString("yyyy")
Replaces in the body of the template.	
<section1>	el Estado de Cuenta de su inversión
section2>	""

Table 3: Statement reports – Tags

Process Flow Diagram for Statement Report 'Email button click'

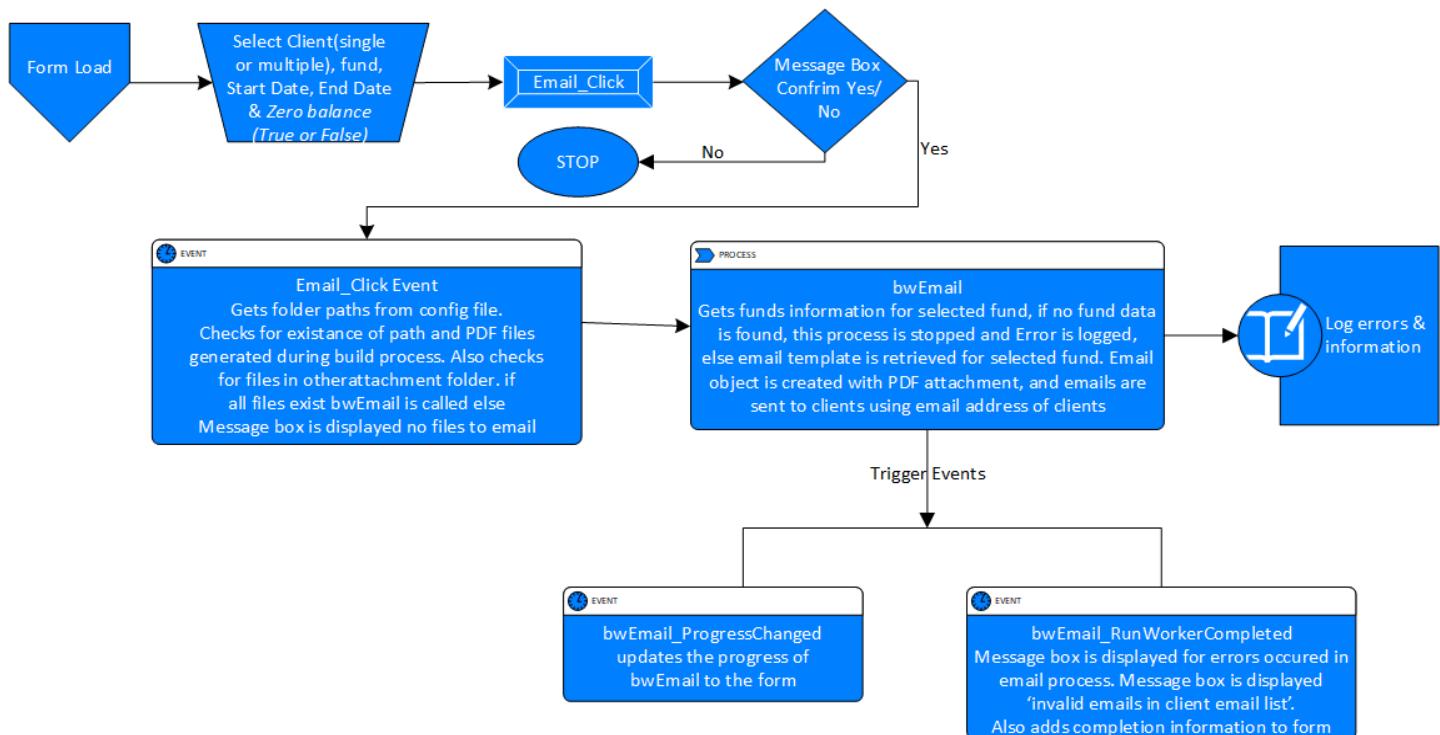


Figure 23: Statement Generation Form - 'Email Button click' - Process Flow diagram

Sequence Diagram for Statement Report 'Email button click'

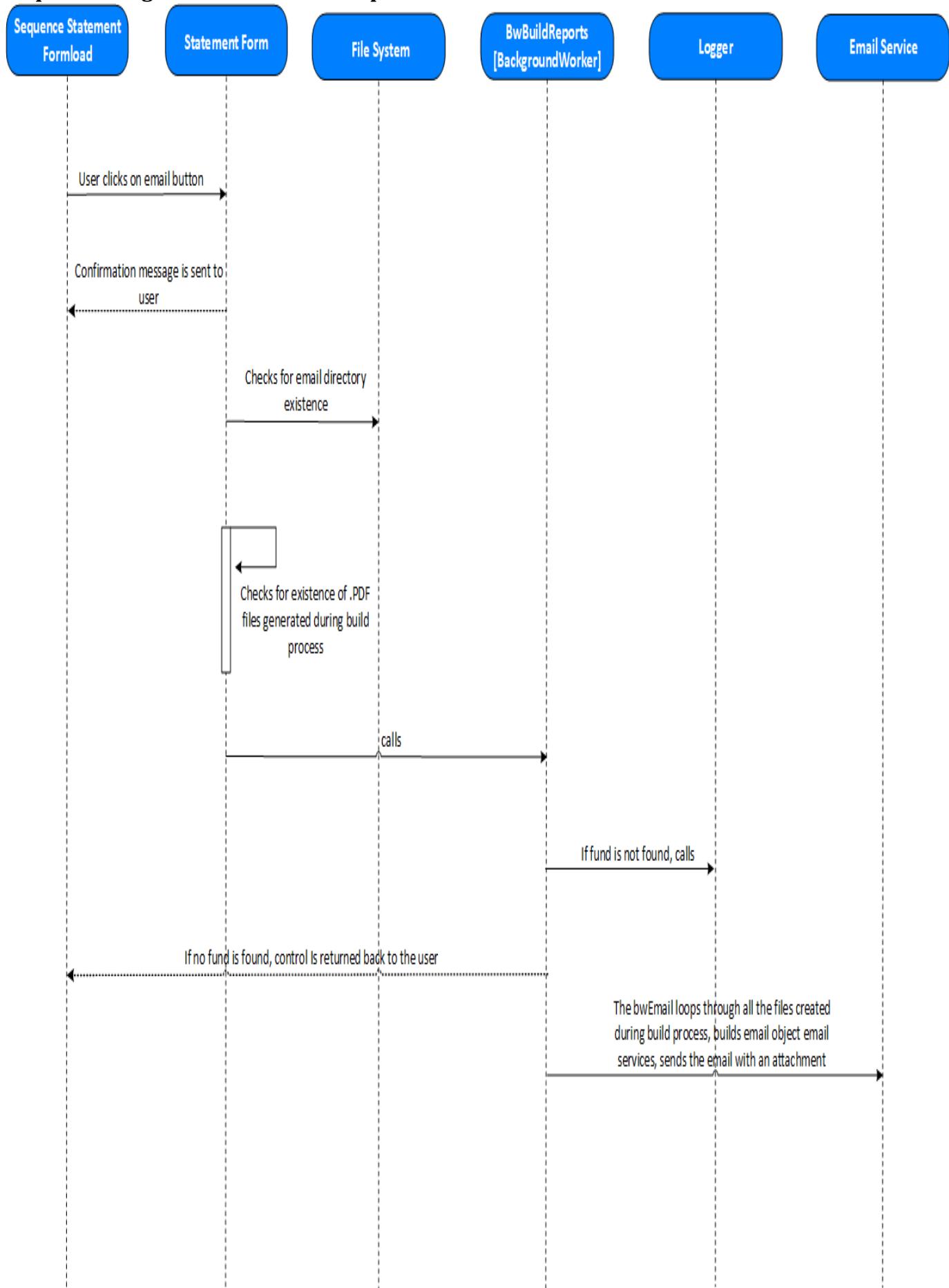


Figure 24: Statement Generation form 'Email Button Click' – Sequence Flow Diagram

4.3.4 Statement Report – Preview button click

The report previewing process involves the user selecting a client from the client dropdown and clicking on the preview button. The report data is loaded and is displayed to the user on report viewer.

Process Flow Diagram for Statement Report 'Preview button click'-

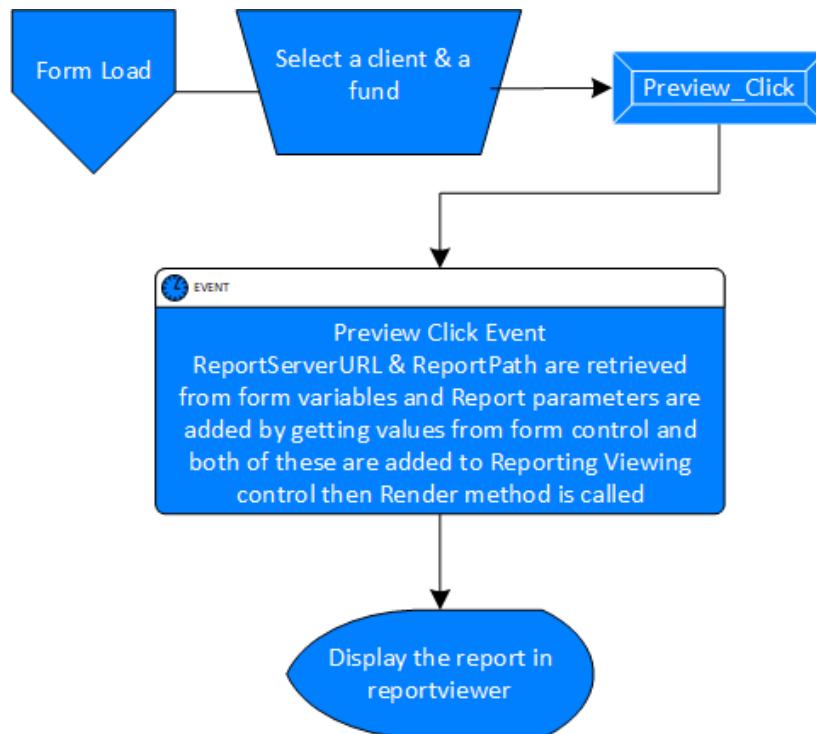


Figure 25: Statement Generation form 'Preview Button Click' – Process Flow Diagram

Sequence Diagram for Statement Report 'Preview button click'-

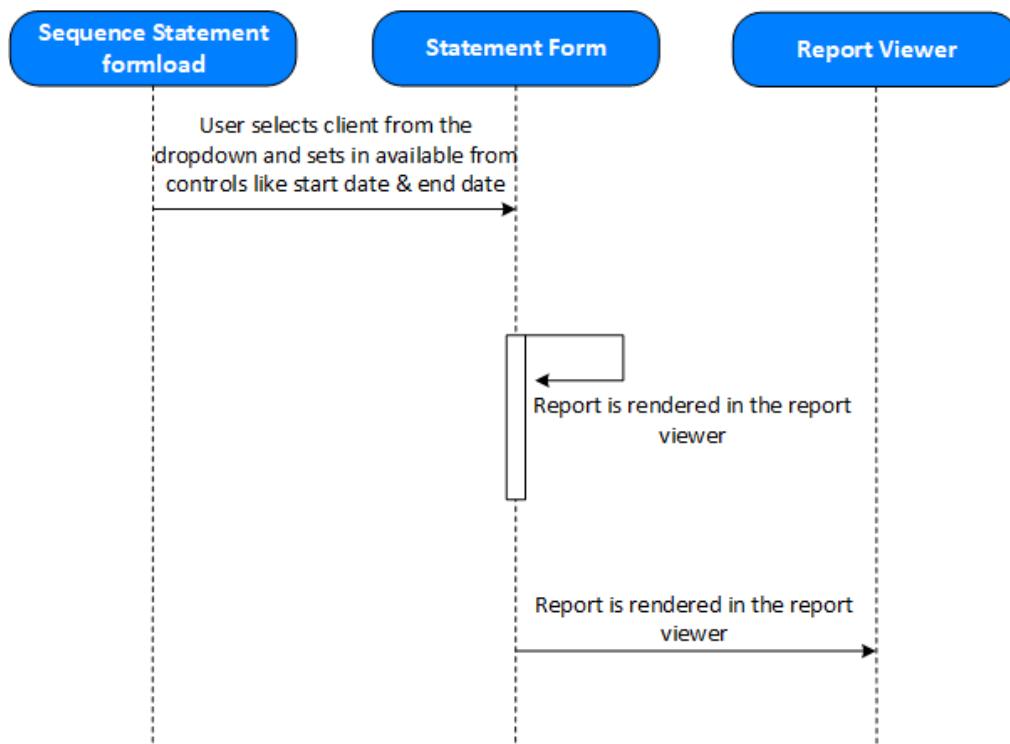


Figure 26: Statement Generation form 'Preview Button Click' – Sequence Flow Diagram

4.3.5 Client Statement Report – Build button click

The Client statement report (ClientStatement.rdl) passes input parameters and data is fetched via the SSRS report, which is then saved. All the data required for input parameters are loaded in form load event. The table 'tblFundSSRSUrl(SAFI_DB)' and column 'StatementSSRSUrl' provides the link to generate reports in the form of PDF and is saved to the disk. Once the parameters are selected from the form controls, the user clicks on 'Build' button. The URL is retrieved from table 'tblFundSSRSUrl' for the reports based on Co_Code of the selected fund. This URL's placeholder parameters, substitutes with values of user selected 'client/clients', then HTTP request to SSRS server for Client Statement report is sent and the response is saved as PDF report.

Process flow diagram for Client Statement report 'Build button click'-

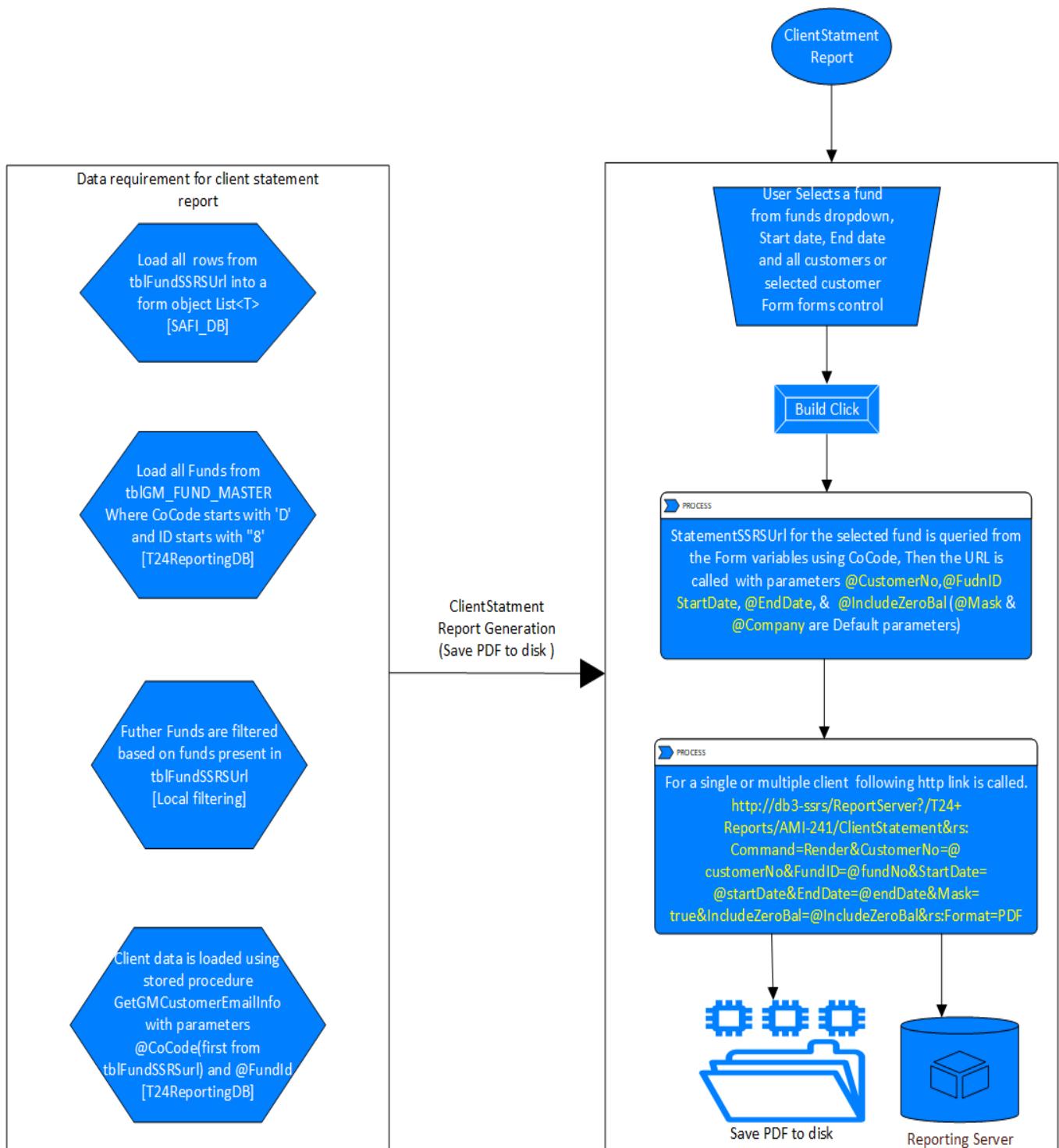


Figure 27: RDL Report File 'Build Button click'- Process Flow Diagram

4.3.6 Client Statement Report – Preview button click

The viewing of the Client statement report (ClientStatement.rdl) involves user selecting the Fund & Client in the form controls and then clicks 'preview'. The report will be visible in the report viewer control. This process is done by setting the path of the report by concatenating three columns from the table 'tblFundSSRSUrl' namely 'ReportServer', 'ReportPath' and 'StatementName'. The parameters required to render the report are attached to the report control, from the values in form controls. After this the report is fetched from SSRS reporting server and rendered in C# windows application report viewer control.

Process Flow Diagram for Client Statement report 'Preview button click'-

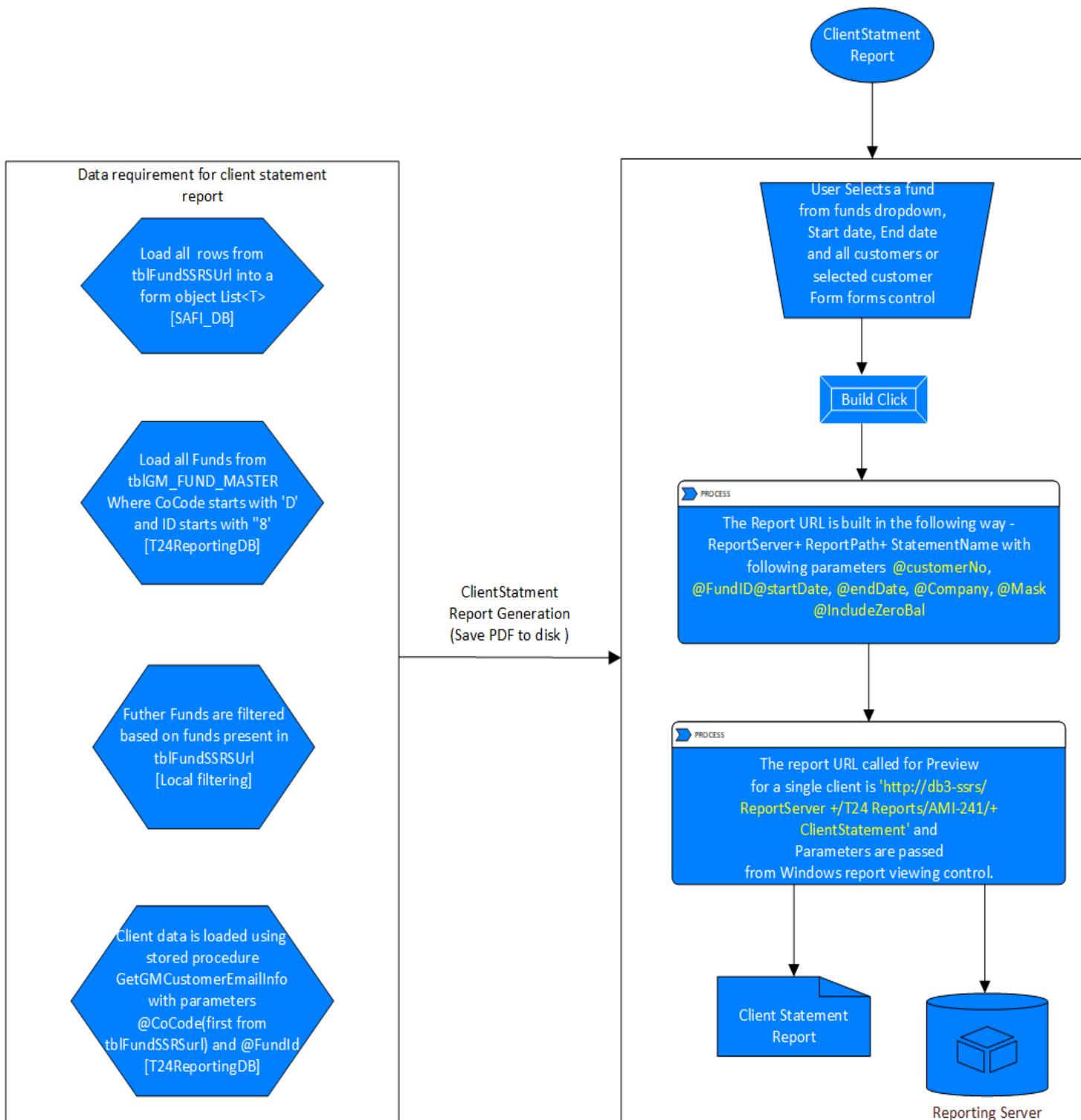


Figure 28: RDL Report File 'Preview button'- Process flow diagram

UML class level diagram details of the classes involved in statement generation-

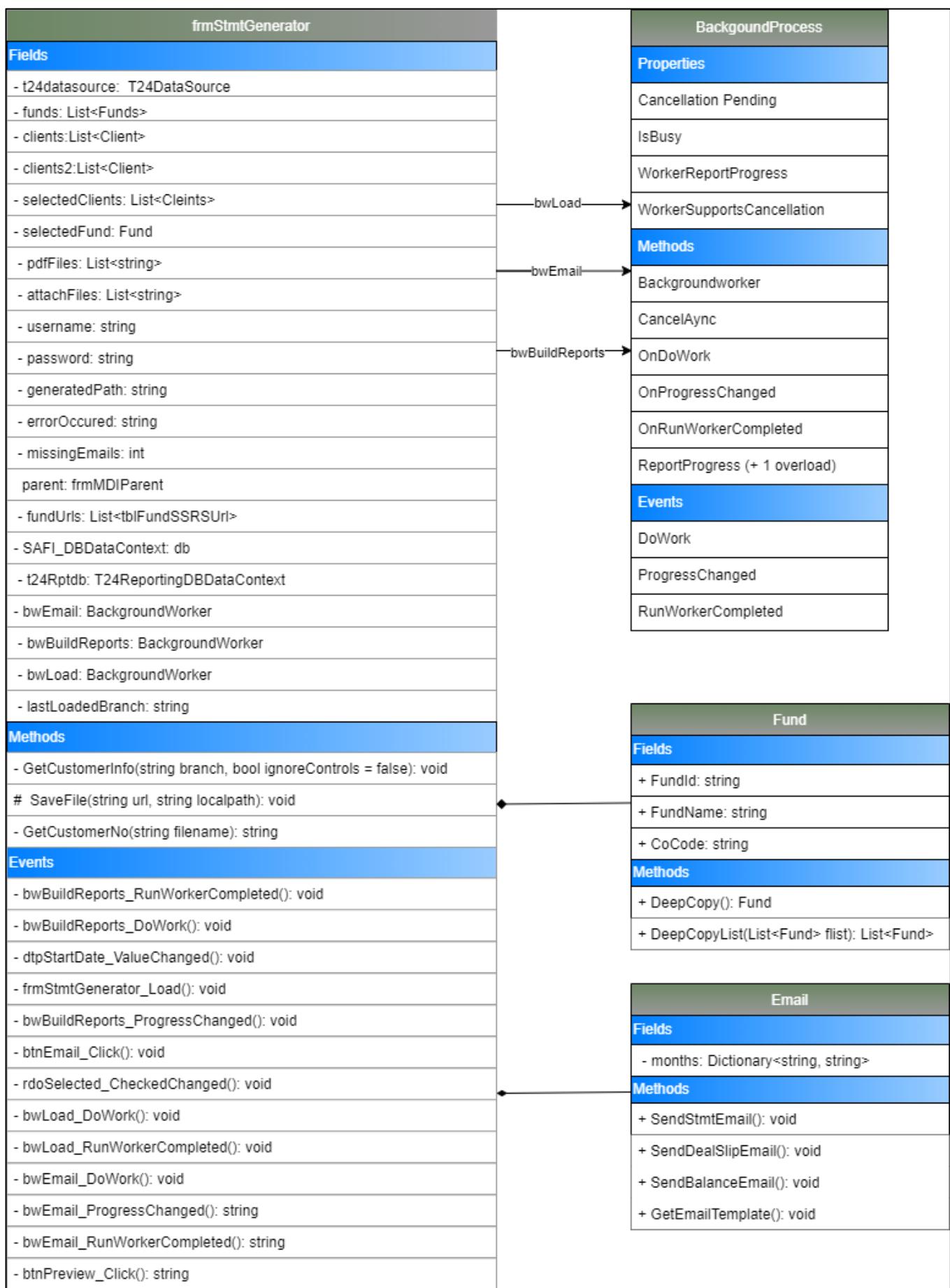


Figure 29: Statement Generation form 'Build Button Click' – UML Class Diagram

ER Diagram for statement generation flow¹



Figure 30: Statement Generation form ‘Build Button Click’ – ER Diagram

¹ columns omitted for brevity

4.4 DR Reporting Application – Deal Slips Report

Code level details for Deal Slips

When user clicks on preview after selecting Fund, Transaction and Time, config data required for report is fetched from database and the returned report is viewed in the application form.

Deal Slips Form does the following:

- Allows building of reports after getting config data from database and fetching the reports from the report server using config data and saving of the reports to file system(disk).
- Once the reports are generated to the file system in the following path
c:\DRStatements\Documents\FundSys\{Selected fundCoCode}\DealSlips\{endDate}, the user validates the reports, then copies to the Email subfolder. The user may also put additional attachments (eg. Quarterly Reports) in the OtherAttachments subfolder inside Email folder. The user then clicks on the Email button to send an email to each client whose statements is in the Email folder along with any attachments.

The Deal Slips form view is below-

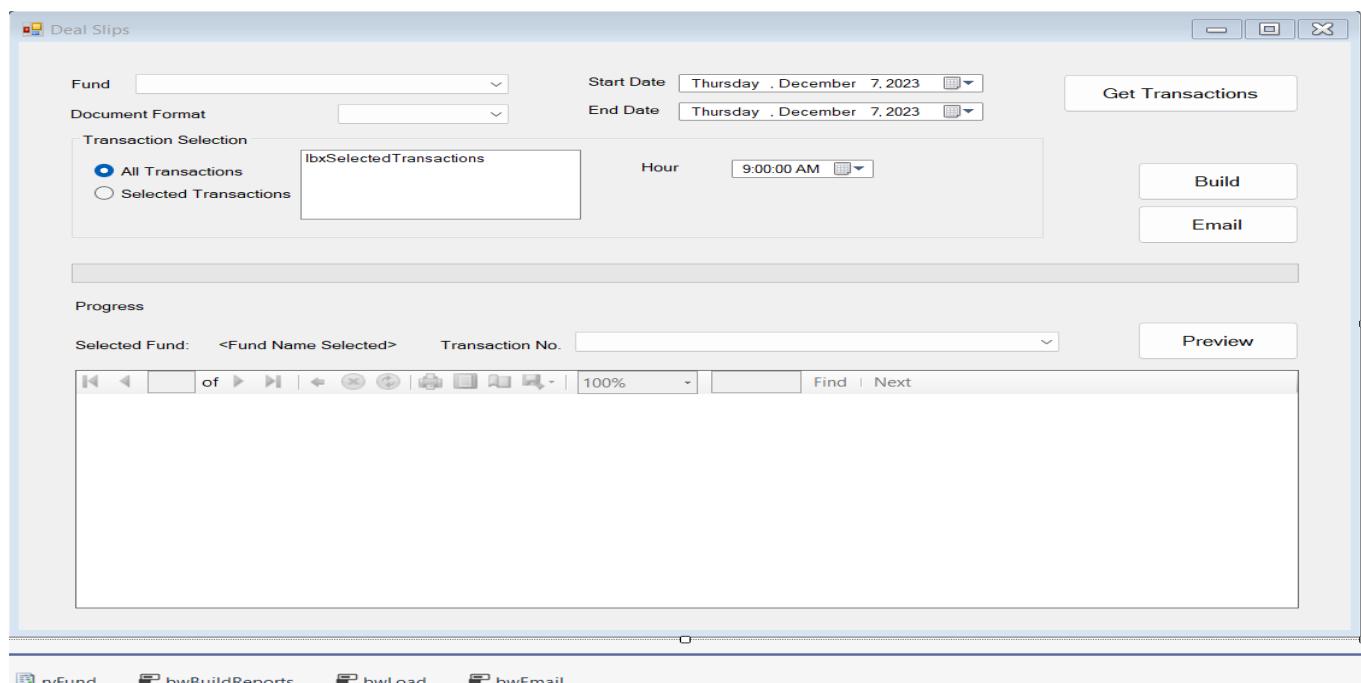


Figure 31: Deal Slips – Application View

The Deal Slips generation form has four action methods:

1. Form Load
2. Build button click.
3. Email button click.
4. Preview button click.
5. GetTransaction button click.

4.4.1 Deal Slips – Form Load

During the form load process, the application establishes two ‘Linq2Sql database’ contexts, one for SAFI_DB and another for T24DataSource. Subsequently, in the form load event of the ‘DealSlips’ form, a background process is triggered. This process is responsible for retrieving client information and default selected funds data from the mentioned database context entities.

The fetched data is then loaded into the client combo box, ensuring that users have access to relevant client and fund information. This approach streamlines the user experience by preloading essential data, facilitating smoother interactions within the ‘DealSlips’ form.

Process Flow Diagram for Deal Slips ‘Form Load action’-

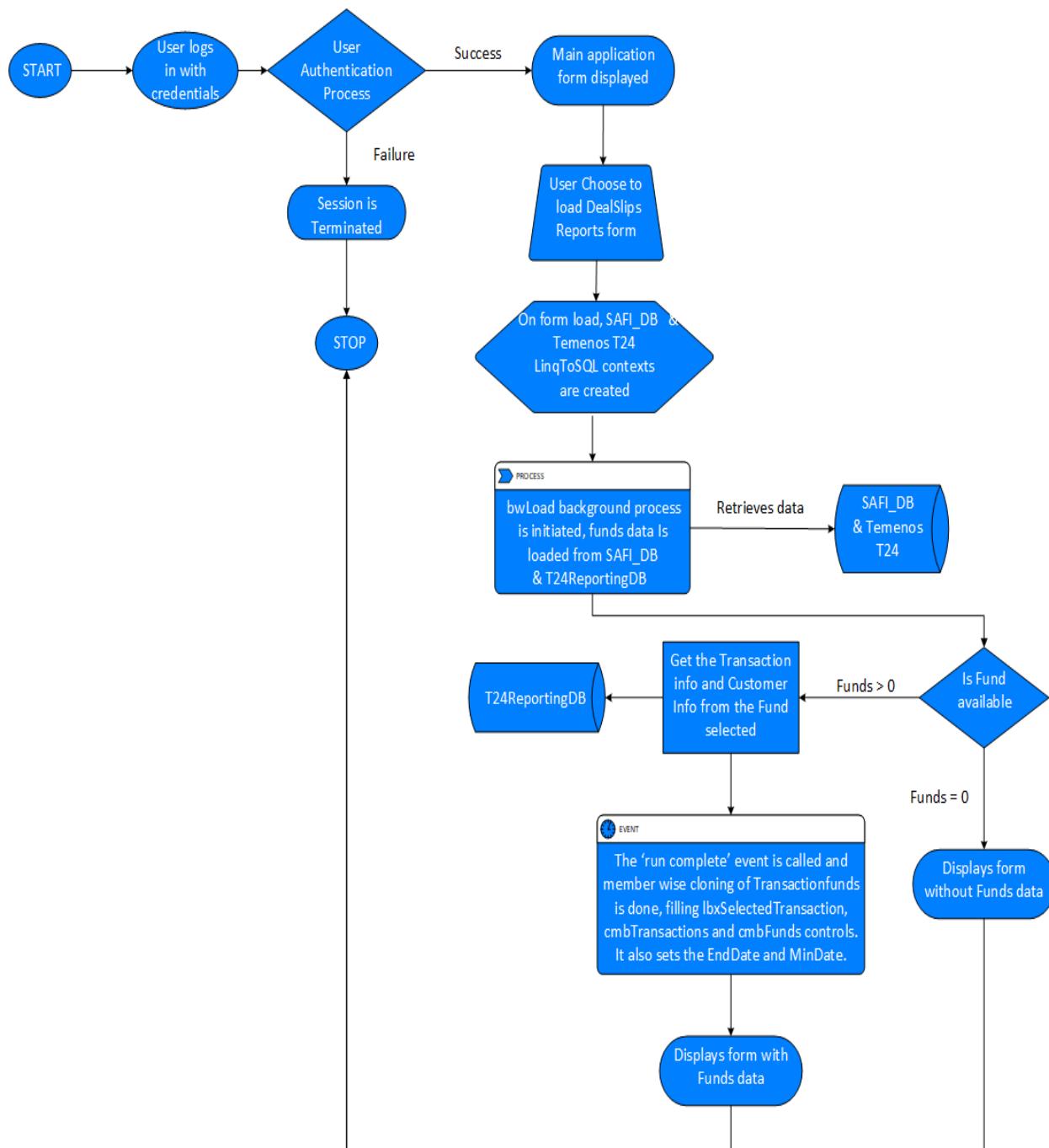


Figure 32: Deal Slips 'Form Load action' – Process Flow Diagram

Sequence Diagram for Deal Slips 'Form Load action'

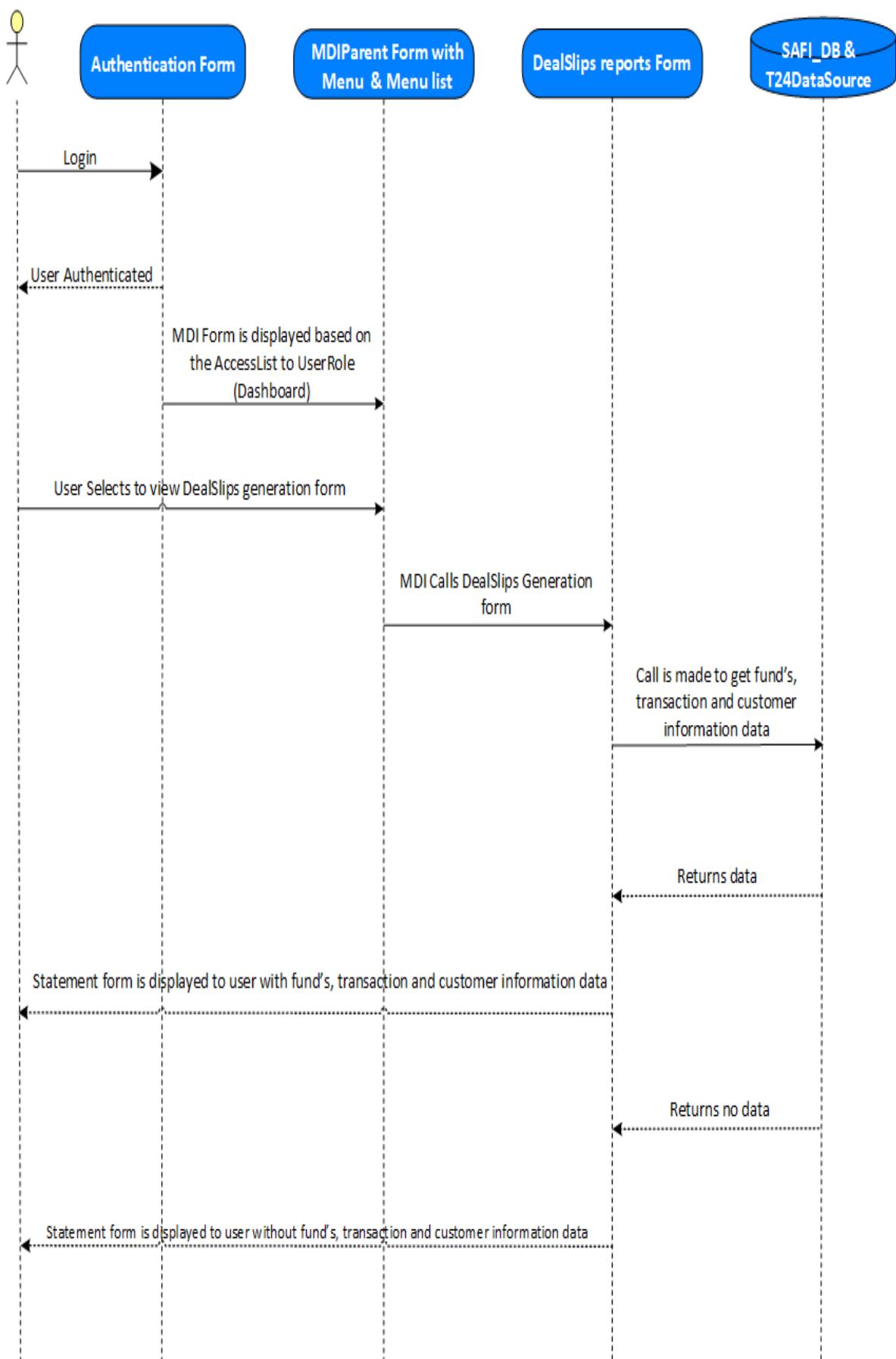


Figure 28: Deal Slips 'Form Load action' – Sequence Flow Diagram

4.4.2 Deal Slips – Build button click.

To initiate the construction process, the end user first selects a client and specifies start date and end date using the provided date fields. Upon clicking the 'Build' button, a confirmation message box appears. If the user selects 'Yes' the system proceeds to create a designated folder, checking its existence before proceeding. During the form's loading phase, data is stored to serve as a URL for fetching additional information. Simultaneously, this data is saved to the previously created local path. The data is formatted in PDF for further processing. Following the completion of background processes involved in generating the PDF reports, a 'complete' event is triggered. This event results in the display of the compiled PDF report based on the Format specified in the Combo box earlier during the load process, by default it is PDF.

Process Flow Diagram for Deal Slips 'Build Button Click'-

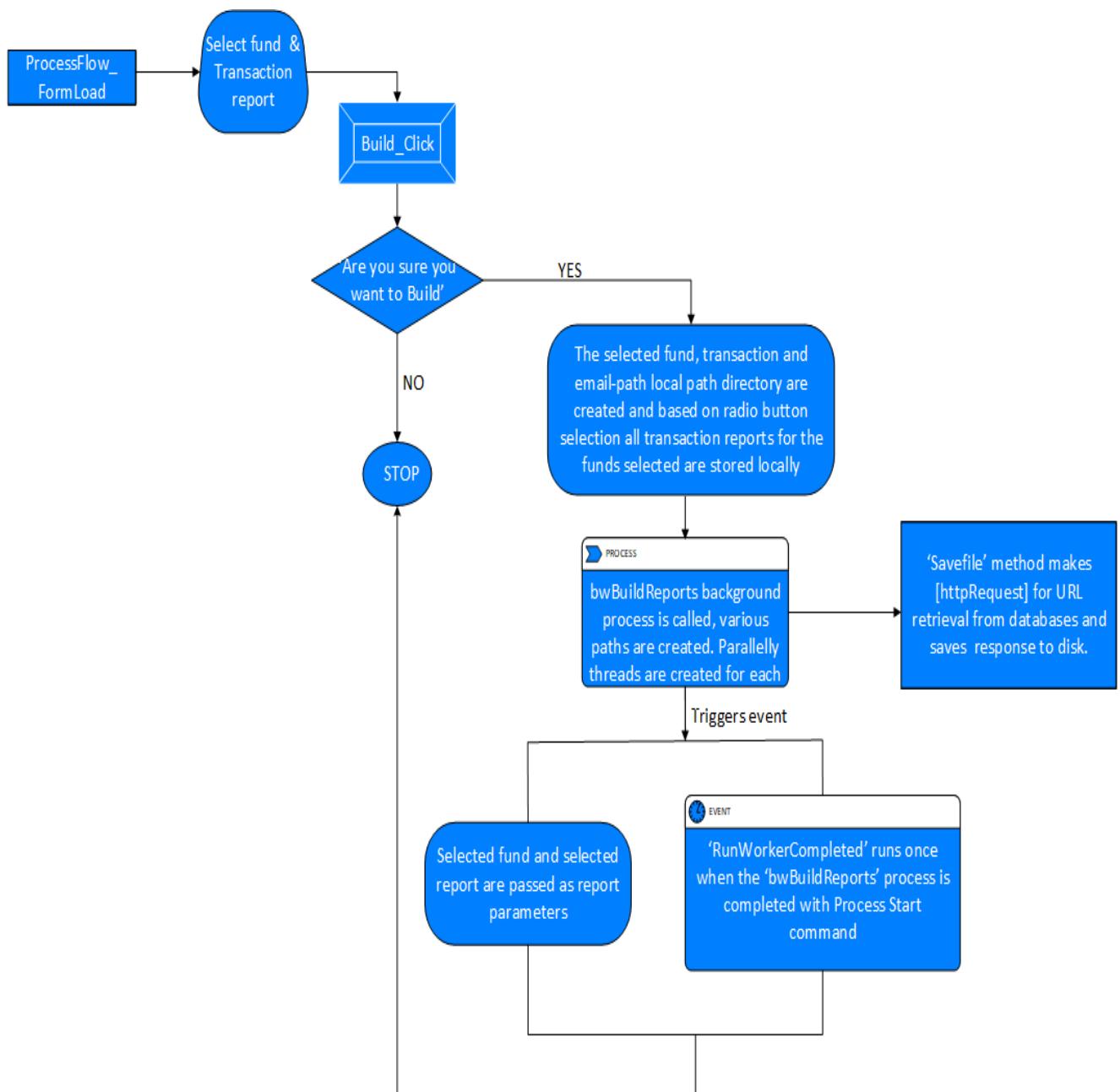


Figure 33: Deal Slips – Build button click – Process flow diagram

Sequence Diagram for Deal Slips 'Build Button Click'-

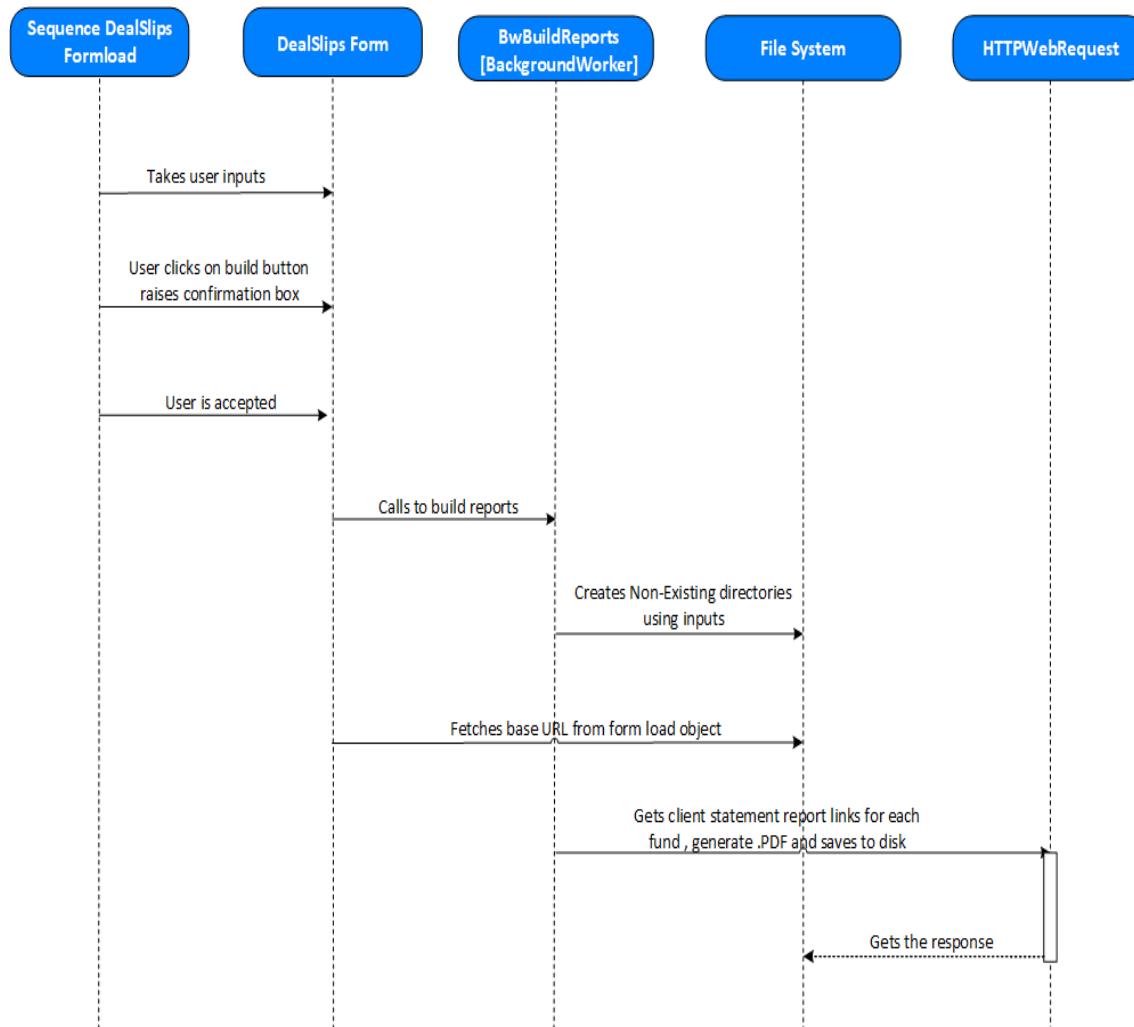


Figure 34: Deal Slips – ‘Build button click’ - Sequence Diagram

4.4.3 Deal Slips – Email button click.

When the user clicks on email and gives confirmation, the email process checks for existence of folder defined in the configuration file, thereafter it checks for the reports (pdf files) generated during the build process. Then a background process is spawned.

An email template from the database is used to create the email that is sent, after the tags have been replaced and email service is called to send email to respective clients.

The following tags are replaced from the application before email is sent

Tags	Data from Application
<customerName>	customerName
<dealSlipType>	tranType
<transactionDate>	endDate.ToString("yyyy-MM-dd")
<fundNo>	fundInfo.CoCode
<Subject>	if (tranType == "SUS") Comprobante de suscripción de cuotas "+ endDate.ToString("dd/MM/yyyy") else "Comprobante de suscripción de cuotas " + endDate.ToString("dd/MM/yyyy")
<DRSignatureTag>	fundInfo.DealSlipSignature

Process Flow Diagram for Deal Slips 'Email Button click'-

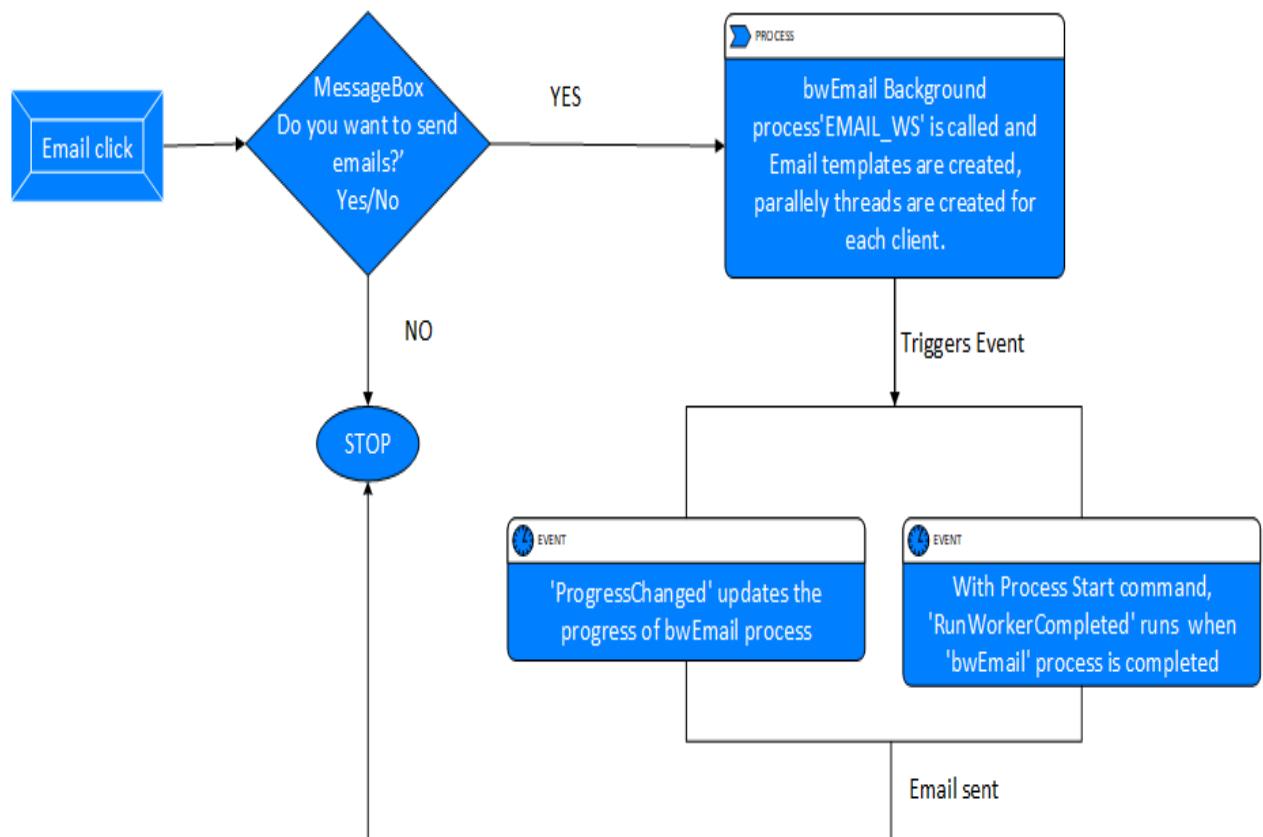


Figure 35: Deal Slips – 'Email button click' - Process Flow Diagram

Sequence Diagram for Deal Slips 'Email Button click'

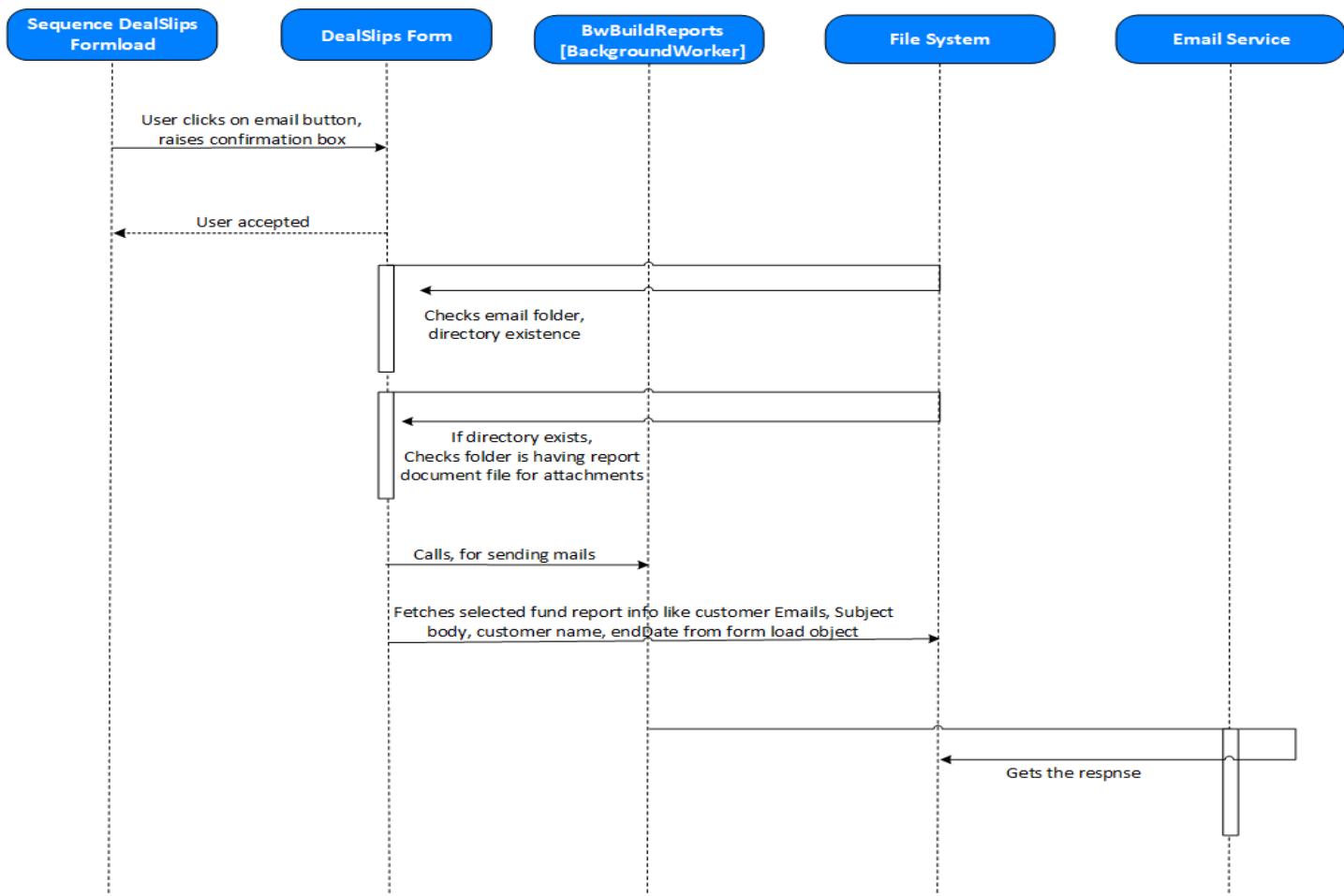


Figure 36: Deal Slips 'Email button click' - Sequence Diagram

4.4.4 Deal Slips – Preview button click.

Upon clicking the 'Preview' button, an authorized user can review the generated Deal Slips file. The user has the option to select the fund from a drop-down list and the selected Transaction to preview the file. This functionality allows users to inspect the contents of the file before proceeding with the email sending process, providing an opportunity to address any doubts or make necessary changes.

It's important to note that the preview functionality is limited to one selected Transaction number for the Fund with Time as input, ensuring a focused and controlled viewing experience. This restriction helps to maintain clarity and precision during the preview process.

Process flow Diagram for Deal Slips 'Preview button click'-

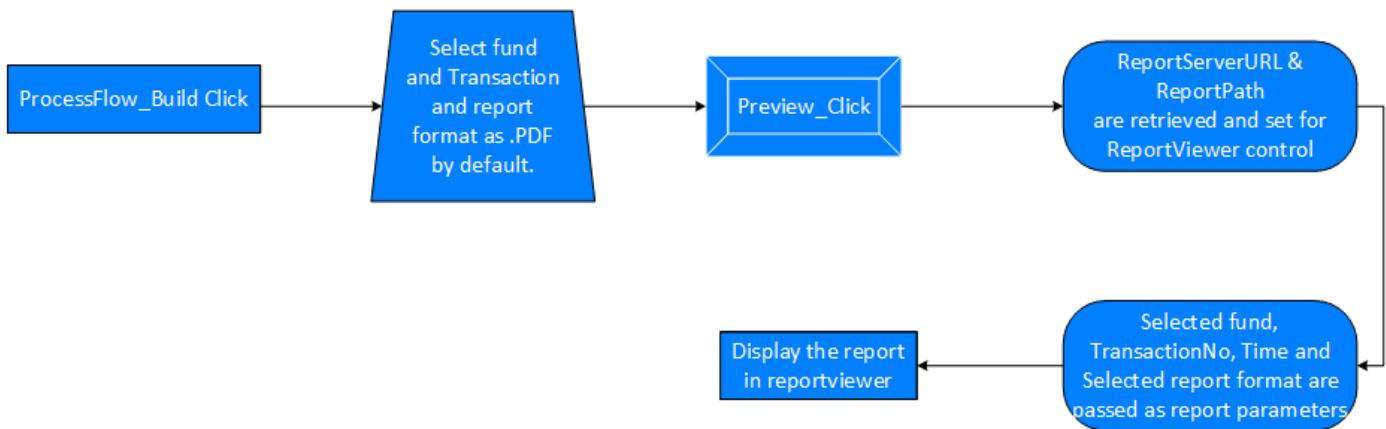


Figure 37: Deal Slips 'Preview Button Click' - Process Flow Diagram

Sequence diagram for Deal slips 'preview button click'-

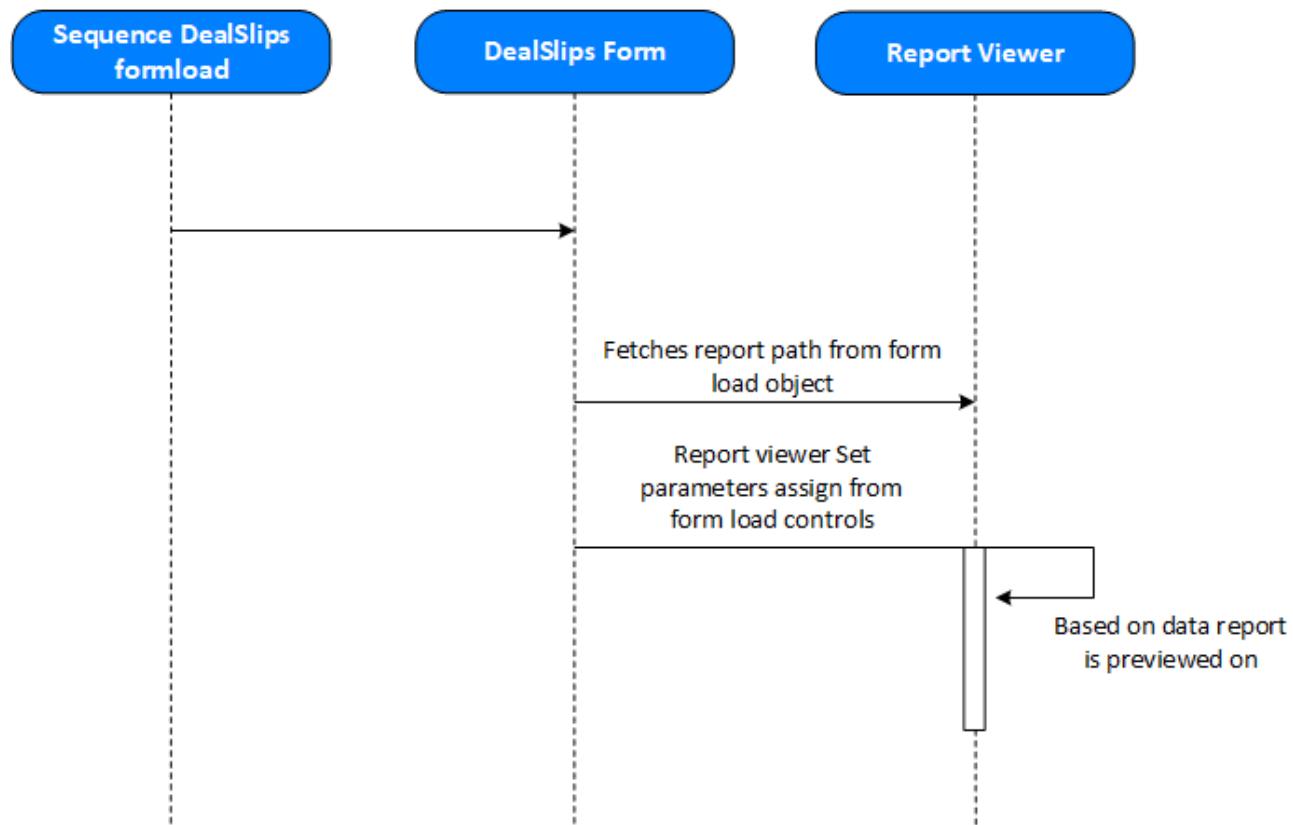


Figure 38: Deal Slips 'Preview button click' - Sequence Diagram

4.4.5 Deal Slips – GetTransaction button click

Upon clicking the Get Transactions button, it calls a method 'GetTransacionInfo' with parameter list (FundId.CoCode, startdate and EndDate), that calls the 'T24 processquery' transaction by Framing the OFS string as the parameter and if it retrieves data, it will list the Transaction in the List box and clones the same and fill the Transaction drop down list with the Transaction numbers.

The control comes back to the calling method and checks If transactions !=null and transaction Count > 0, then all the btnBuild, btnEmail, btnPreview and gbxTransactions will get enabled else btnBuild, btnEmail, btnPreview and gbxTransactions will get disabled and throws and error message saying, 'There are no transactions for this Fund and Date Range.', 'No Records Found'

Process flow diagram for Get Transactions Button Click-

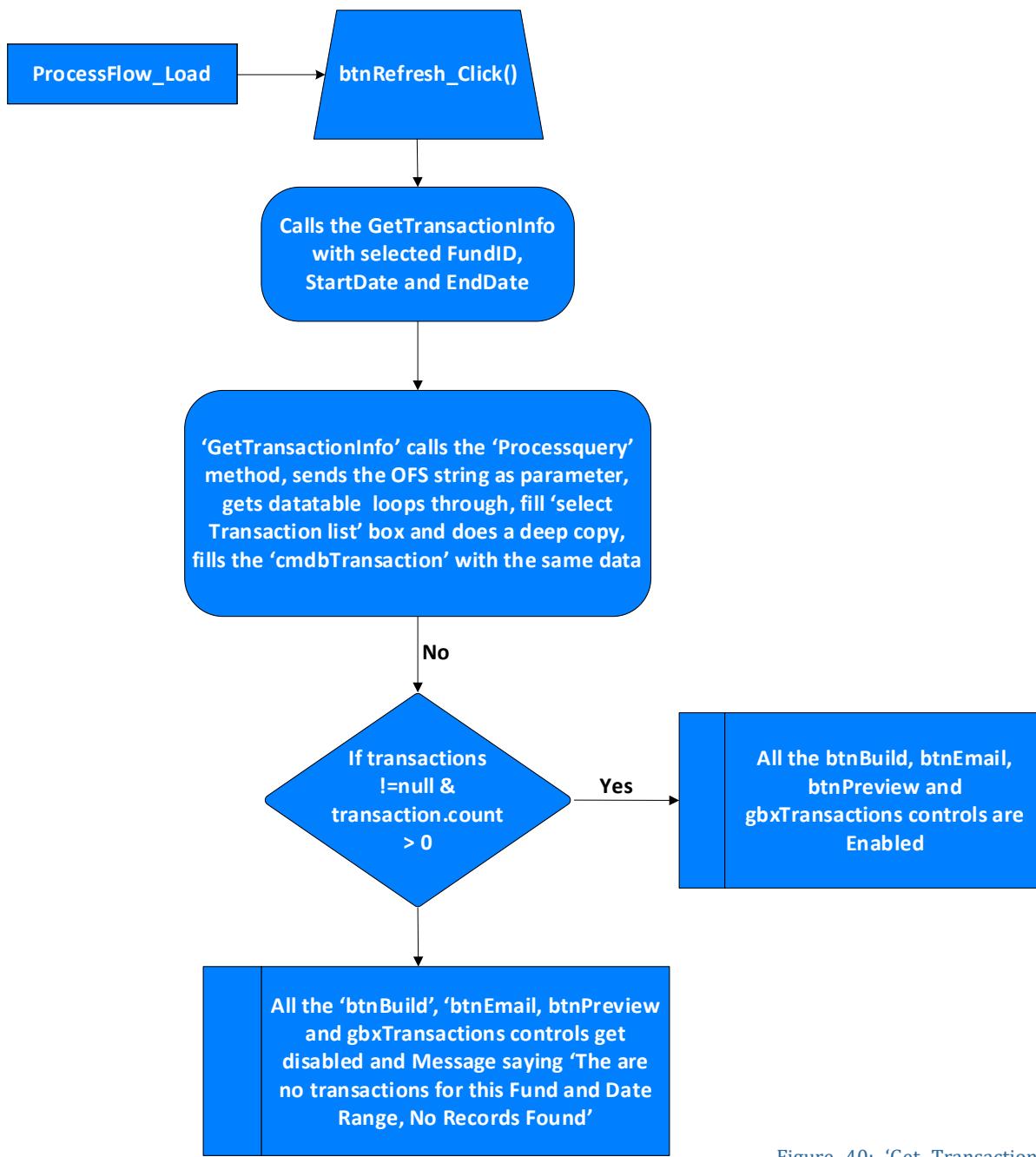


Figure 40: 'Get Transaction button click' -

Process Flow Diagram

Sequence diagram for Get Transactions Button Click-

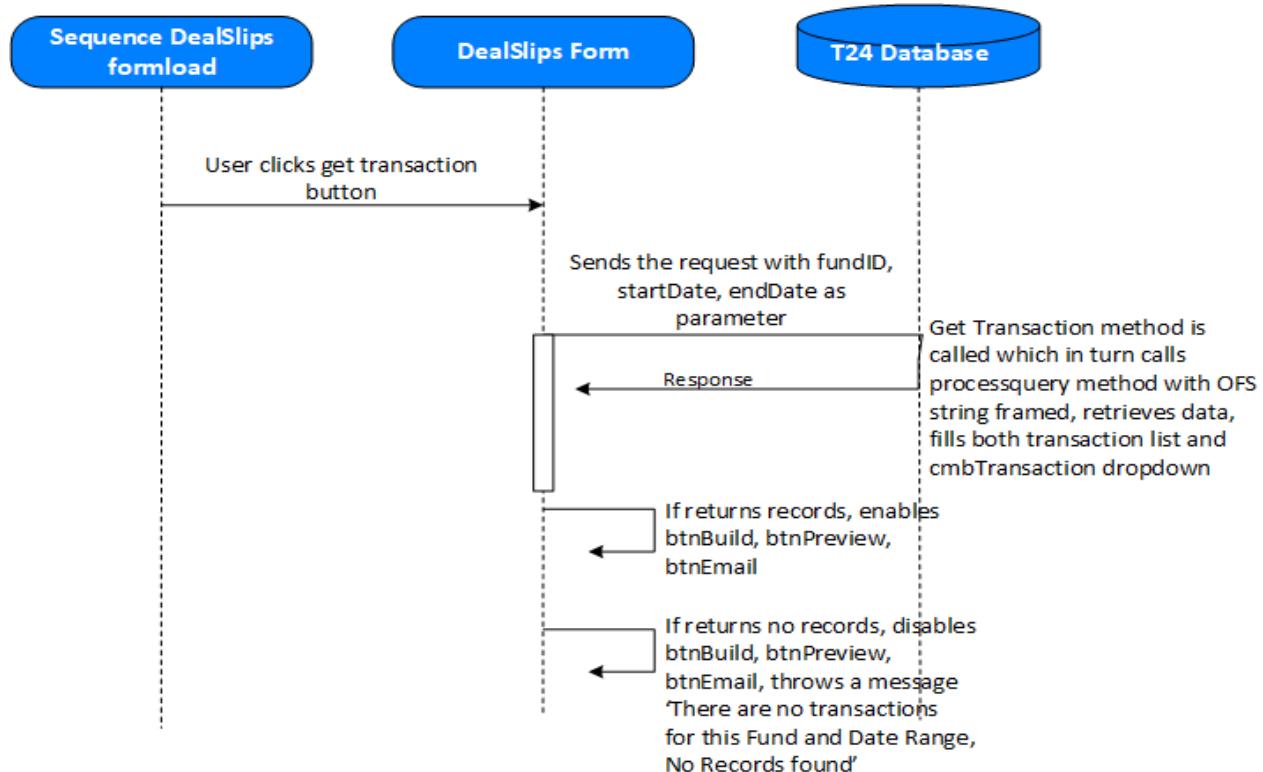


Figure 39: 'Get Transaction button click' - Sequence Diagram

4.4.6 DomRepDealSlip report flow – Build button click

The DR Reporting DealSlip report (DomRepDealSlip.rdl) involves data to be provided for input parameters and data fetched from SSRS to save the reports. All the data required for input parameters are loaded in form load event. The table 'tblFundSSRSUrl(SAFI_DB)' and column 'StatementSSRSUrl' provides the link to generate reports in the form of PDF and is saved to the disk. Once the parameters are selected from the form controls, the user clicks on 'Build' button. The URL is retrieved from table 'tblFundSSRSUrl' for the reports based on Co_Code of the selected fund. This URL's placeholder parameters, substitutes with values of user DomRepDealSlipselected 'client/clients', then HTTP request to SSRS server for Client Statement report is sent and the response is saved as PDF report.

4.4.7 DomRepDealSlip report – Build Button flow

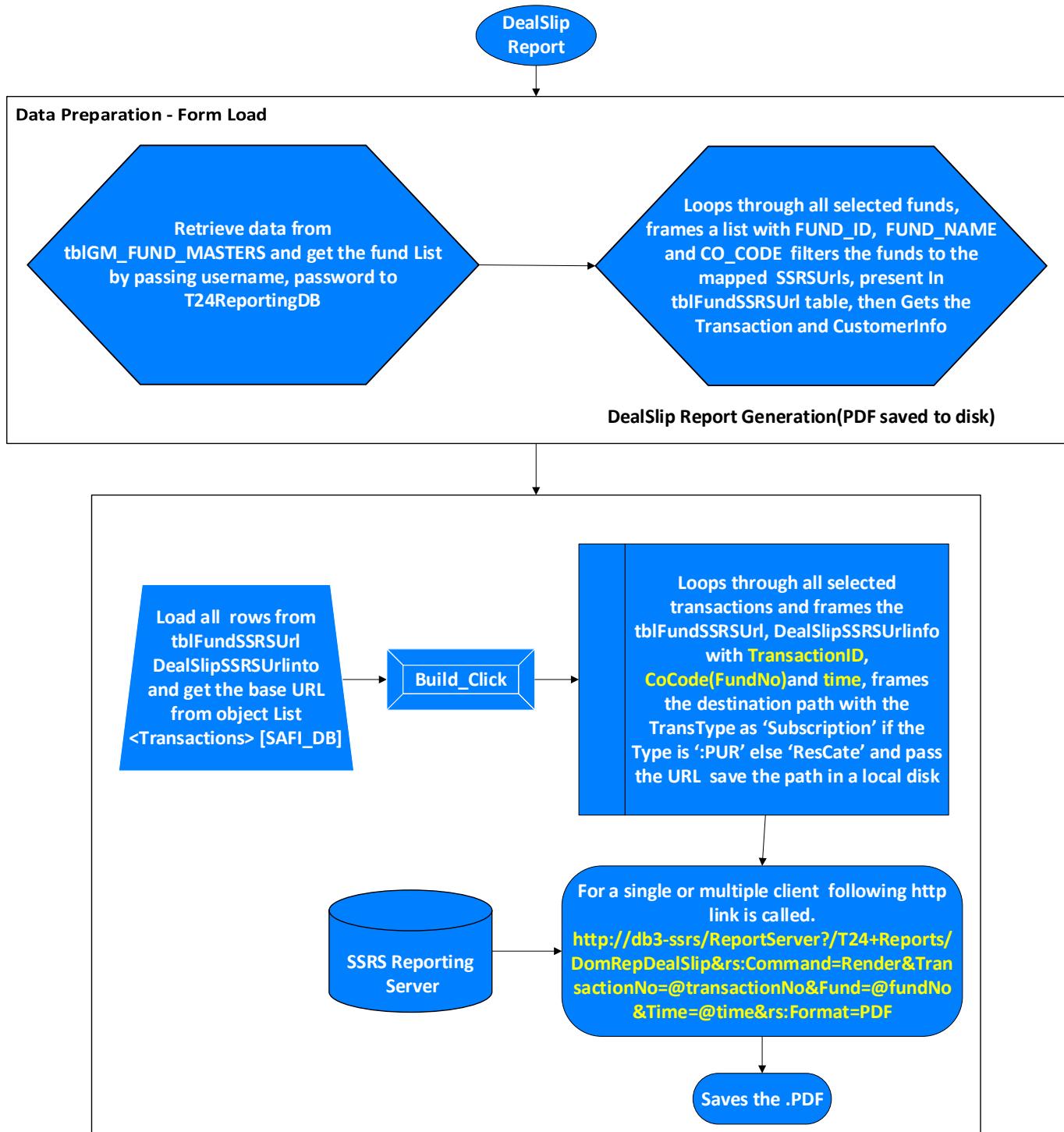


Figure 40: DomRepDealSlip Report Flow – Build button click

4.4.8 DomRepDealSlip report flow for Preview button click

The viewing of the DR Reporting DealSlip report (DomRepDealSlip.rdl) involves user selecting the fund and Time and Transaction in the form controls and then clicking 'preview'. The report will be visible in the report viewer control. This process is done by setting the path of the report by concatenating three columns from the table 'tblFundSSRSUrl' namely 'ReportServer', 'ReportPath' and 'StatementName'. The parameters required to render the report are attached to the report control from the values in form controls, After this the report is fetched from SSRS reporting server and rendered in C# windows application report viewer control

DomRepDealSlip review Preview button flow –

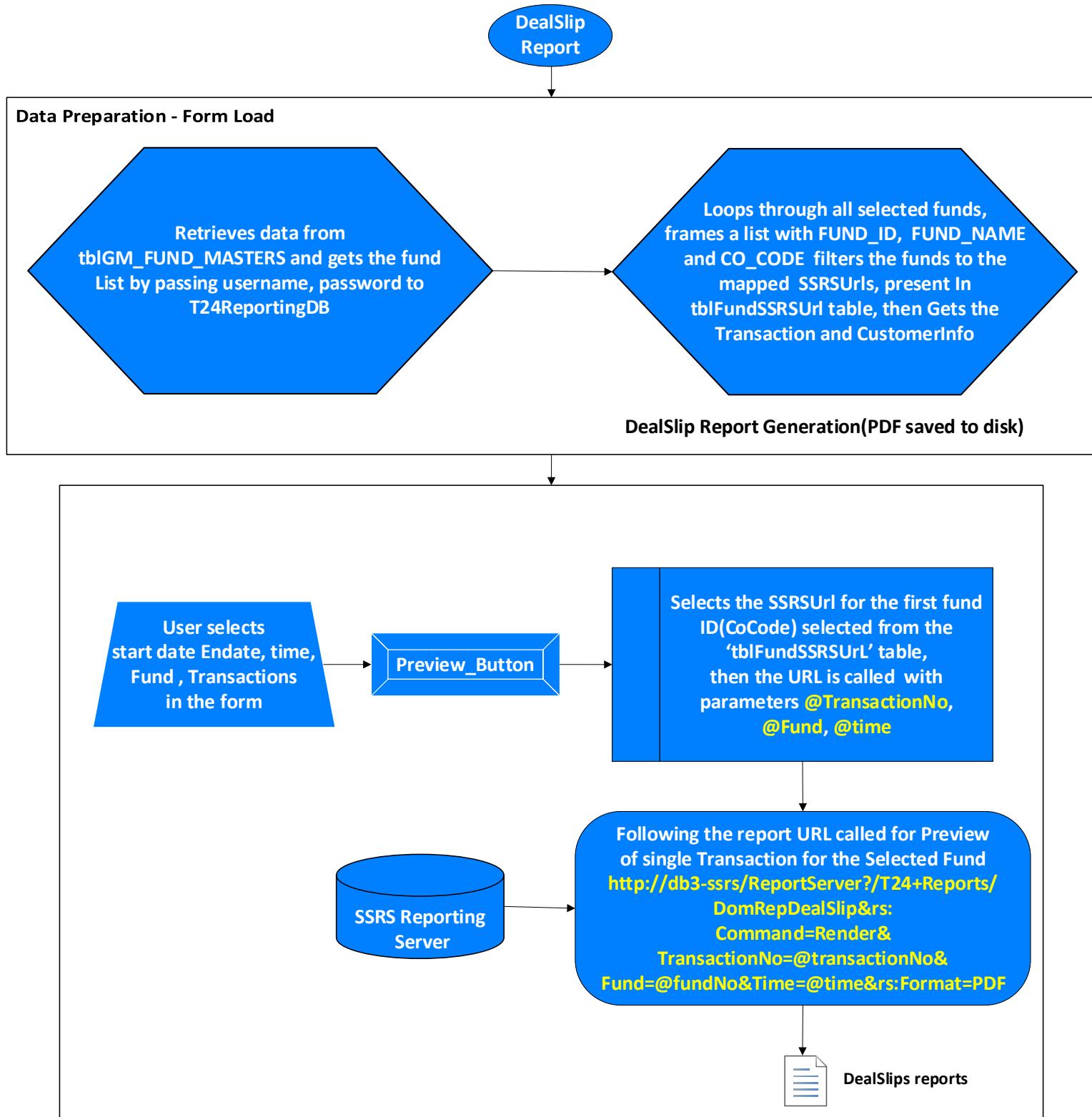


Figure 41: DomRepDealSlip Report Flow – Preview button click

UML Class Diagram for Get Transactions Button Click-

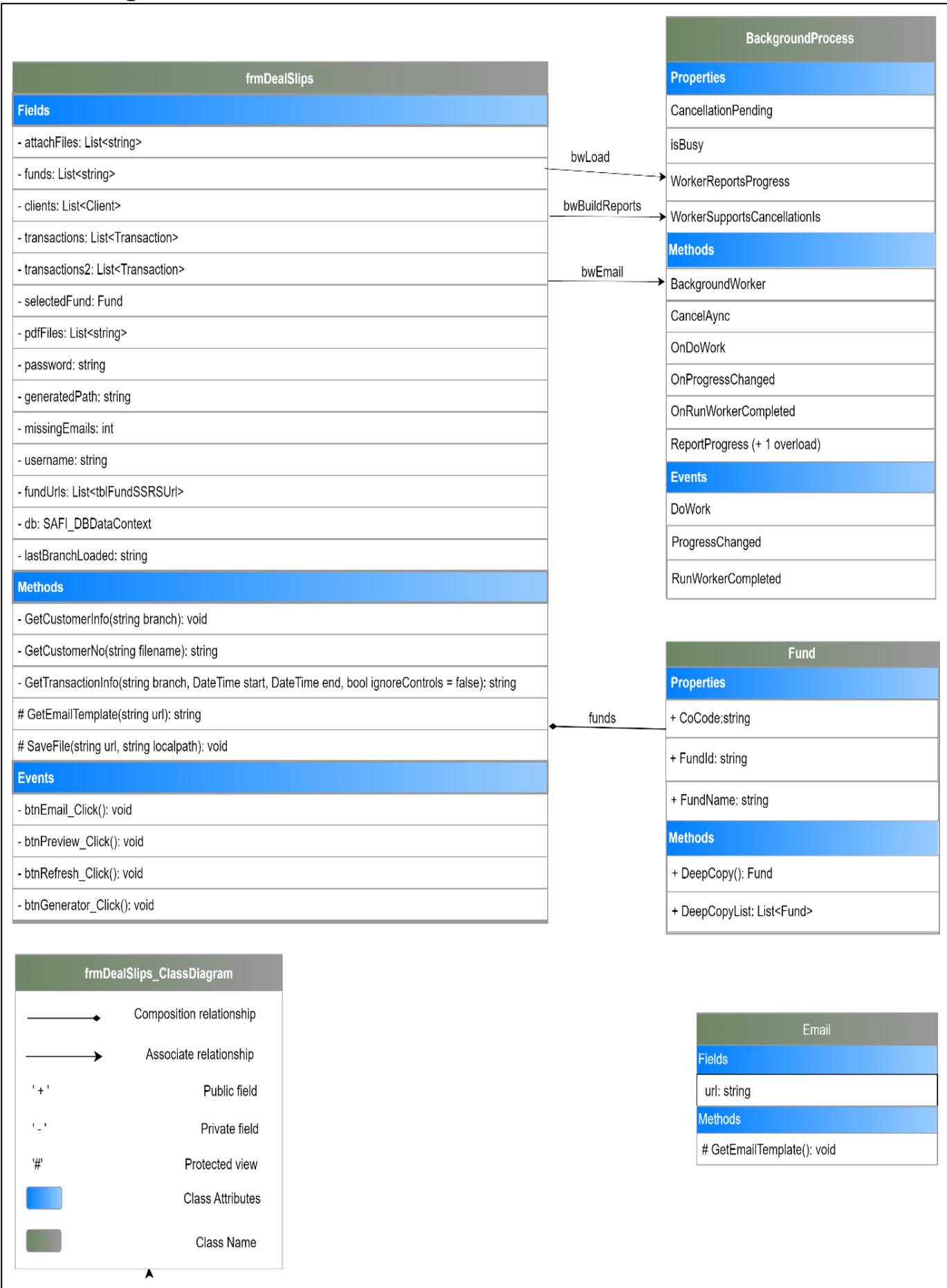


Figure 42: 'Get Transaction button click' - UML Class Diagram

ER Diagram for Deal Slips-



Figure 43: 'Get Transaction button click' - ER Diagram

4.5 DR Reporting Application – Regulatory report

Code level details for Regulatory Reports

A regulatory report demonstrates how is the performance of the fund and if they have met the relevant compliance requirements of a regulatory body or not. The regulatory report can be generated by logging into DR application in the MDIform (Dashboard), by selecting 'Regulatory Reports' from the Report Menu.

When user clicks on preview after selecting Fund, Report & report date, config data required for report is fetched from database and the returned report is viewed in the application form.

Form frmRegulatory has the below workflow:

- Allows building of reports after getting config data from database and fetching the reports from the Report server using config data and saves the reports to file system(disk).
- The reports are generated to the file system in the following path
c:\DRStatements\Documents\FundSys\ {Selected fund CoC0de}\Statements\{endDate}

The regulatory report generator form view is below-

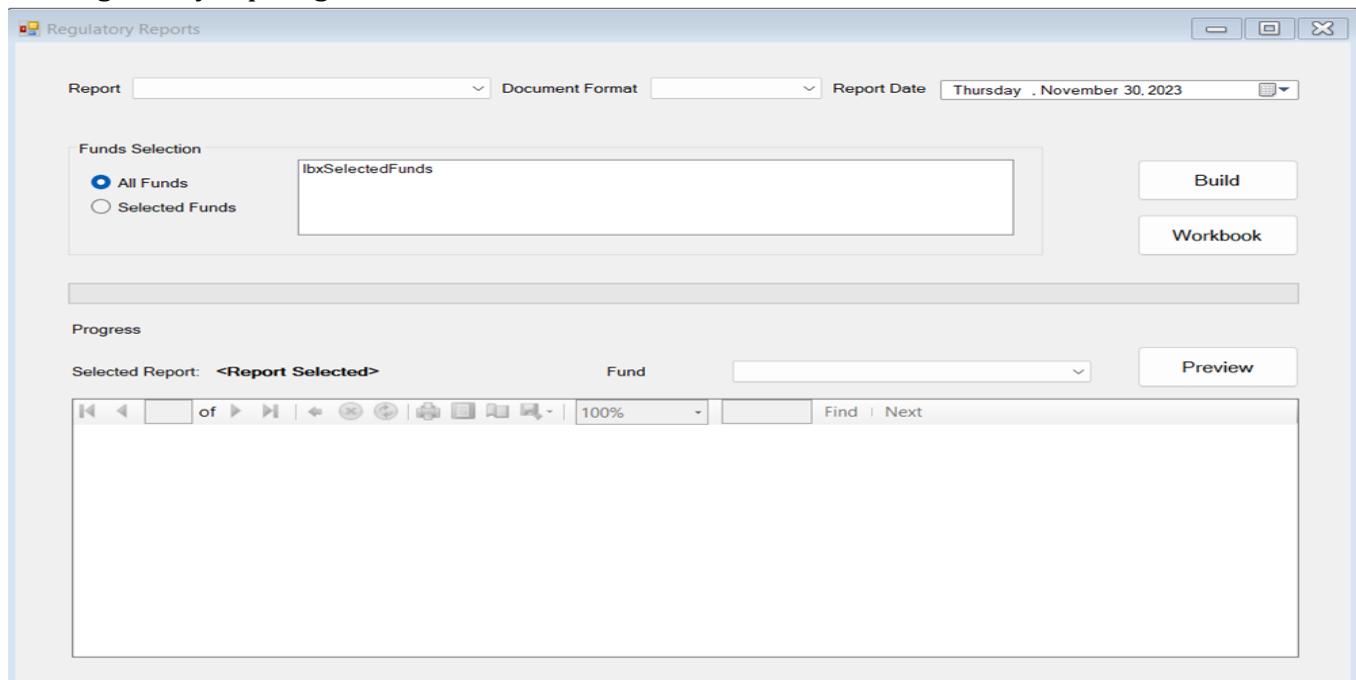


Figure 44: DRReporting application View – Regulatory Reports

The regulatory report generation form has four action methods:

1. Form Load
2. Build button click.
3. Preview button click.
4. Workbook button click

4.5.1 Regulatory Report – Form Load

During the form load process, the application creates the two Linq2Sql dB context, one for SAFI_DB and another for T24DataSource. In the form load event of regulatory report generation, a background process is initiated which fetches the funds data and loads the data into funds combo box.

Process flow diagram for Regulatory report generator 'Form Load action'

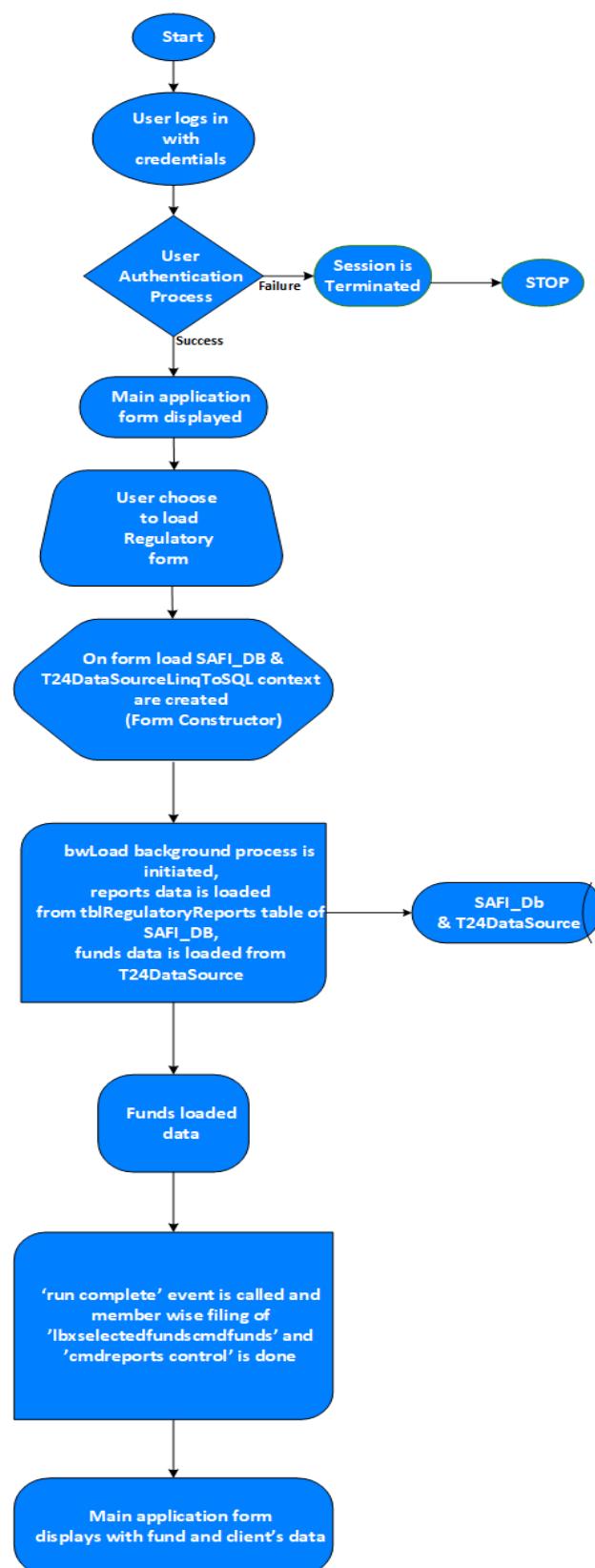


Figure 45: frmRegulatory 'Form Load action' – Process Flow Diagram

Sequence Diagram for Regulatory report generator 'Form Load action'-

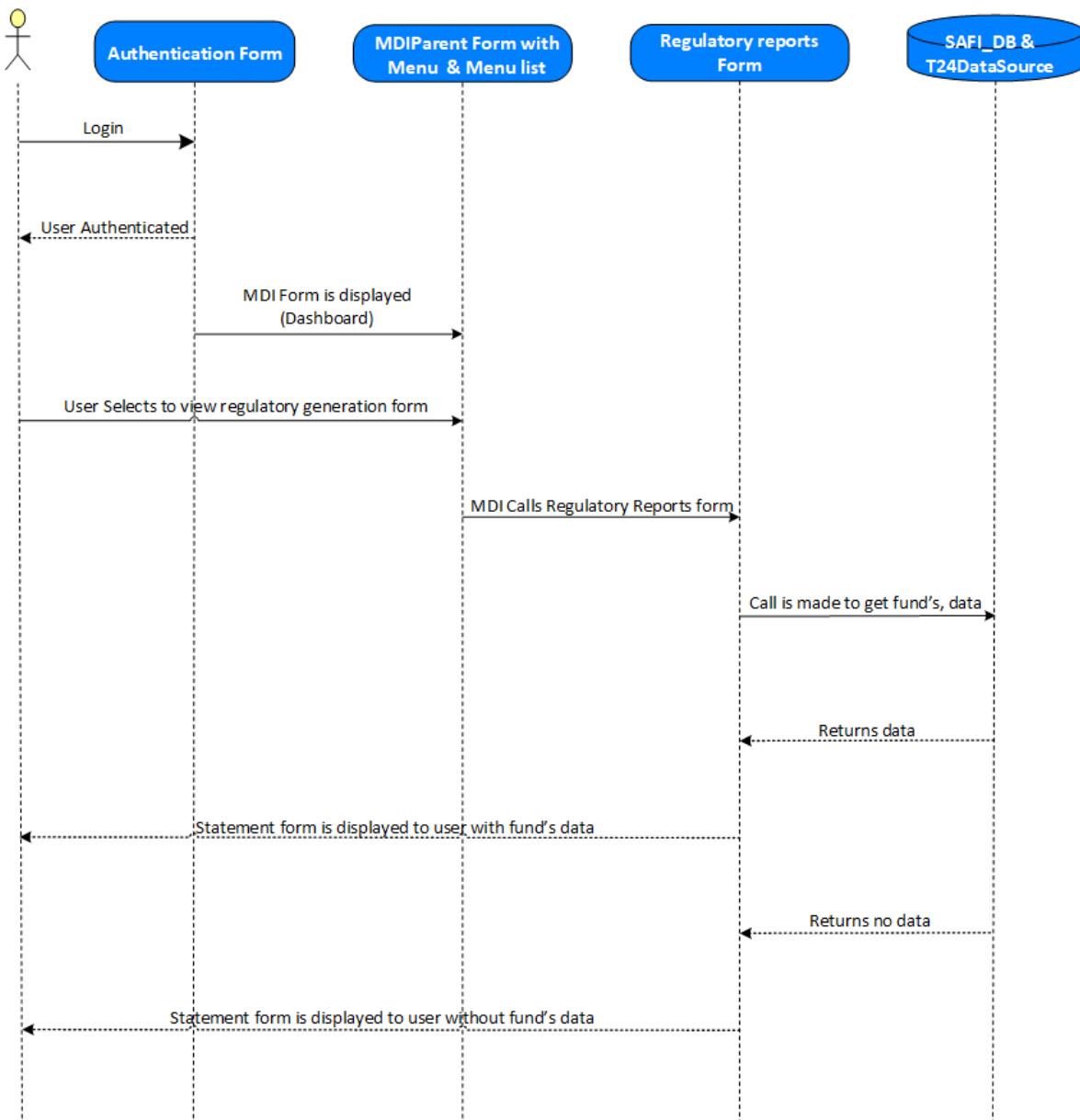


Figure 46: frmRegulatory 'Form Load action' – Sequence Diagram

4.5.2 Regulatory Report – Build button click

To start the 'build' process the end user has to select a report and fund in the combo boxes, then set a report date from the date fields. After this when user clicks on 'build', he is greeted with a confirmation message box; by selecting 'yes', the creation of folder with different path is done by checking if they exist. The data stored during form load is used to get URL for fetching data and same is stored in the local path, with the selected report format.

Process Flow diagram for Regulatory Report 'Build Button Click'-

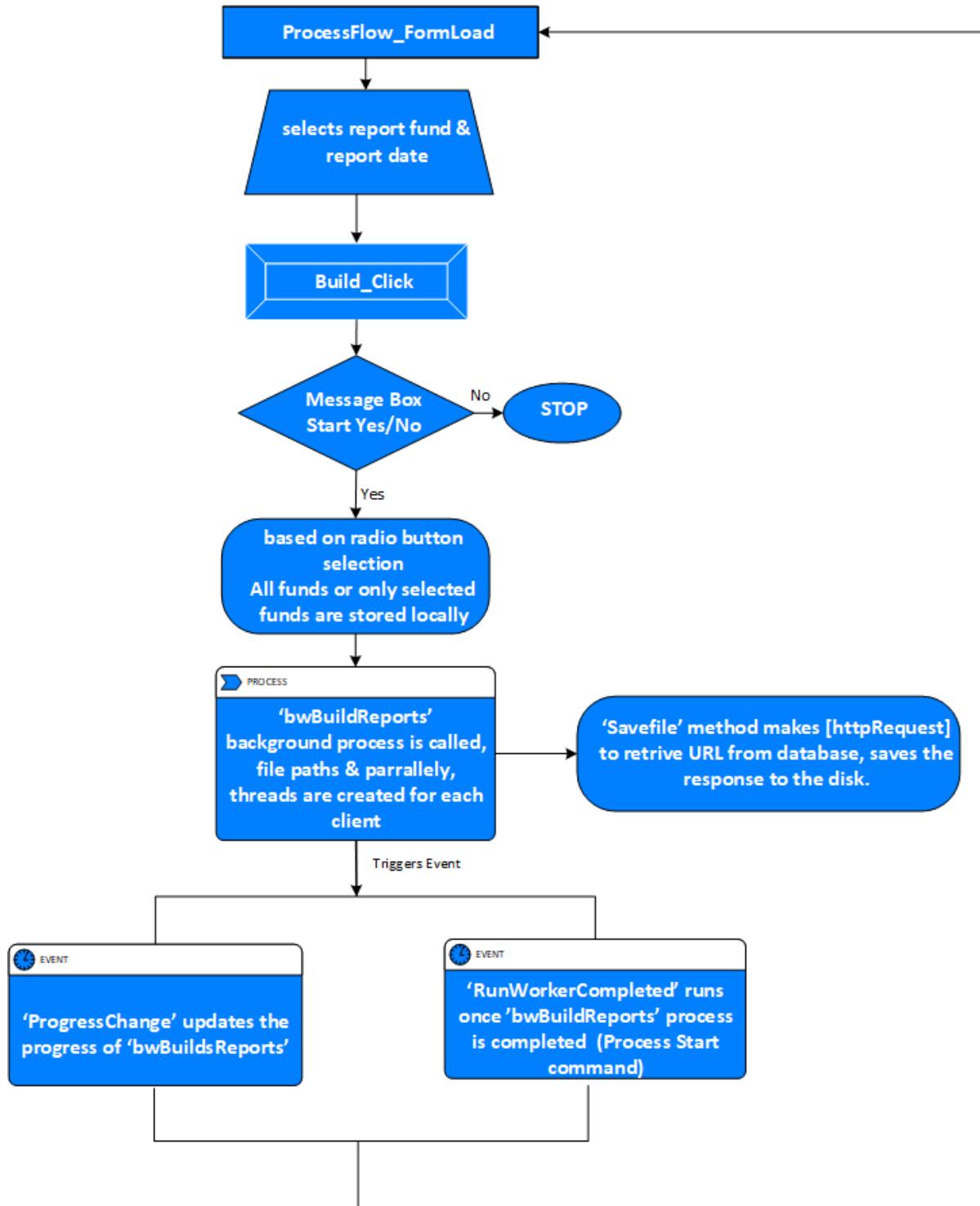


Figure 47: frmRegulatory 'Build Button Click' – Process Flow Diagram

Sequence Diagram for Regulatory Report 'Build Button Click'

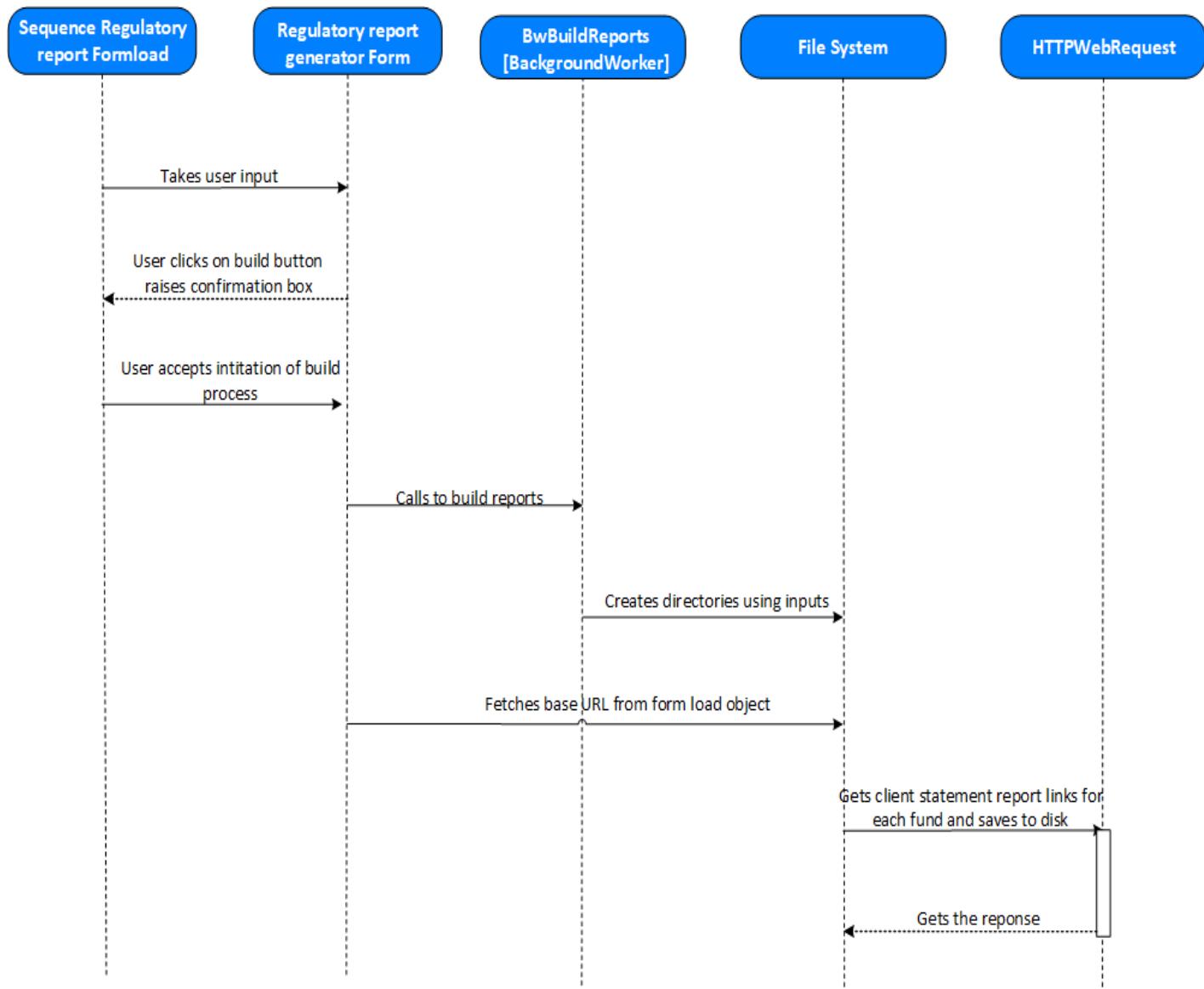


Figure 48: frmRegulatory 'Build Button Click' – Sequence Diagram

4.5.3 Regulatory Report – Preview button click

Here the report will be displayed in the 'Report Viewer control' on the form after selecting fund and the respective report.

Process flow diagram for Regulatory Report 'Preview button click'-

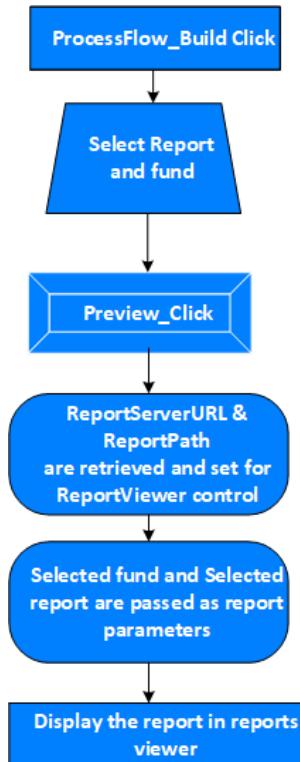


Figure 49: frmRegulatory 'Preview button click' – Process Flow Diagram

Sequence diagram for Regulatory Report 'Preview button click'-

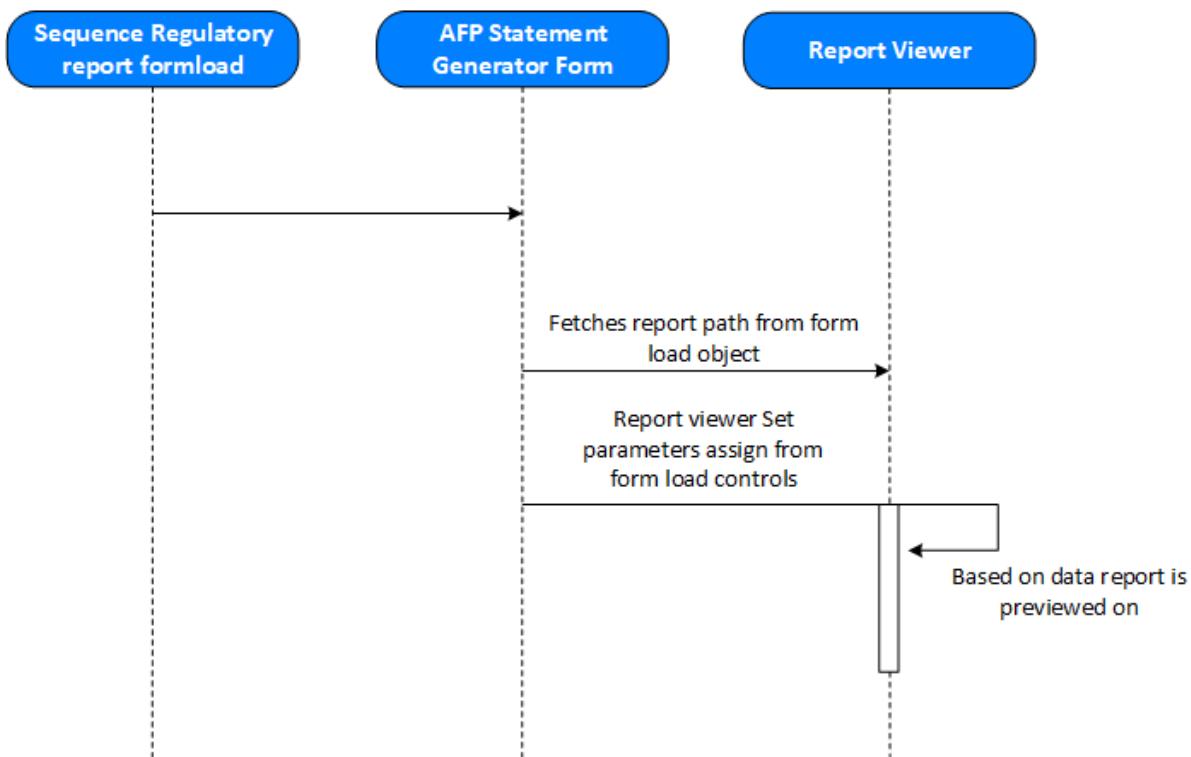


Figure 50: frmRegulatory 'Preview button click' – Sequence Diagram

4.5.4 Regulatory Report – Workbook button click

Here the application builds a workbook and adds a new sheet for each fund and generates in the application path.

Process flow diagram for Regulatory Report 'Workbook button click'-

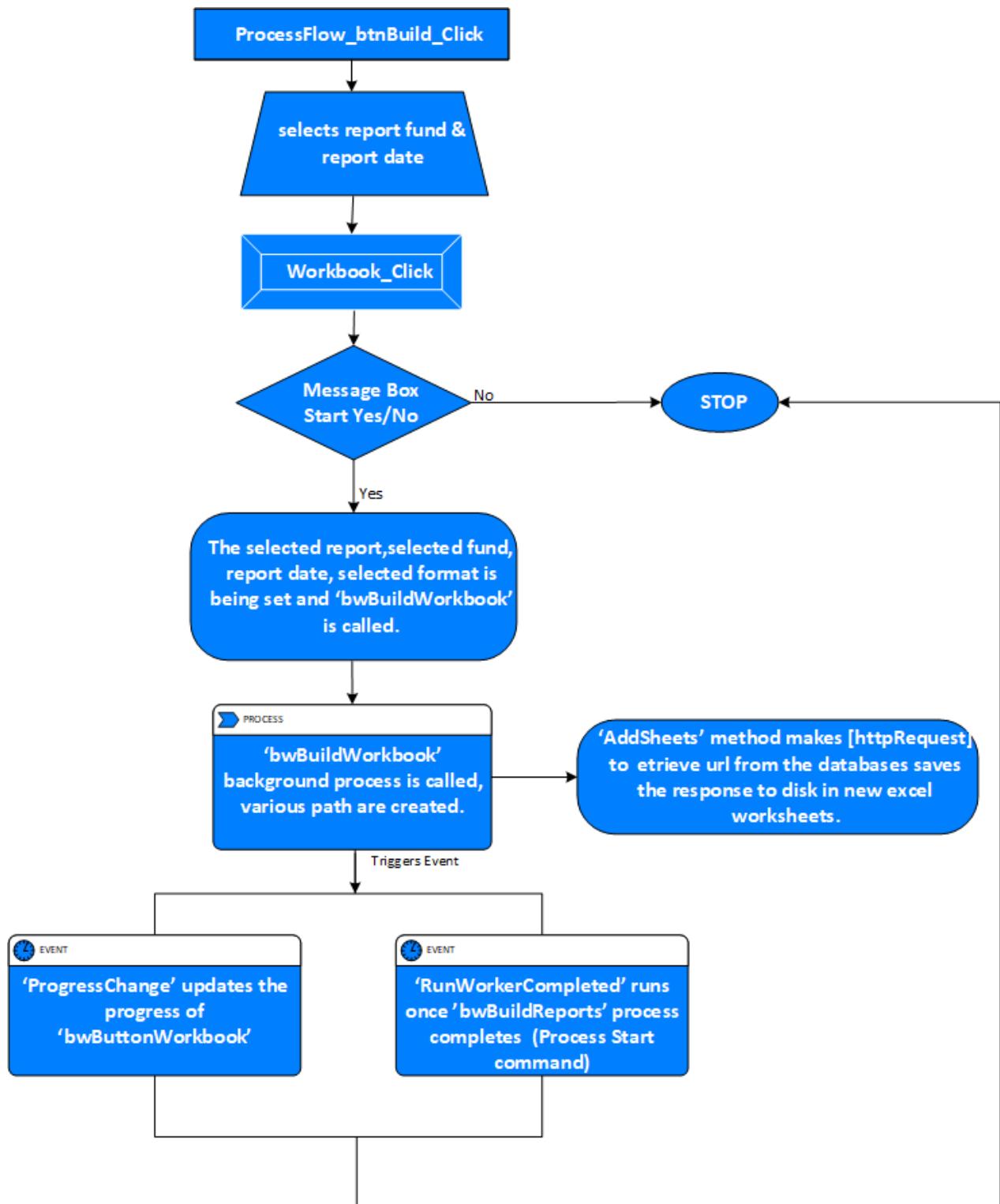


Figure 51: frmRegulatory 'Workbook button click' – Process Flow Diagram

Sequence diagram for Regulatory Report 'Workbook button click'

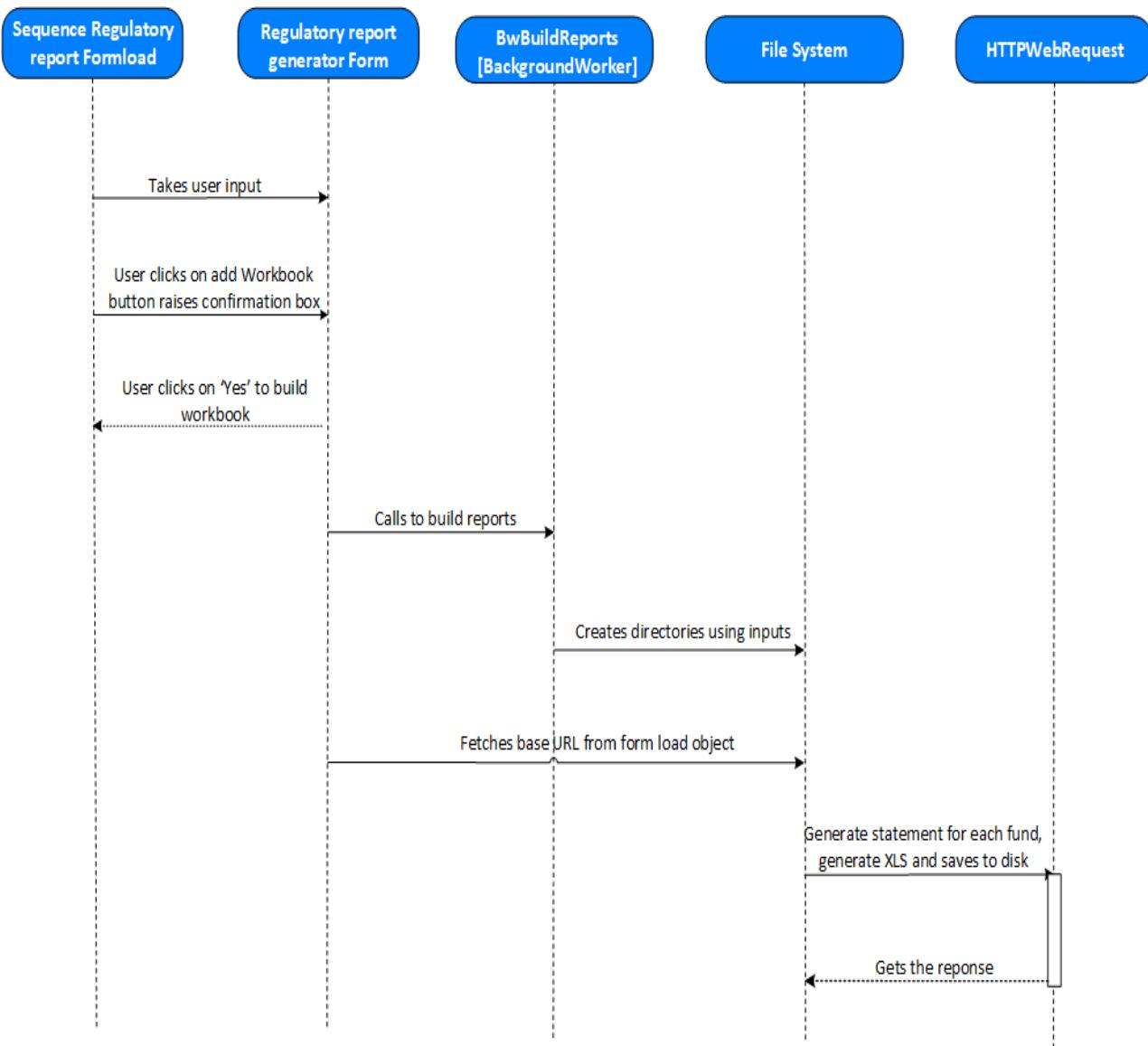


Figure 52: frmRegulatory 'Workbook button click' – Sequence Diagram

4.5.5 InformacionDiariaDePublicar Report – Build button click

The regulatory statement report (InformacionDiariaDePublicar.rdl) involves data to be provided for parameters and data fetched from SSRS to save the reports. All the data required for parameters is loaded in form load event. The tblRegulatoryReports (SAFI_DB) table SSRSUrl column provides the link to generate reports in form PDF saved to disk. Once the parameters are selected from the form controls and the user clicks Build button, tblRegulatoryReports provided a URL for report based on selected fund & report date, this URL's placeholder parameters or substitute with values of user selected fund/funds, then an HTTP request to SSRS server for InformacionDiariaDePublicar report is sent and the response is saves as PDF report.

Process flow diagram for InformacionDiariaDePublicar report ‘Build button click’-

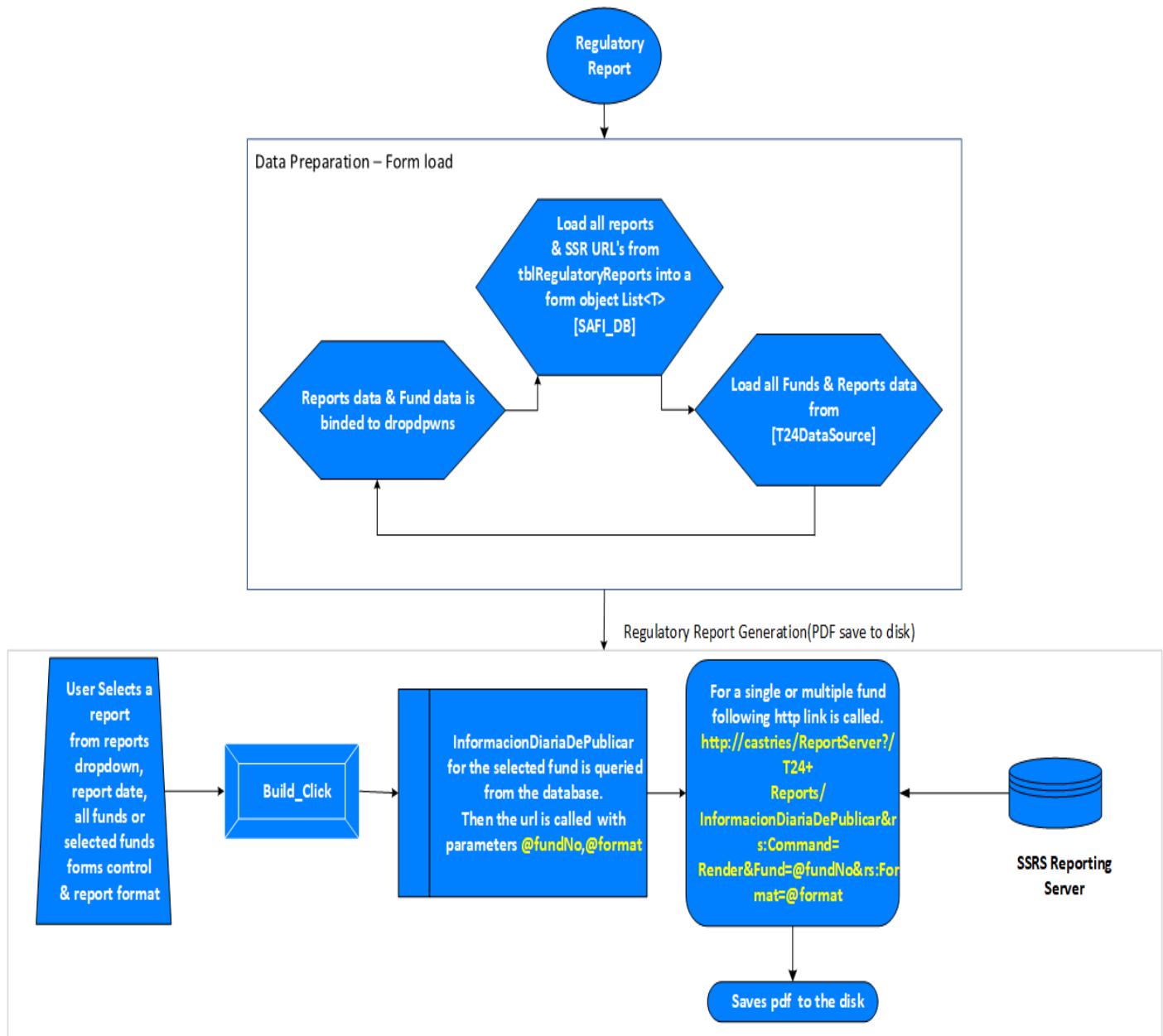


Figure 53: **InformacionDiariaDePublica** File ‘Build button click’ – Process flow

4.5.6 InformacionDiariaDePublicar Report – Preview button click

The viewing of the InformacionDiariaDePublicar report (InformacionDiariaDePublicar.rdl) when the user select the fund and report in the form controls and click preview, and the report will be visible in report viewing control This process is done by setting the path of the report by, concatenating three columns of tblRegulatoryReport table namely ReportServer, ReportPath & ReportName. The parameters required to render the report are attached to the report control from the values in form control, after this the report is fetched from SSRS reporting server and render in C# windows application report viewing control.

Process flow diagram for InformacionDiariaDePublicar report 'Preview button'-

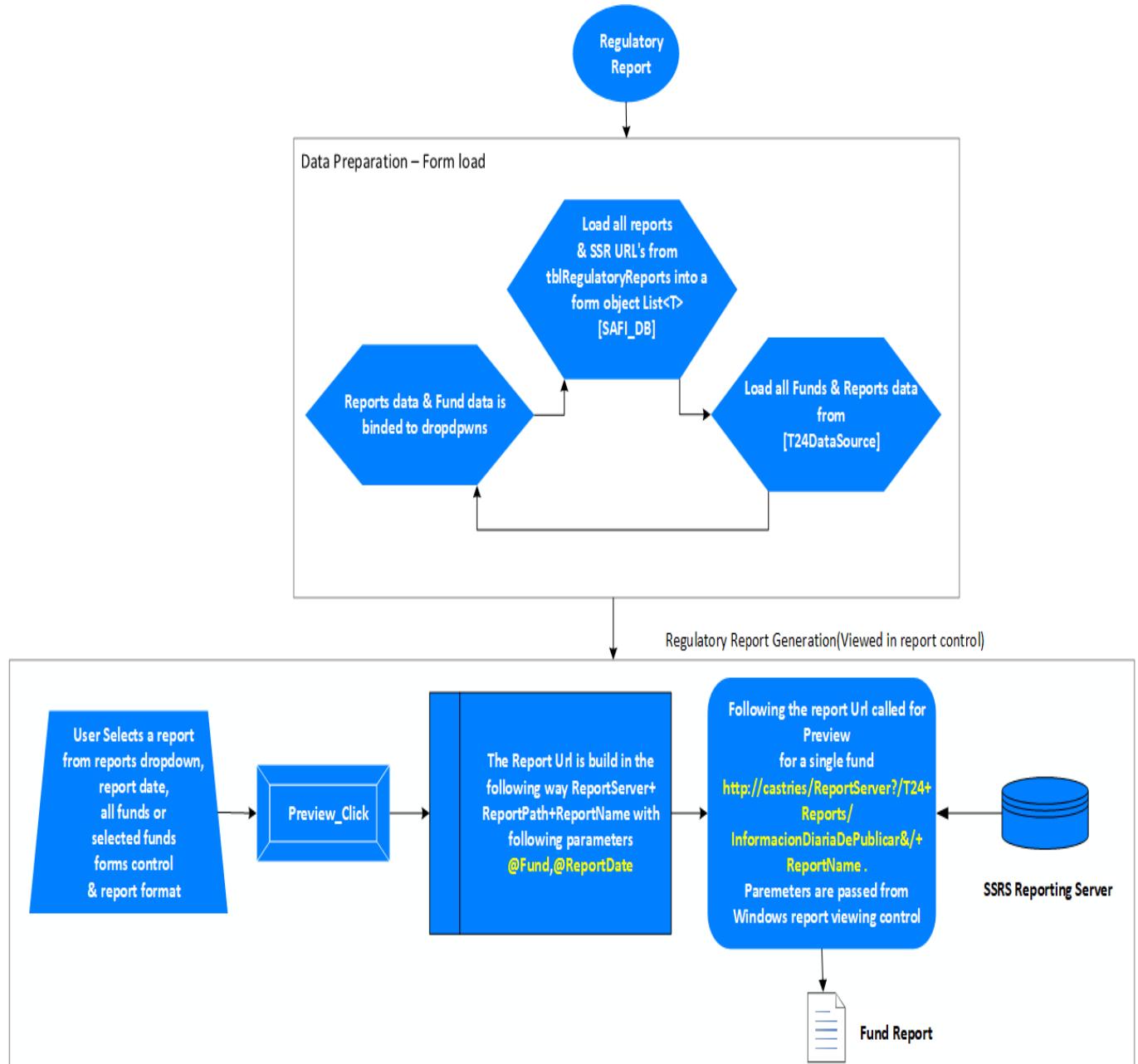


Figure 54: InformacionDiariaDePublicar Report 'Preview Button' - Process Flow

4.5.7 InformacionDiariaDePublicar Report – Workbook button click

Once the report is built then user can generate a XSLX file for all or the selected fund. For this user selects a report from the report's dropdown, report date and a fund or multiple funds and hits Workbook click. For each selected fund an XLSX file will be created. The report is passed with @fundNo, @format, @reportdate. The report link will be called single or multiple times to generate the report in separate work sheets in the excel file

Process flow diagram for InformacionDiariaDePublicar report ‘Workbook button’-

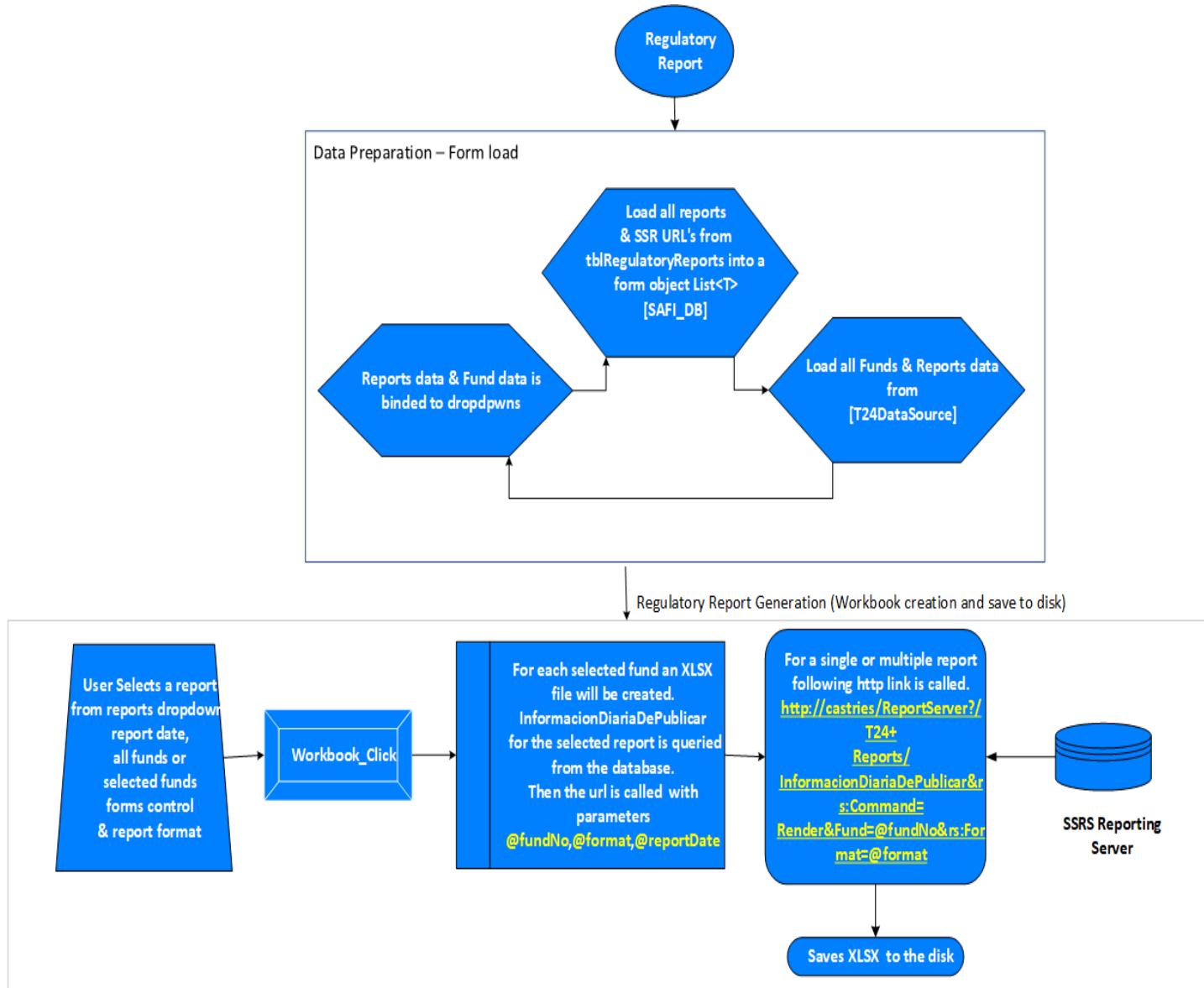


Figure 55: InformacionDiariaDePublicar report - ‘Preview button’ process flow

UML class level diagram details of the classes involved in frmregulatory-

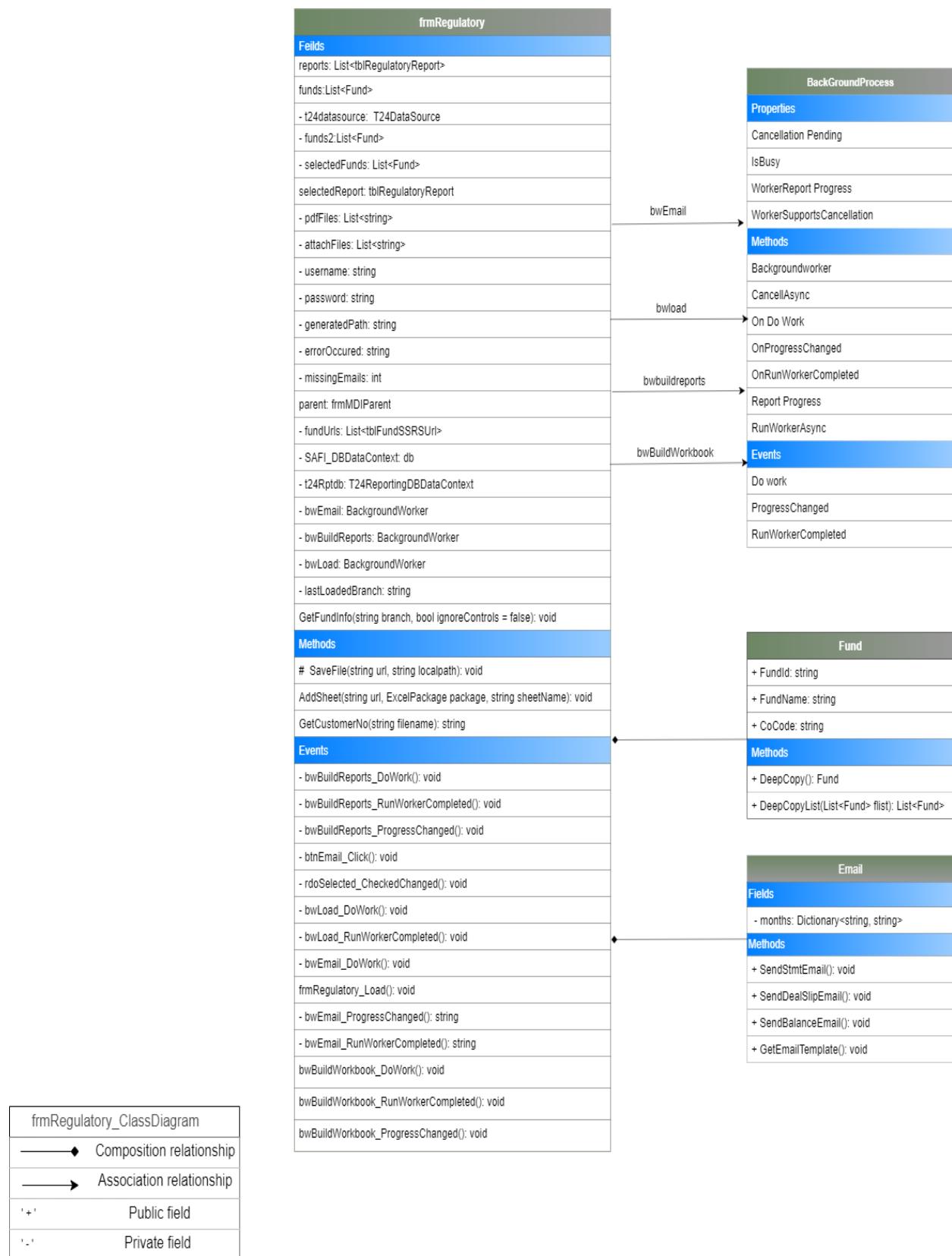


Figure 56: Regulatory Report – UML Class Diagram

ER Diagram for Regulatory report-

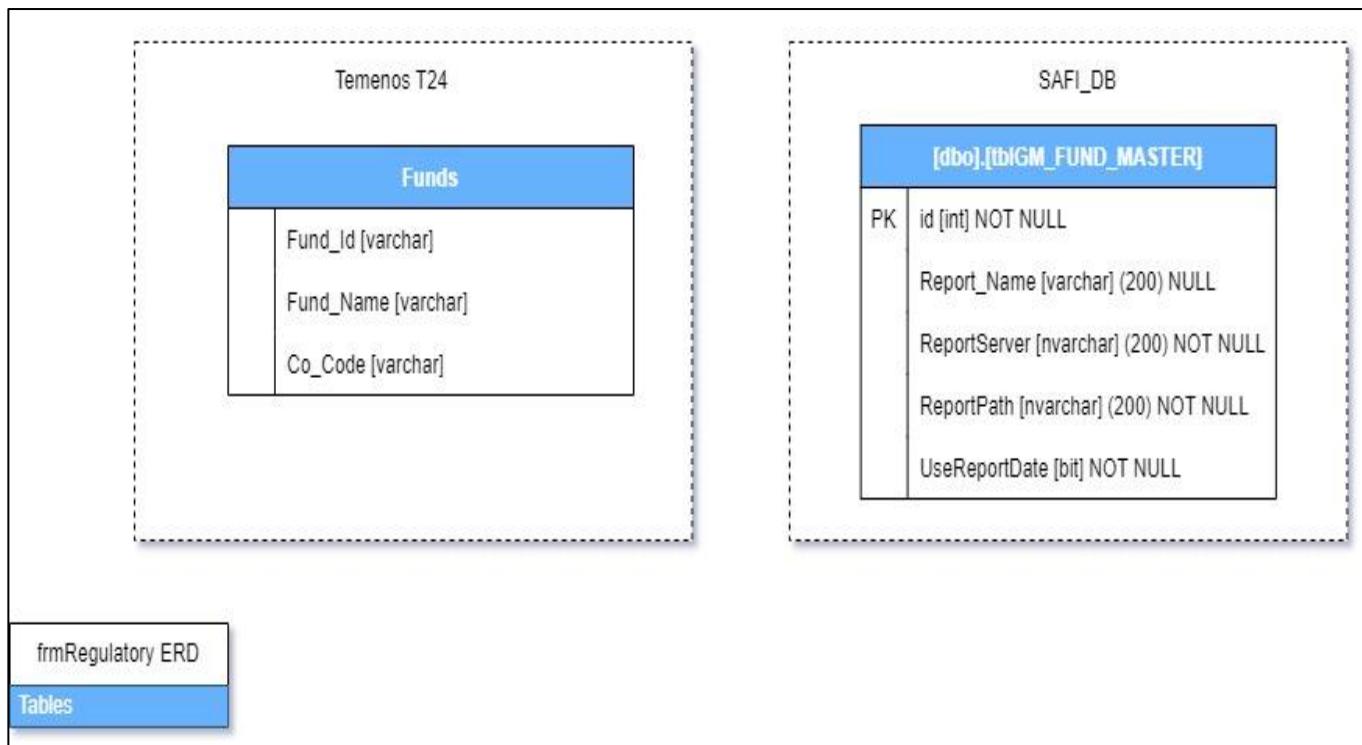


Figure 57: Regulatory Report – ER Diagram

4.6 DR Reporting Application – Balances report

Code level details for Balances report

The Balances Form can be generated by logging into DR application in the MDIform (Dashboard) and then selecting Balances from the Reports Menu.

The Balances Form is a component within a reporting application that provides users with a consolidated view of financial balances. Users can select clients and funds, preview balances, and perform actions such as generating reports or sending emails based on the displayed financial information. The form is designed to facilitate efficient navigation and decision-making in the context of financial reporting and analysis. When user clicks on preview after selecting Fund & Client, config data required for report is fetched from database and the returned report is viewed in the application form.

Form frmBalances does the following:

- Allows building of reports after getting config data from database and fetching the reports from the Report server using config data and saves the reports to file system(disk).
- Once the reports are generated to the file system in the following path
c:\DRStatements\Documents\FundSys\{Selected fund CoCode}\Statements\{endDate}
The user validates reports, then copies to the Email subfolder. The user may also put additional attachments in the OtherAttachments subfolder inside Email folder. The user then clicks on the Email button to send an email to each client whose statements is in the Email folder along with any attachments.

The Balances form view is below-

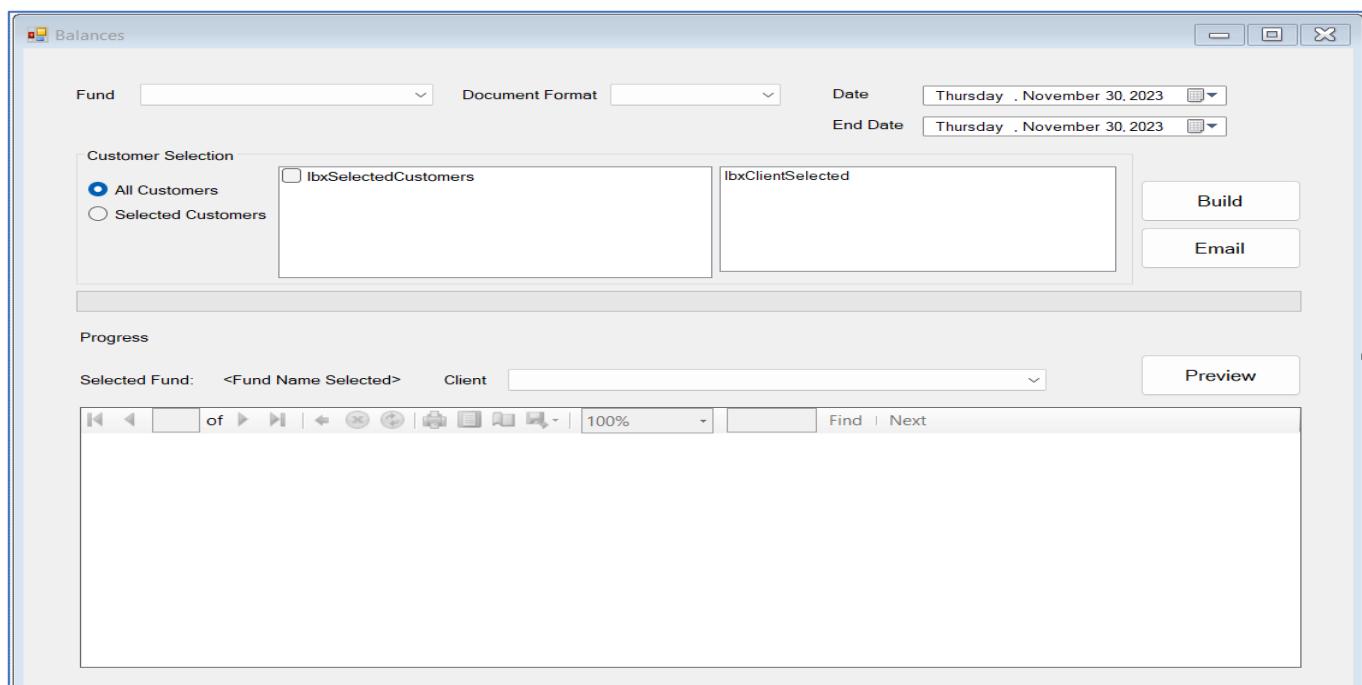


Figure 58: DRReportng Balances Report – Application view

The Balances generation form has four action methods:

-
1. Form Load
 2. Build button click.
 3. Email button click.
 4. Preview button click.

4.6.1 Balances Report – Form Load

During the form load process, the application establishes two ‘Linq2Sql database’ contexts, one for SAFI_DB and another for T24DataSource. Subsequently, in the form load event of the ‘Balances’ form, a background process is triggered. This process is responsible for retrieving client information and default selected fund data from the mentioned database context entities.

The fetched data is then loaded into the client combo box, ensuring that users have access to relevant client and fund information. This approach streamlines the user experience by preloading essential data, facilitating smoother interactions within the ‘Balances’ form.

Process Flow Diagram for Balances report 'Form Load action'-

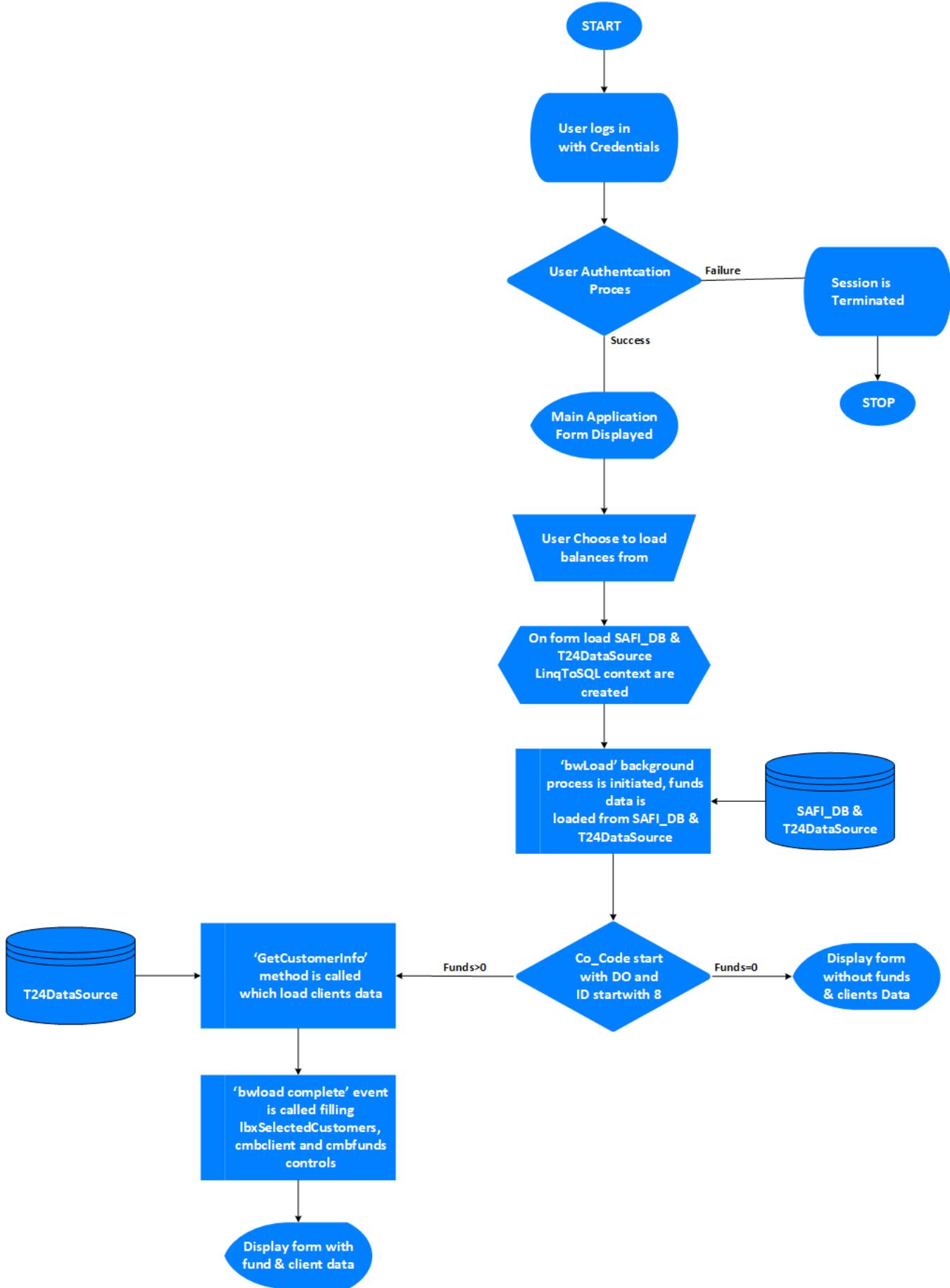


Figure 59: Balances Report 'Form Load action' - Process Flow Diagram

Sequence Diagram for Balances report 'Form Load action'-

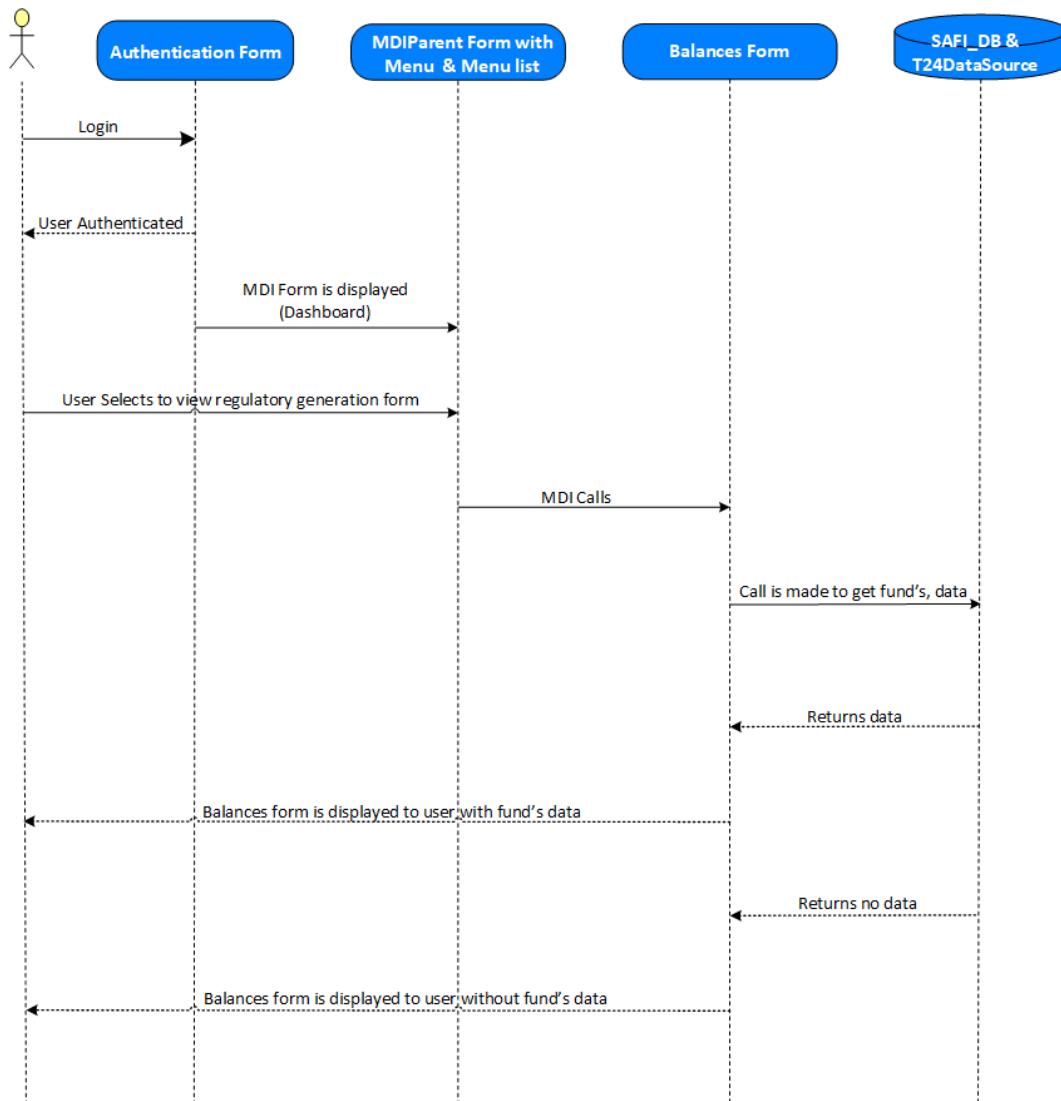


Figure 60: Balances Report 'Form Load action' - Sequence Diagram

4.6.2 Balances Report – Build button click

To initiate the construction process, the end user first selects a client and specifies a start date and end date using the provided date fields. Upon clicking the 'Build' button, a confirmation message box appears. If the user selects 'Yes', the system proceeds to create a designated folder, checking its existence before proceeding.

During the form loading phase, data is stored to serve as a URL for fetching additional information. Simultaneously, this data is saved to the previously created local path. The data is formatted in PDF for further processing.

Following the completion of background processes involved in generating the PDF reports, a 'complete' event is triggered. This event results in the display of the compiled PDF report.

Process Flow Diagram for Balances report 'Build button click'-

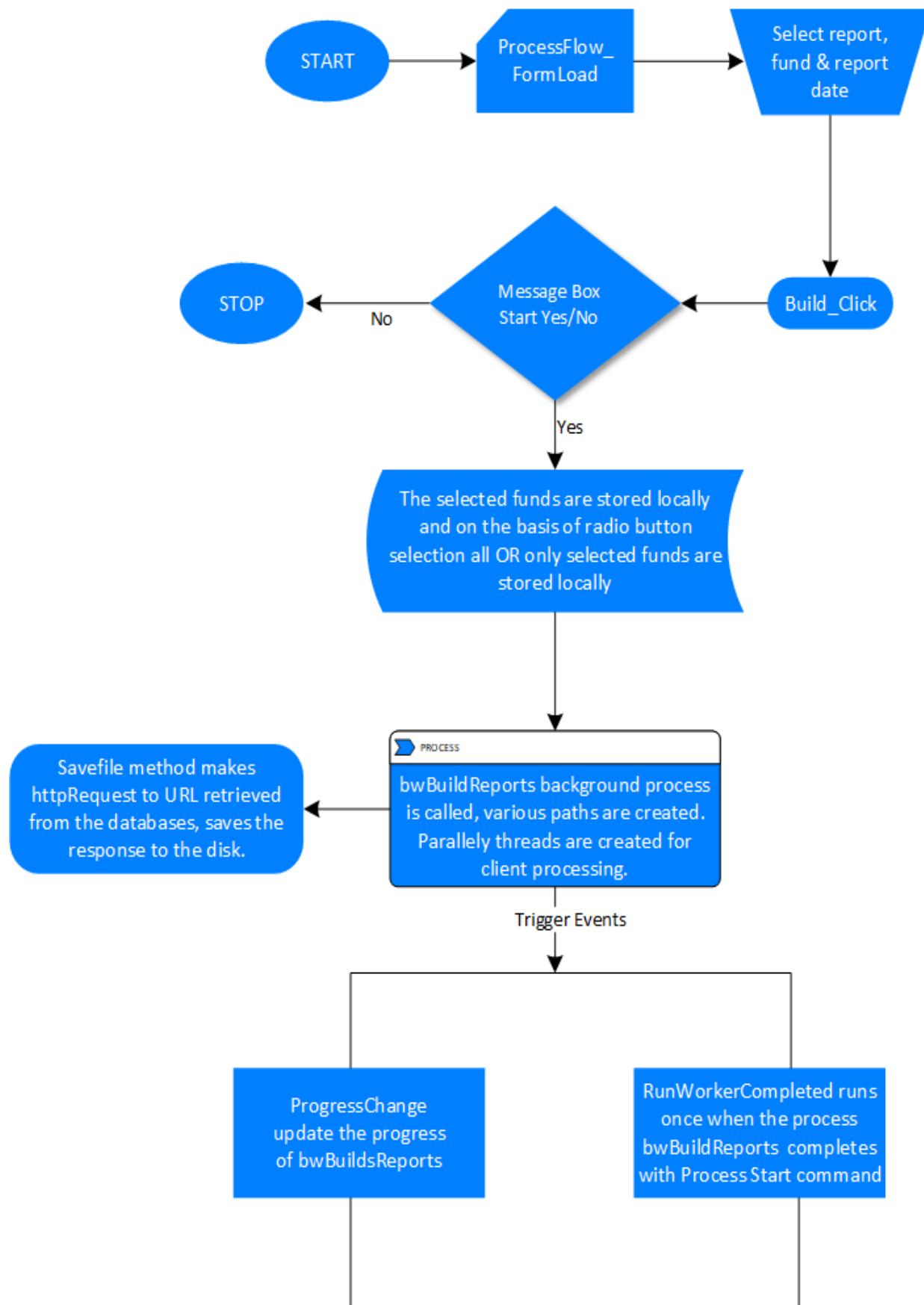


Figure 61: Balances Report 'Build button Click' - Process Flow Diagram

Sequence Diagram for Balances report 'Build button click'

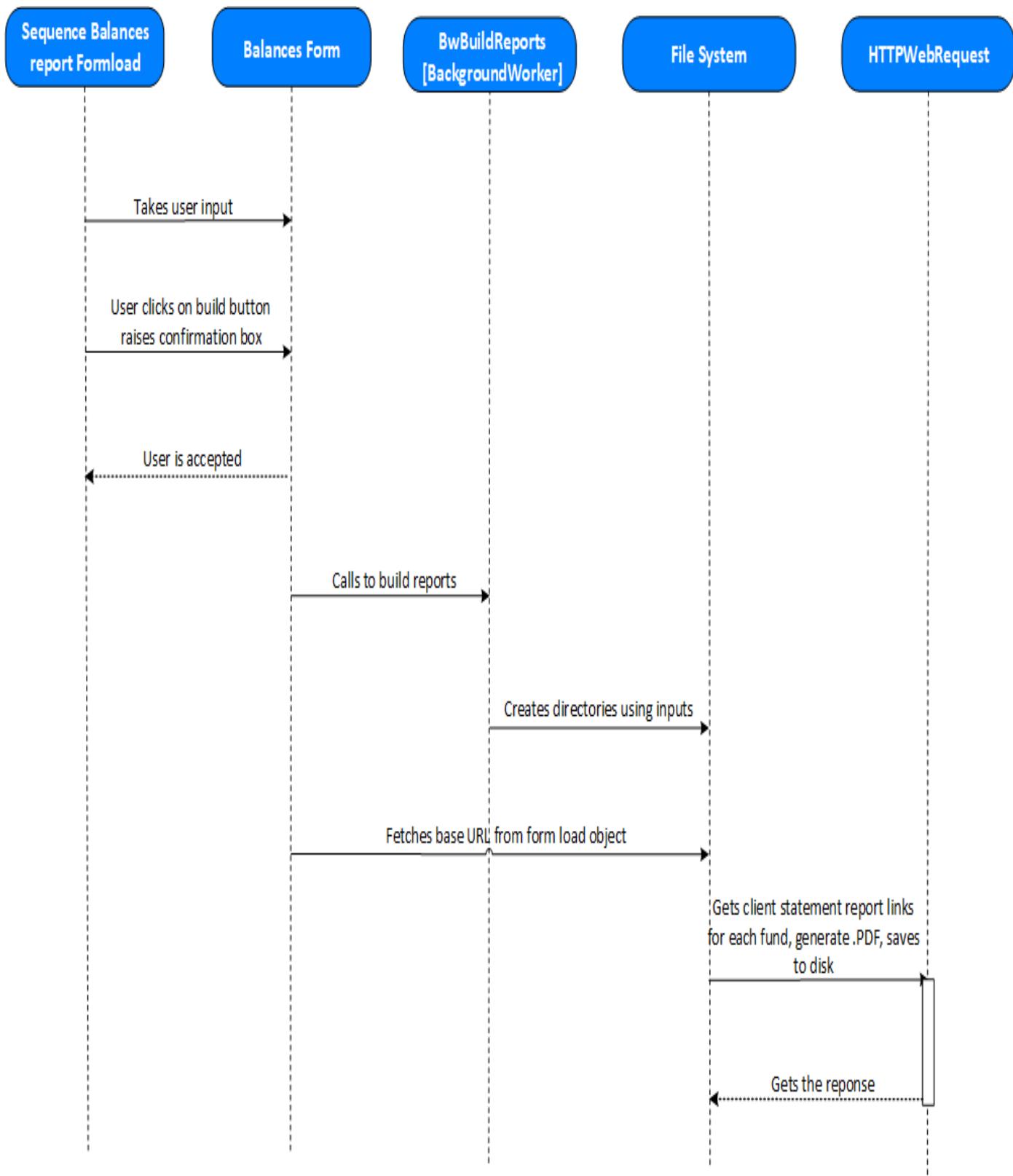


Figure 62: Balances Report 'Build button Click' - Sequence Diagram

4.6.3 Balances Report – Email button click

After executing the 'Build' button click event, all report files are stored in the designated file location. Subsequently, when the user clicks the 'Email' button, a confirmation message box appears. Upon selecting 'Yes' the system checks whether the email path directory exists. If it does, the system proceeds to retrieve all files from that path.

Following the retrieval, the system checks if the count of PDF files is greater than 0. If affirmative, it attaches the physical files to the email.

Event "Complete" is triggered once background process is done. An email template from the database is used to create the email that is sent, after the tags have been replaced as below, and email service is called to send email to respective clients.

The following Tags are replaced with data from the application

Tags	Data from the Application	
	Replaces	Subject Template
<Estado>		Balance
<y Reporte Mensual JMMB >		<blank>
<day>		endDate.ToString("dd")
<month>		months[endDate.ToString("MM")]
<year>		endDate.ToString("yyyy")
	Replaces Body Tempate (fundInfo.BalanceEmailTemplate)	
<CustomerNumber>		DRReporting.Client – CustomerNo
<StartDate>		endDate.ToString("yyyy-MM-dd")
<FundID>		tblFundSSRSUrl – fundInfo.FundID
<FundName>		fundInfo.StatementSubject
<Estado de Cuenta y Reporte Mensual >	<blank>	
<al <day> de <month> de <year>>	<blank>	
<DRSignatureTag>		fundInfo.BalanceSignature
<DRHeaderTag>		fundInfo.BalanceHeader

Table 5: Email Tags – Balances report

Process Flow Diagram for Balances Report 'Email Button click'-

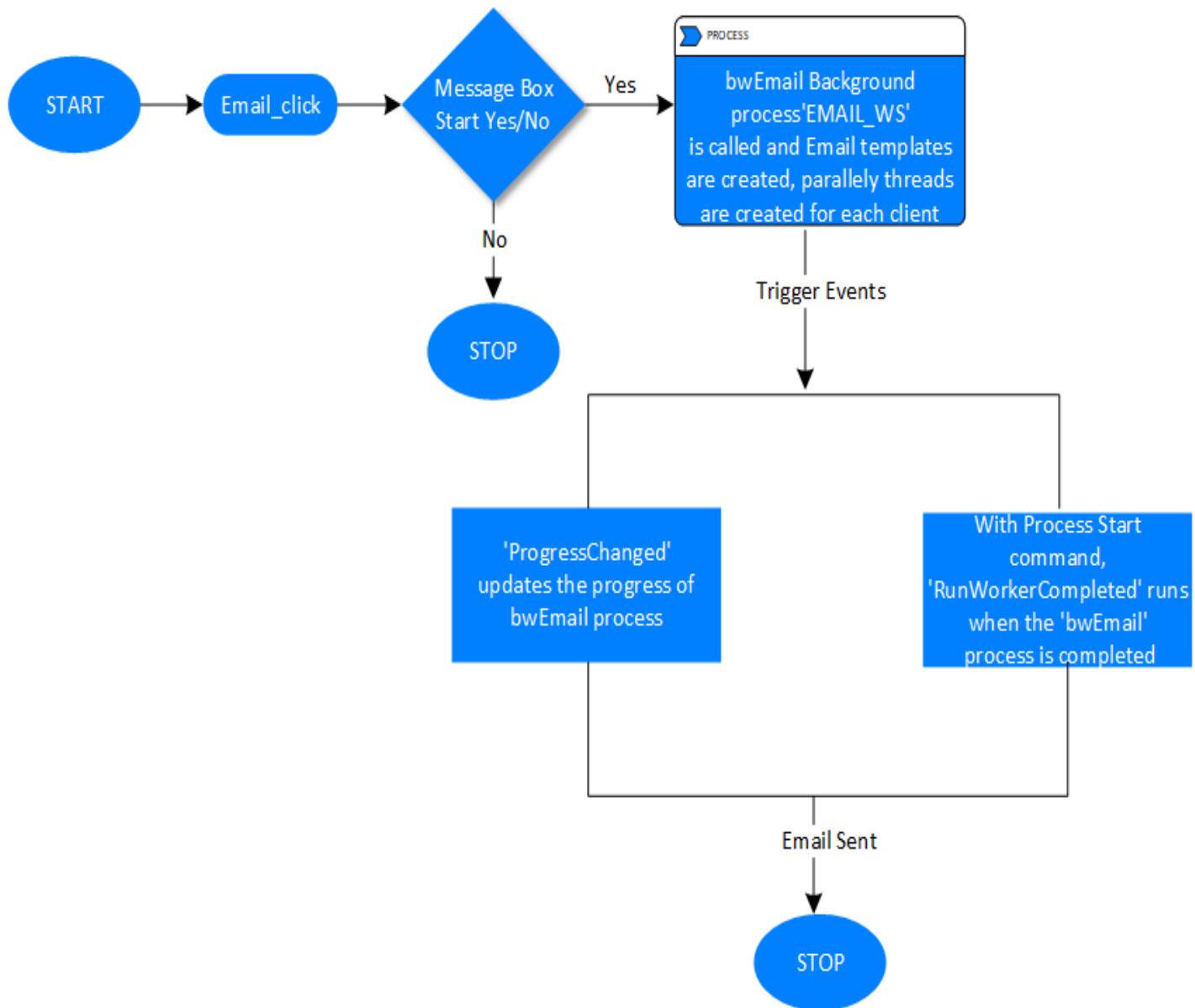


Figure 63: Balances Report 'Email Button Click' - Sequence Diagram

Sequence Diagram for Balances report 'Email Button click'-

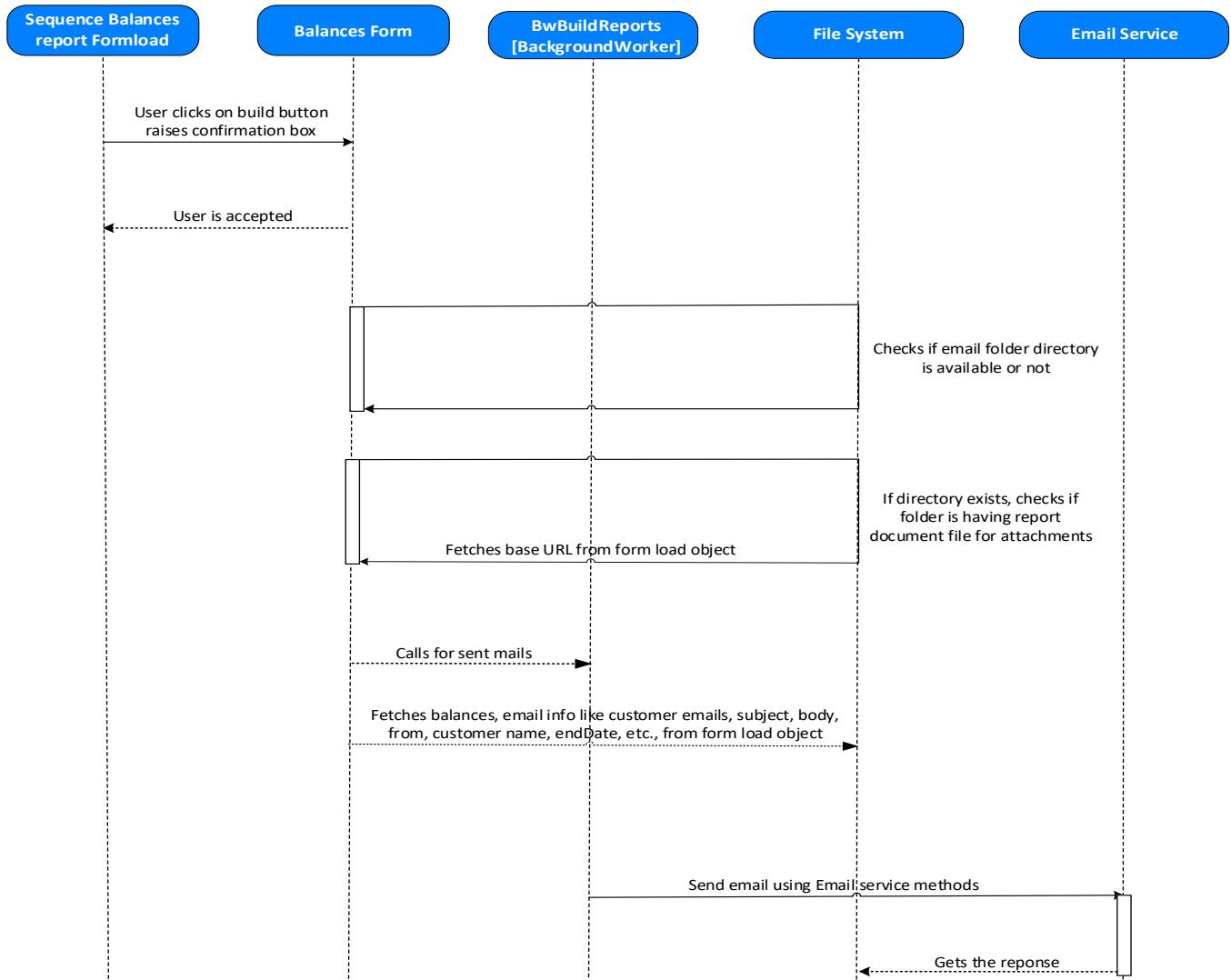


Figure 64: Balances Report 'Email Button Click' - Sequence Diagram

4.6.4 Balances Report – Preview button click

Upon clicking the “Preview” button, an authorized user can review the generated Balances file. The user has the option to select the client from a dropdown menu to preview the file. This functionality allows users to inspect the contents of the file before proceeding with the email sending process, providing an opportunity to address any doubts or make necessary changes.

It's important to note that the preview functionality is limited to one client at a time, ensuring a focused and controlled viewing experience. This restriction helps maintain clarity and precision during the preview process.

Process Flow Diagram for Balances Report 'Preview button click'-

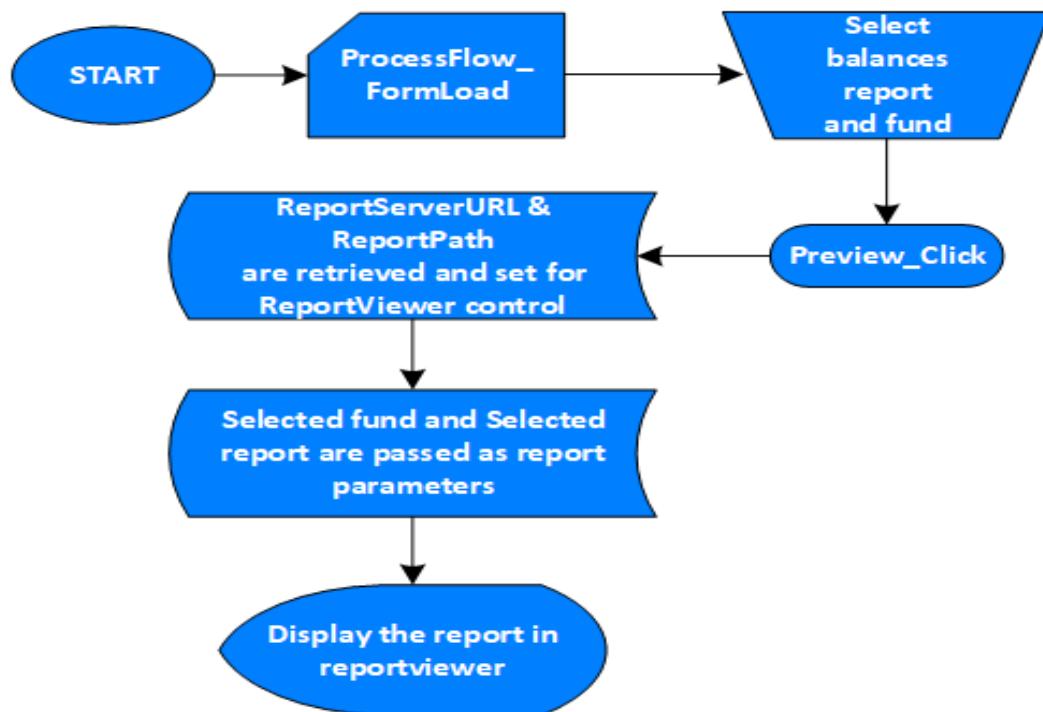


Figure 65: Balances Report 'Preview button click' - Process Flow Diagram

Sequence Diagram for Balances Report 'Preview button click'-

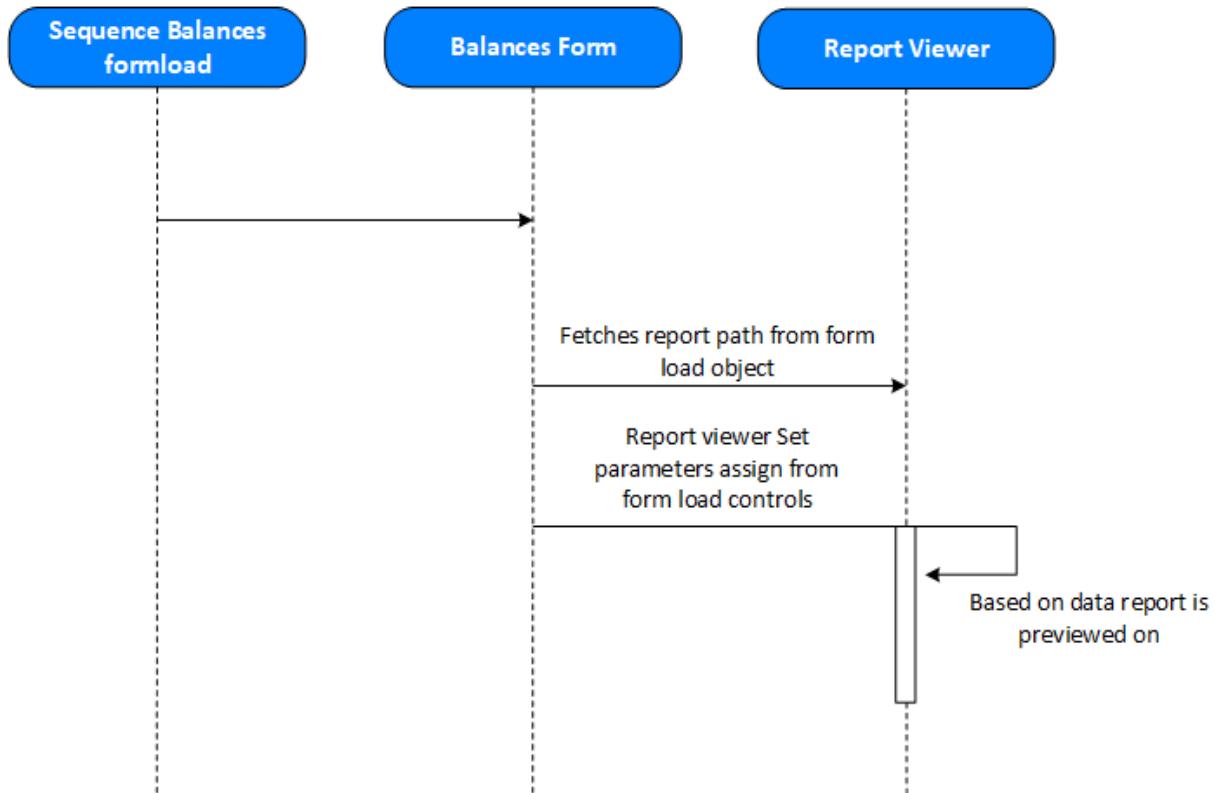


Figure 66: Balances Report 'Preview button click' - Sequence Diagram

4.6.5 Balances (FondoMutuoRentabilidad) Report – Build button click

The balances statement report (FondoMutuoRentabilidad.rdl) involves data to be provided for parameters and data fetched from SSRS to save the reports. All the data required for parameters is loaded in form load event. The table `tblFundSSRSUrl` (SAFI_DB) and column `SSRSUrl` provides the link to generate reports and PDF is saved to disk. Once the parameters are selected from the form controls and the user clicks Build button, `tblFundSSRSUrl` provides a URL for report based on selected fund & report date, this URL's placeholder parameters substitutes with values of user selected fund/funds, then an HTTP request to SSRS server for FondoMutuoRentabilidad report is sent and the response is saved as PDF report.

Process flow diagram for FondoMutuoRentabilidad report ‘Build button click’-

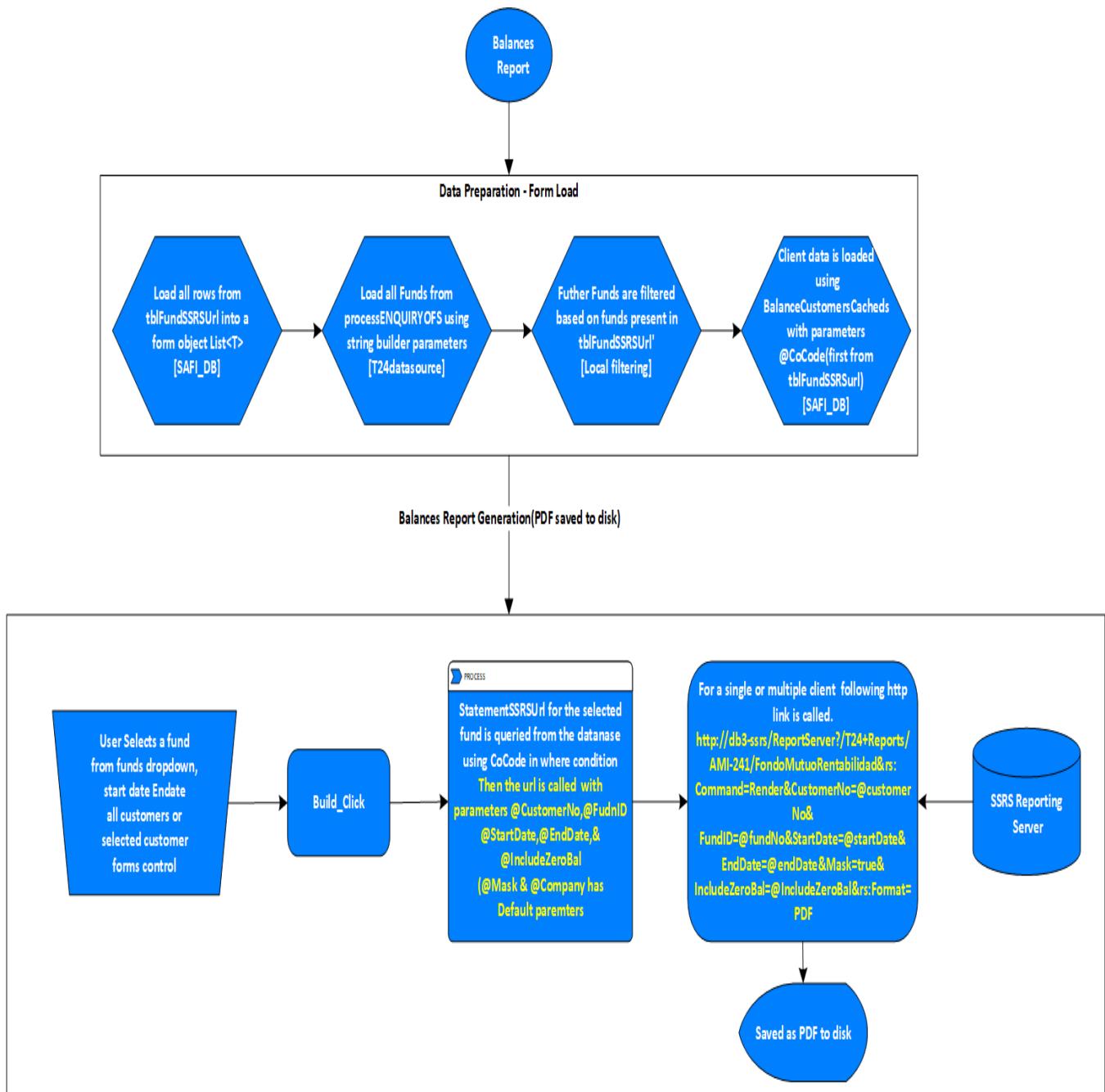


Figure 67: FondoMutuoRentabilidad ‘Build button click’ - Process Flow Diagram

4.6.6 Balances (FondoMutuoRentabilidad) Report – Preview button click

The viewing of the FondoMutuoRentabilidad report (FondoMutuoRentabilidad.rdl) when the user select the fund and report in the form controls and click preview, and the report will be visible in report viewing control This process is done by setting the path of the report by, concatenating three columns of tblFundSSRSUrl table namely ReportServer, ReportPath & ReportName. The parameters required to render the report are attached to the report control from the values in form control, after this the report is fetched from SSRS reporting server and render in C# windows application report viewing control.

Process flow diagram for FondoMutuoRentabilidad report ‘Preview button click’-

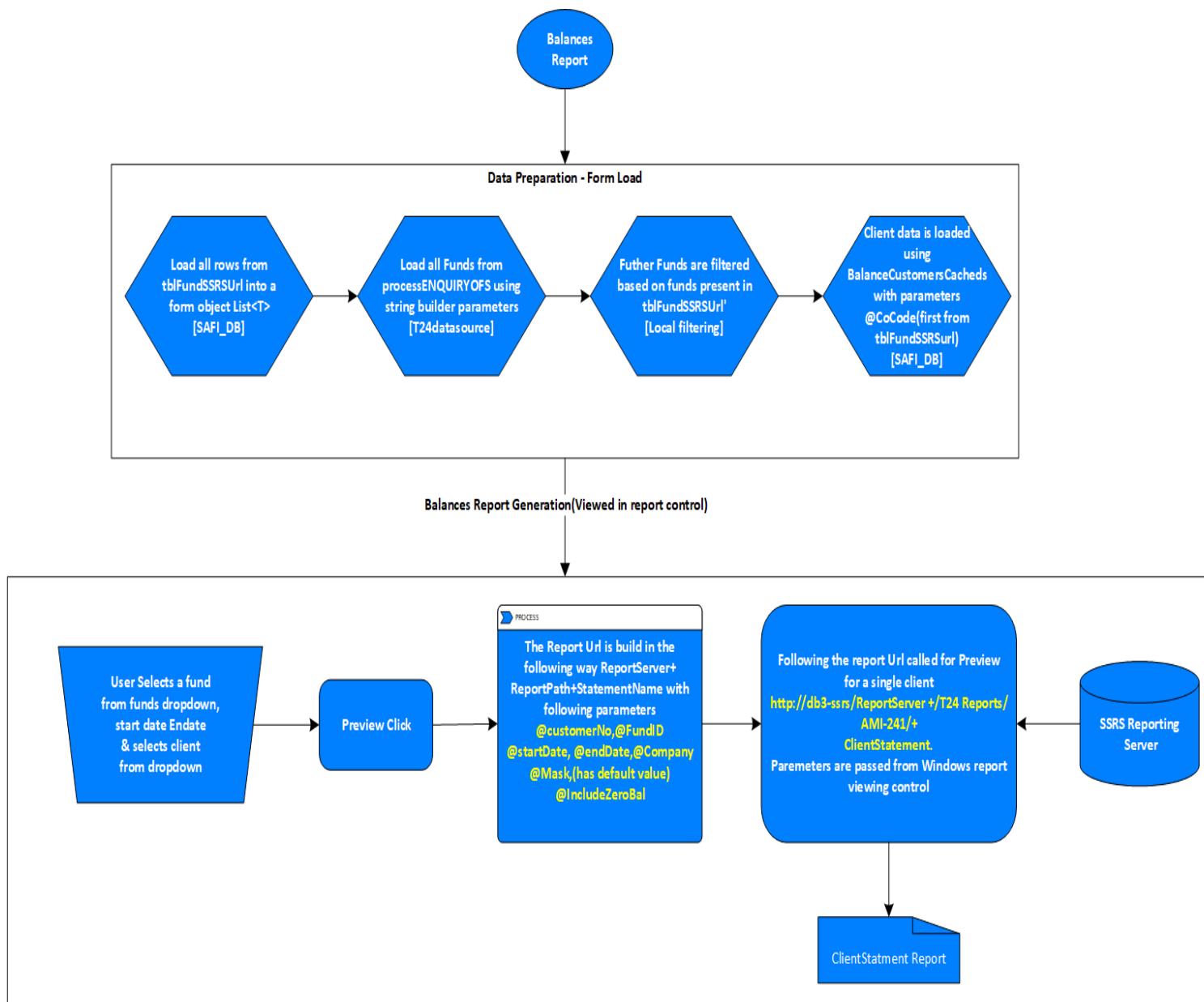


Figure 68: FondoMutuoRentabilidad ‘Preview button click’ - Process Flow Diagram

UML class level diagram details of the classes involved in balances form generation-

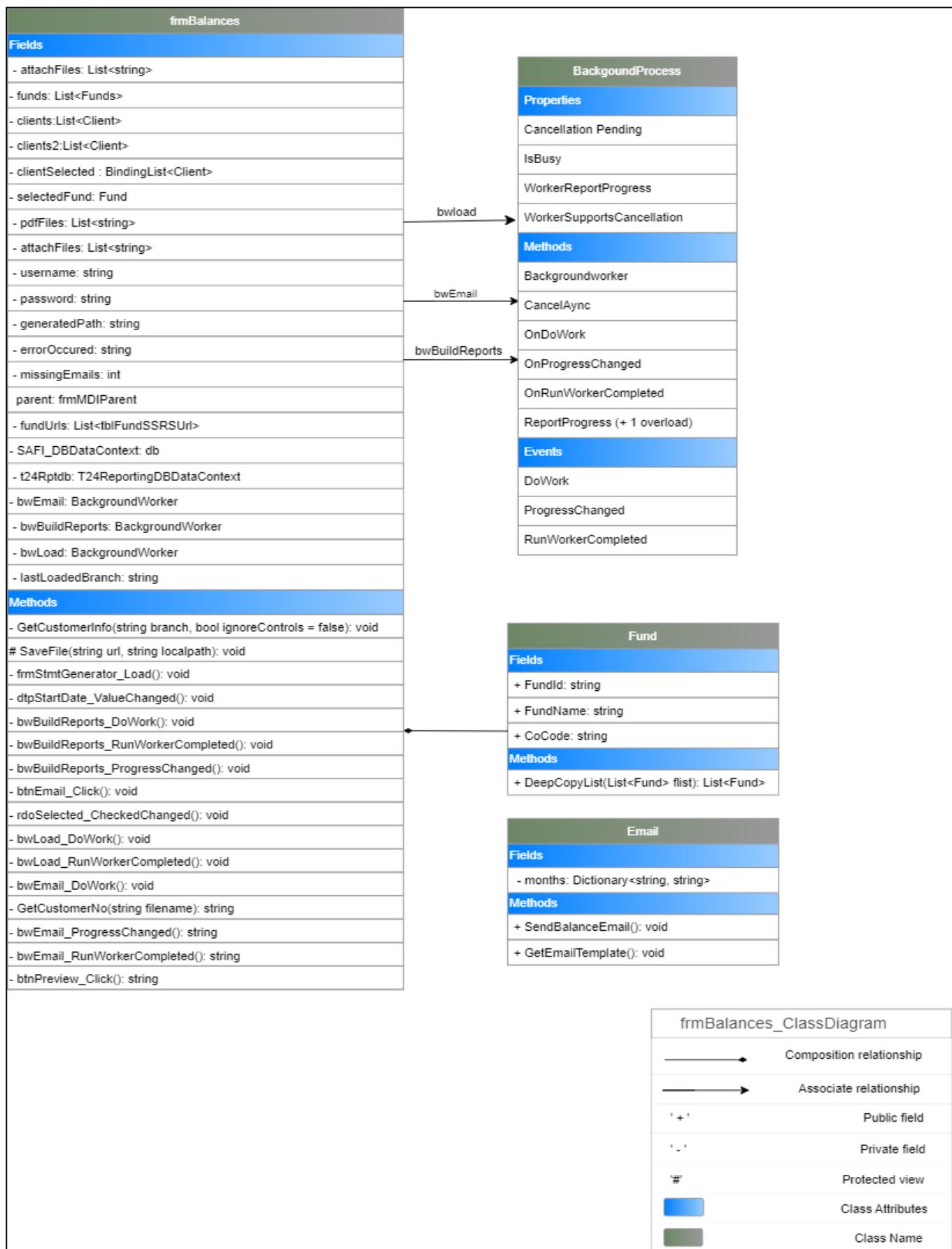


Figure 69: Balance form generation - UML Class diagram

ER Diagram for Balances Form

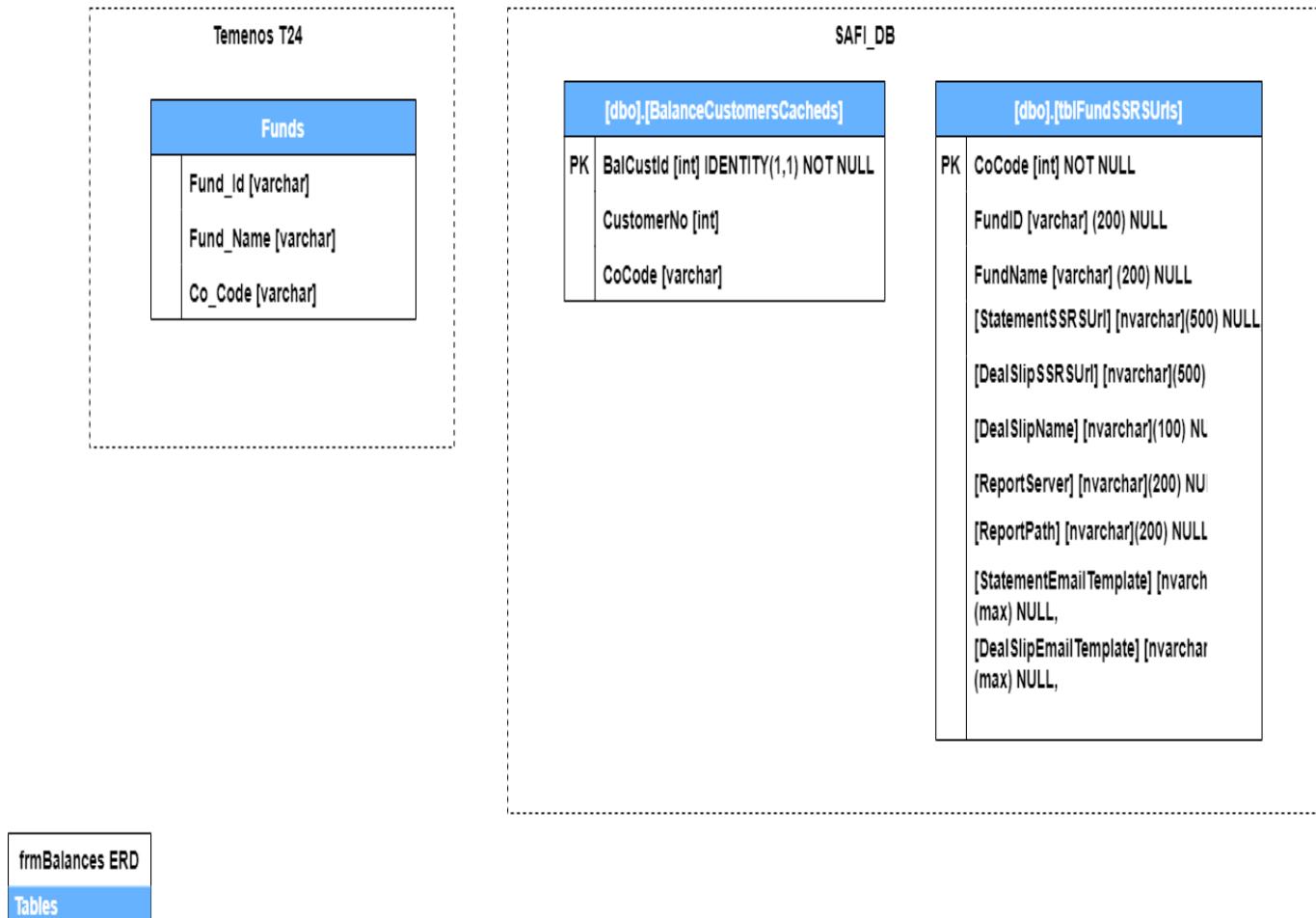


Figure 70: Balance form generation - ER Diagram

4.7 DR Reporting Application – AFP Statements

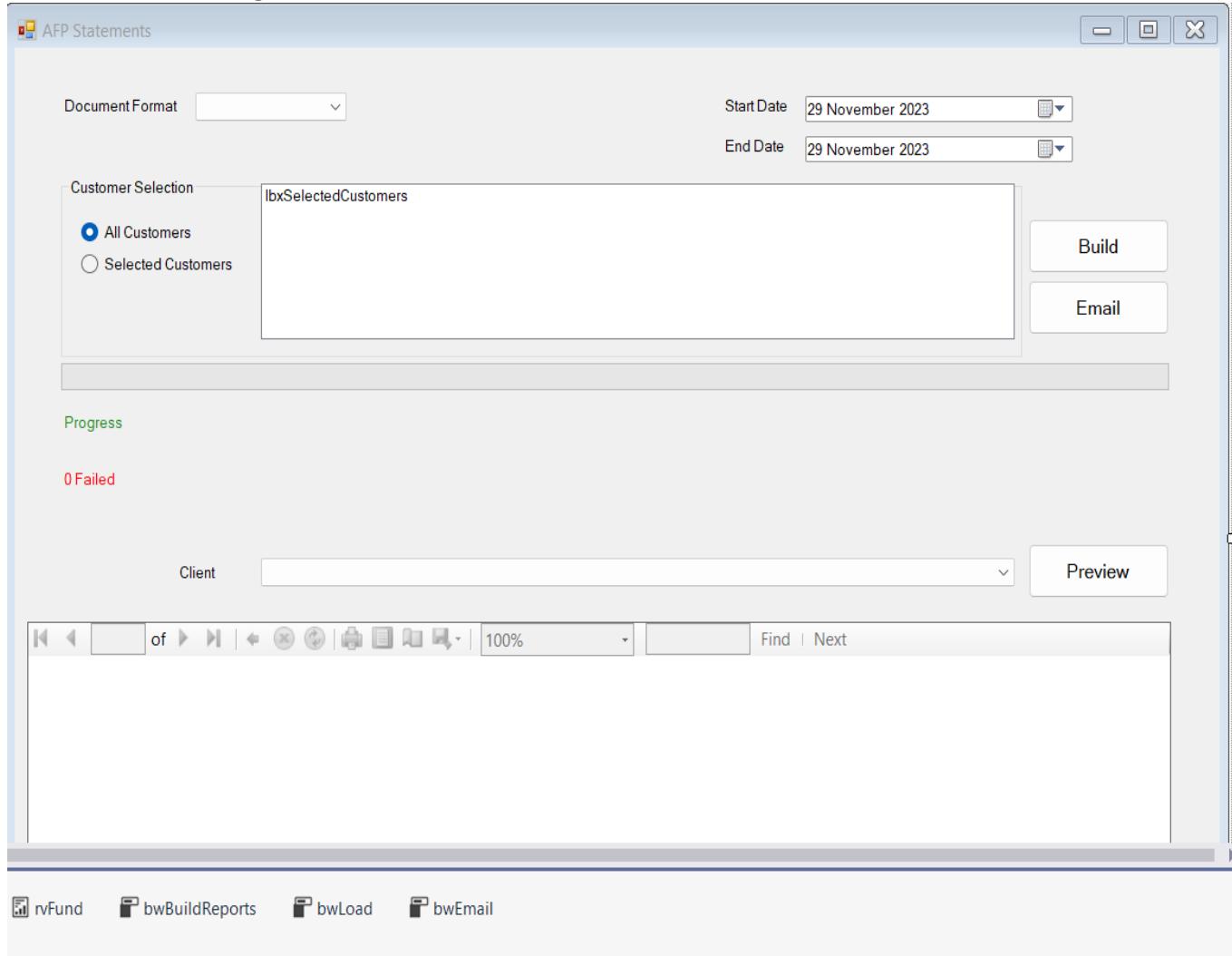
Code level details for AFP Statements

The AFP statement report can be generated by logging into DR application in the MDIform and selecting AFP Statements from the Reports Menu.

AFP Statement Generator does the following:

- When user clicks on preview after selecting client and default Fund, config data required for report is fetched from database and the returned report is viewed in the application form.
- Allows building of reports after getting config data from database and fetching the reports from the Report server using config data and saves the reports to file system(disk).
- Once the reports are generated to the file system in the following path
c:\DRStatements\AFPStatements\{endDate} The user validates the reports, then copies to the Email subfolder. The user may also put additional attachments (eg Quarterly Report) in the OtherAttachments subfolder inside Email folder. The user then clicks on the Email button to send an email to each client whose AFP statements is in the Email folder along with any attachments.

The AFP Statement generator form view Is below-



Document Format

Start Date

End Date

Customer Selection

All Customers
 Selected Customers

lbxSelectedCustomers

Build
Email

Progress

0 Failed

Client **Preview**

of | | Find | Next

Figure 71: DRReporting - AFP Statements Form view

The AFP Statement generation form has four action methods:

1. Form Load
2. Build button click
3. Email button click
4. Preview button click

4.7.1 AFP Statements – Form Load Action

During the form load process, the application creates the two ‘Linq2Sql’ DB contexts:

- SAFI_DB and
- T24ReportingDB

In the form load event of AFP statement generation, a background process is initiated which fetches the client data context entity and loads the data into client combo-box.

Process Flow Diagram for AFP Statements 'Form Load Action'-

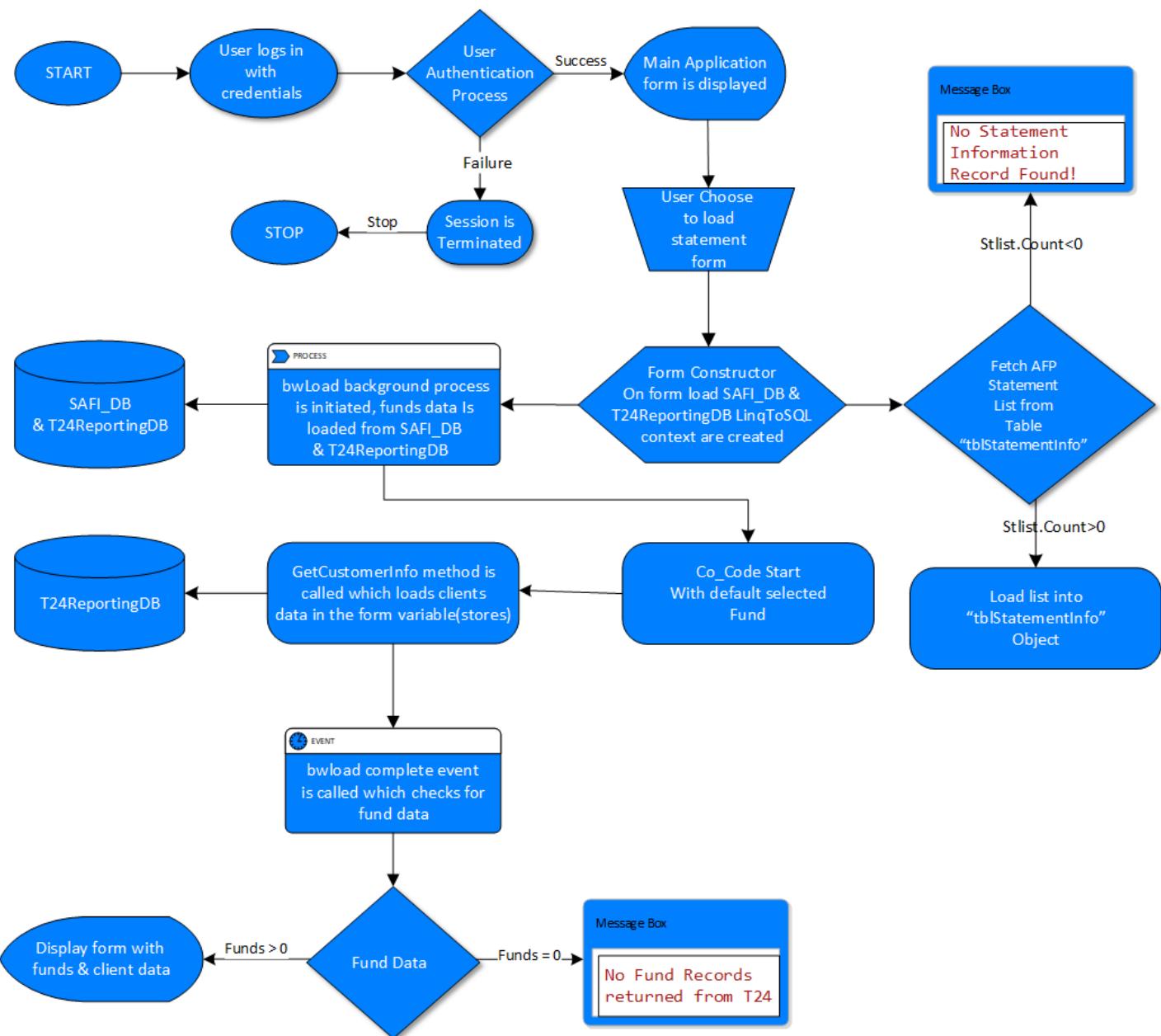


Figure 72: AFP Statements 'Form Load Action' – Process Flow Diagram

Sequence Diagram for AFP Statements 'Form Load Action'-

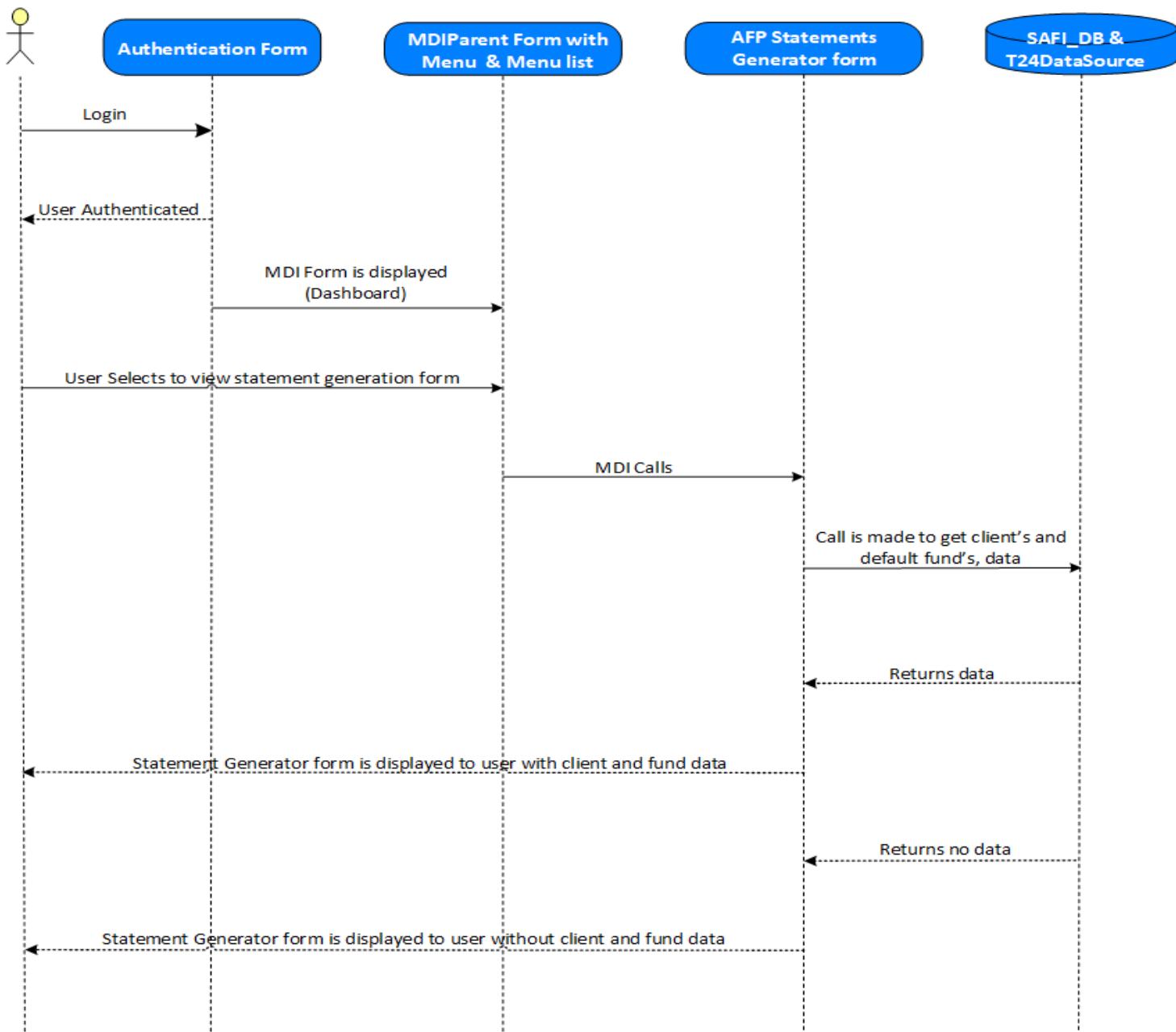


Figure 73: AFP Statements 'Form Load action' – Sequence Diagram

4.7.2 AFP Statements – Build button click

To start the build, the end user must select client, then set a 'Start date' and 'End date' from the date fields. After this when the user clicks on the build, he is greeted with a confirmation message box, on selecting 'Yes'. Creation of the folder with different path is done by checking if it does not exist already. The data stored during form load is used to get the URL for fetching data and same is stored in the local path created earlier, in the PDF format. Once the background process involved in building the PDF reports is completed, it raises a 'complete' event which displays the pdf built.

Process Flow Diagram for AFP Statements 'Build Button action'-

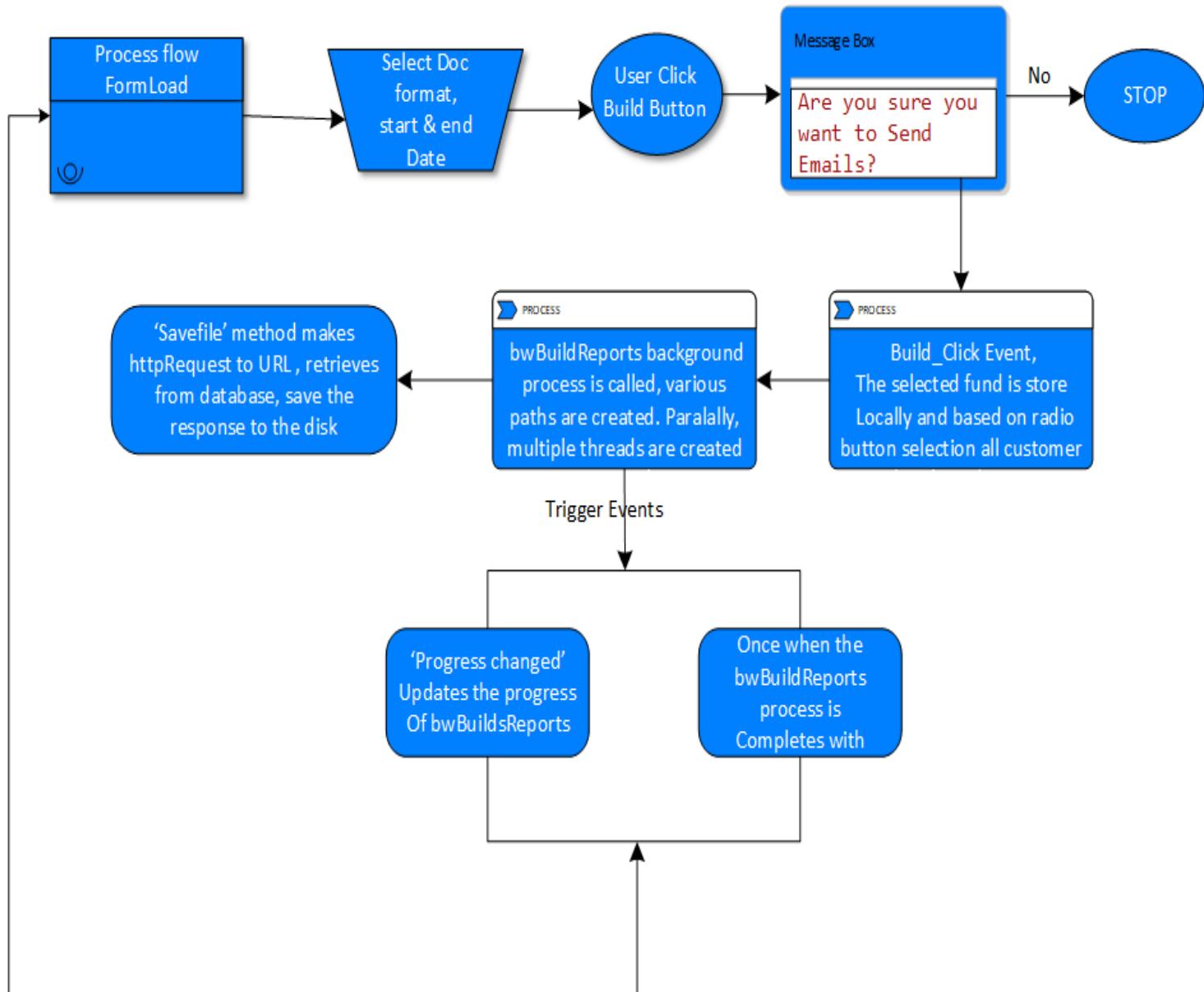


Figure 74: AFP Statements 'Build button' – Process Flow Diagram

Sequence Diagram for AFP Statements 'Build Button action'-

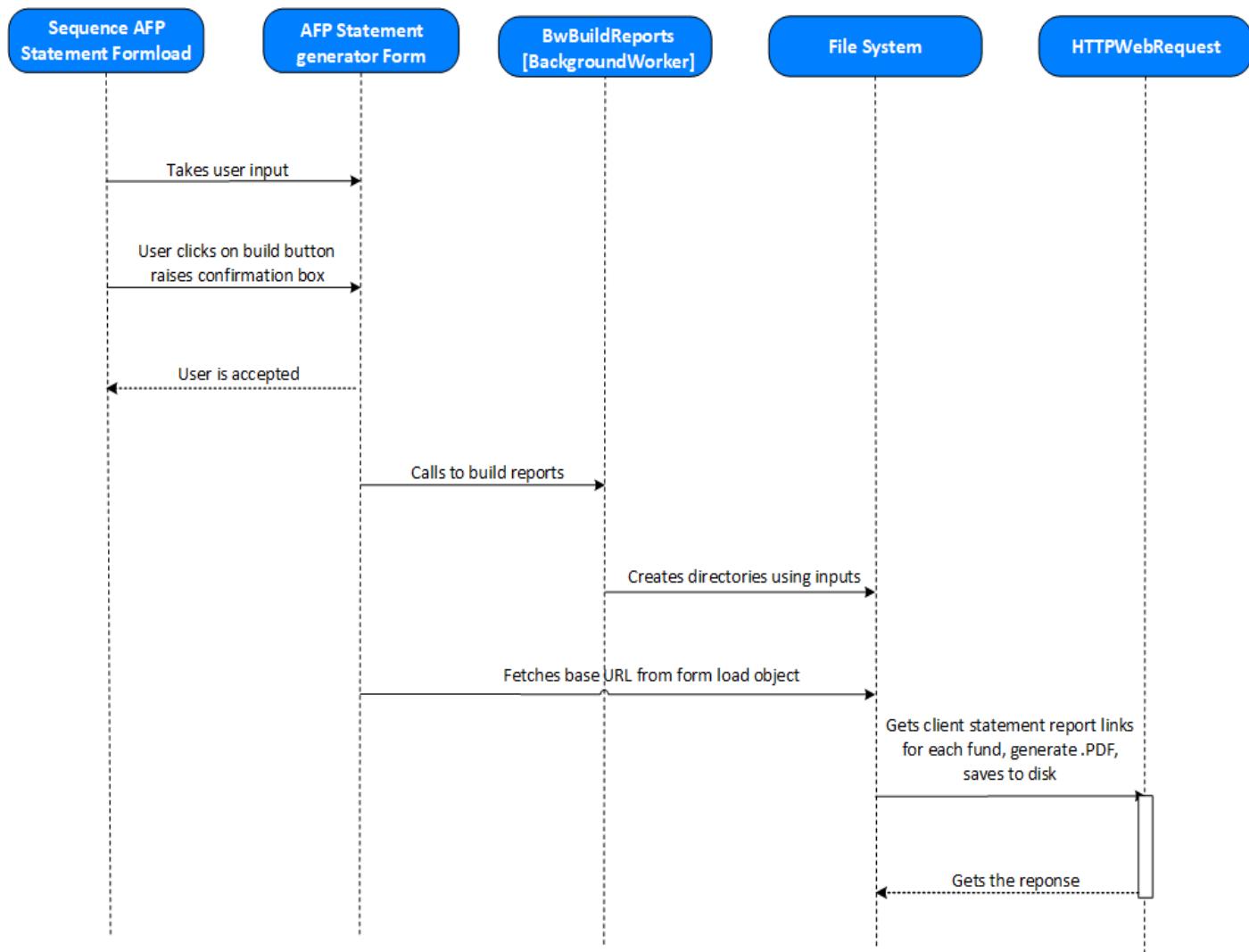


Figure 75: AFP Statements 'Build button' – Sequence Diagram

4.7.3 AFP Statements – Email button click.

When the user clicks on email and provides confirmation, the email process checks for existence of folder defined in the configuration file and it checks for the reports (pdf files) generated during the build process. Then a background process is spawned.

An email template from the database is used to create the email that is sent, after the tags have been replaced and email service is called to send email to respective clients.

The following Tags are replaced with data from the application before email Is sent.

Tags	Data from application
Replaces in the body & subject of the template.	
<Nombre>	Full Name
<Nombre>	If fullname=null then Cliente
<day>	endDate.ToString("dd")
<month>	months[endDate.ToString("MM")]
<year>	endDate.ToString("yyyy")
Replaces in the body of the template.	
<section1>	el Estado de Cuenta de su inversión
section2>	""

Table: Statement reports – Tags

Process Flow Diagram for AFP Statements 'Email Button click'-

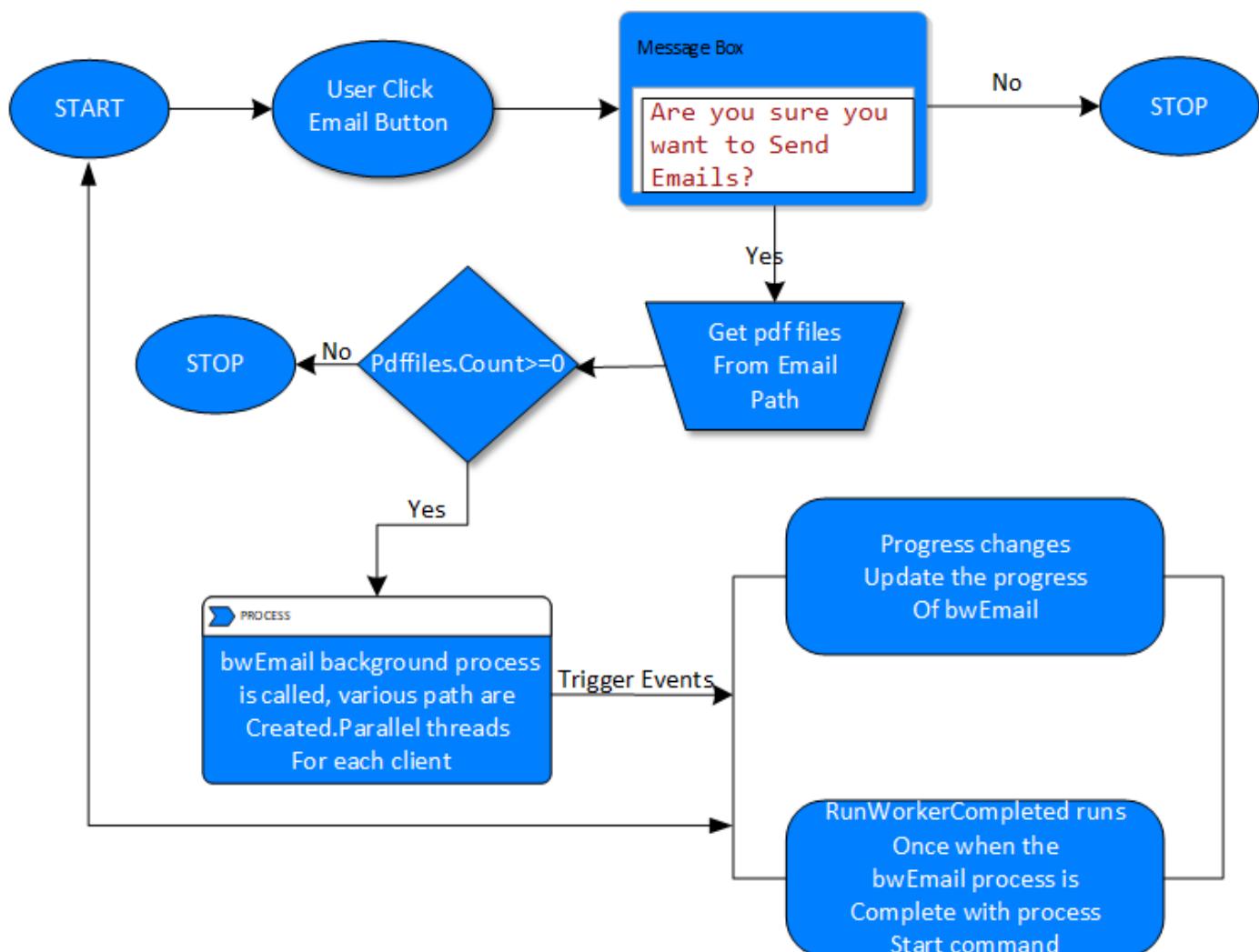


Figure78: AFP Statements 'Email Button' – Process Flow Diagram

Sequence Diagram for AFP Statements 'Email Button Click'-

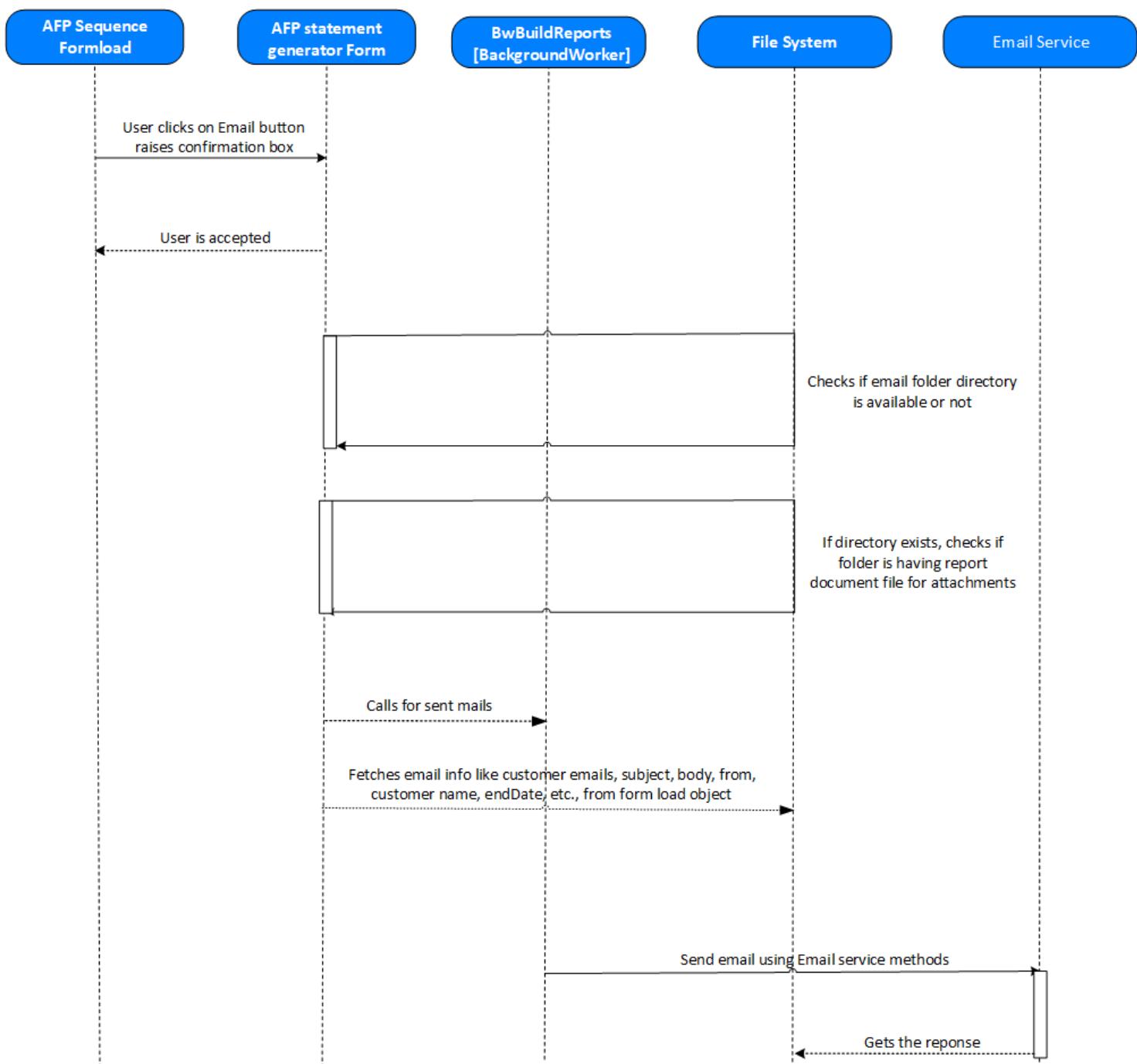


Figure 76: AFP Statements 'Email Button Click' - Sequence Diagram

4.7.4 AFP Statements – Preview button click

Here In the Report viewer, the report will be displayed after selecting Document format and respective report.

Process flow diagram for AFP Statements ‘Preview button click’-

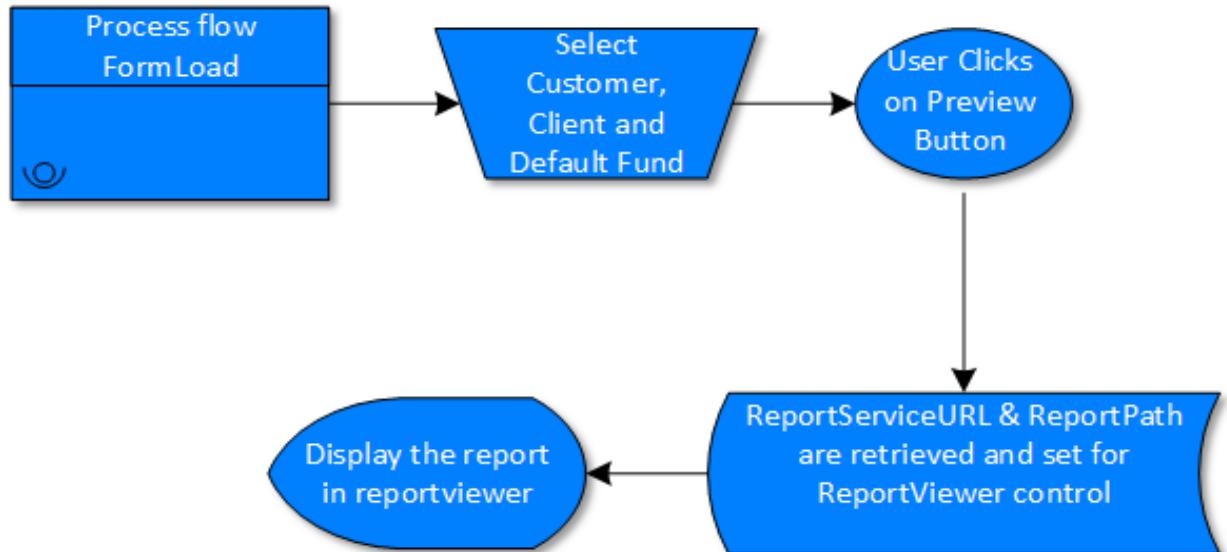


Figure 65: AFP Statements ‘Preview button’ - Process Flow diagram

Sequence Diagram for AFP Statements ‘Preview button click’-

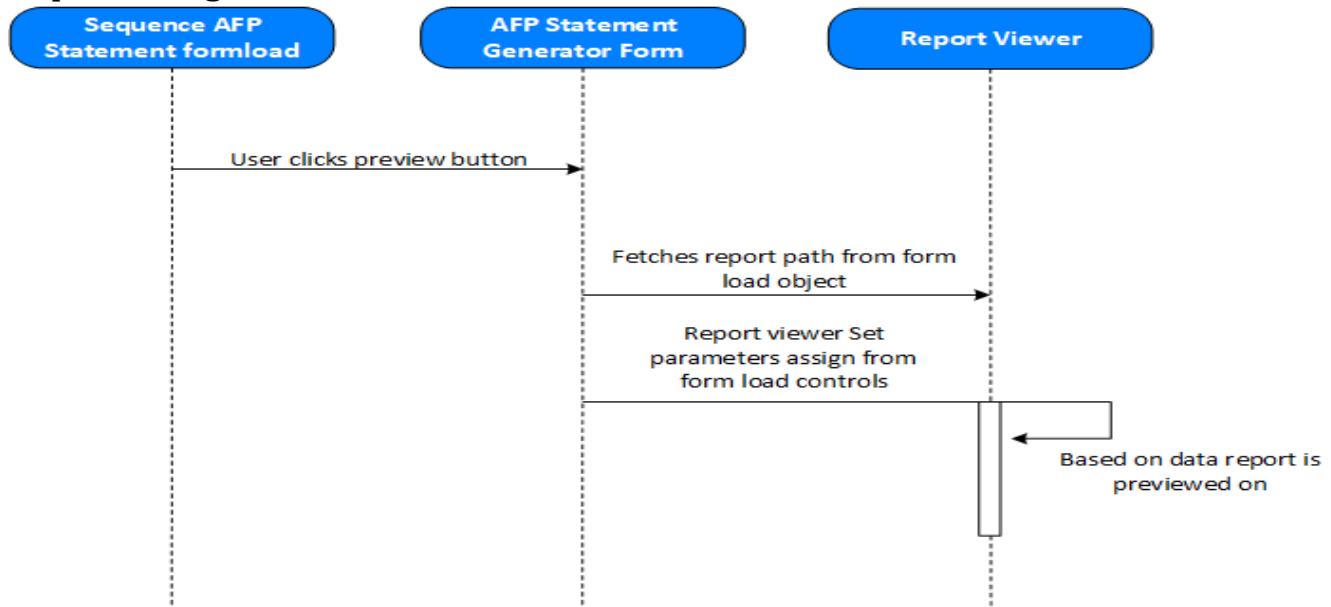


Figure 66: AFP Statements ‘Preview button’ – Sequence Diagram

4.7.5 AFP Statement Report – Build button click

The AFP statement report (AFPSatement.rdl) involves data to be provided for input parameters and data fetched from SSRS to save the reports. All the data required for input parameters are loaded in form load event. The table ‘tblStatementInfo (SAFI_DB)’ and column ‘SSRSUrl’ provides the link to generate reports in the form of PDF and is saved to the disk. Once the parameters are selected from the form controls, the user clicks on ‘Build’ button. The URL is retrieved from table ‘tblStatementInfo’ for the reports based on Co_Code of the selected fund. This URL’s placeholder parameters, substitutes with values of user selected ‘client/clients’, then HTTP request to SSRS server for AFP Statement report is sent and the response is saved as PDF report.

Process flow diagram for AFP Statement report ‘Build button’ -

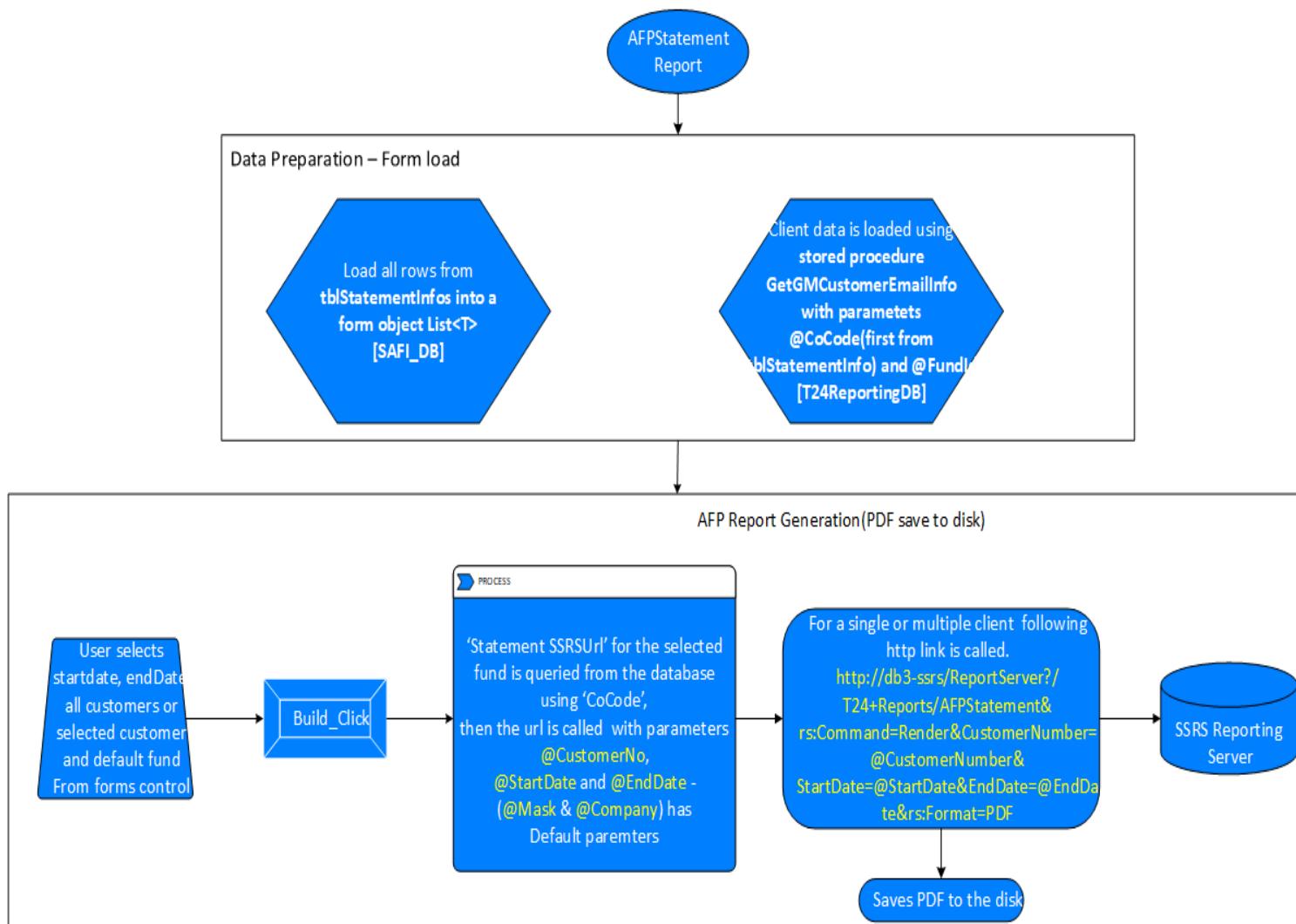


Figure 81: AFP Statement Report ‘Build button click’- Process Flow Diagram

4.7.6 AFP Statement Report – Preview button click

The viewing of the AFP statement report (AFPStatement.rdl) involves user selecting the Default fund and client in the form controls and then clicks 'preview'. The report will be visible in the report viewer control. This process is done by setting the path of the report by concatenating three columns from the table 'tblStatementInfo' namely 'ReportServer', 'ReportPath' and 'StatementName'. The parameters required to render the report are attached to the report control from the values in form controls, after this the report is fetched from SSRS reporting server and rendered in C# windows application report viewer control.

Process Flow Diagram for AFP Statement report 'Preview button click'-

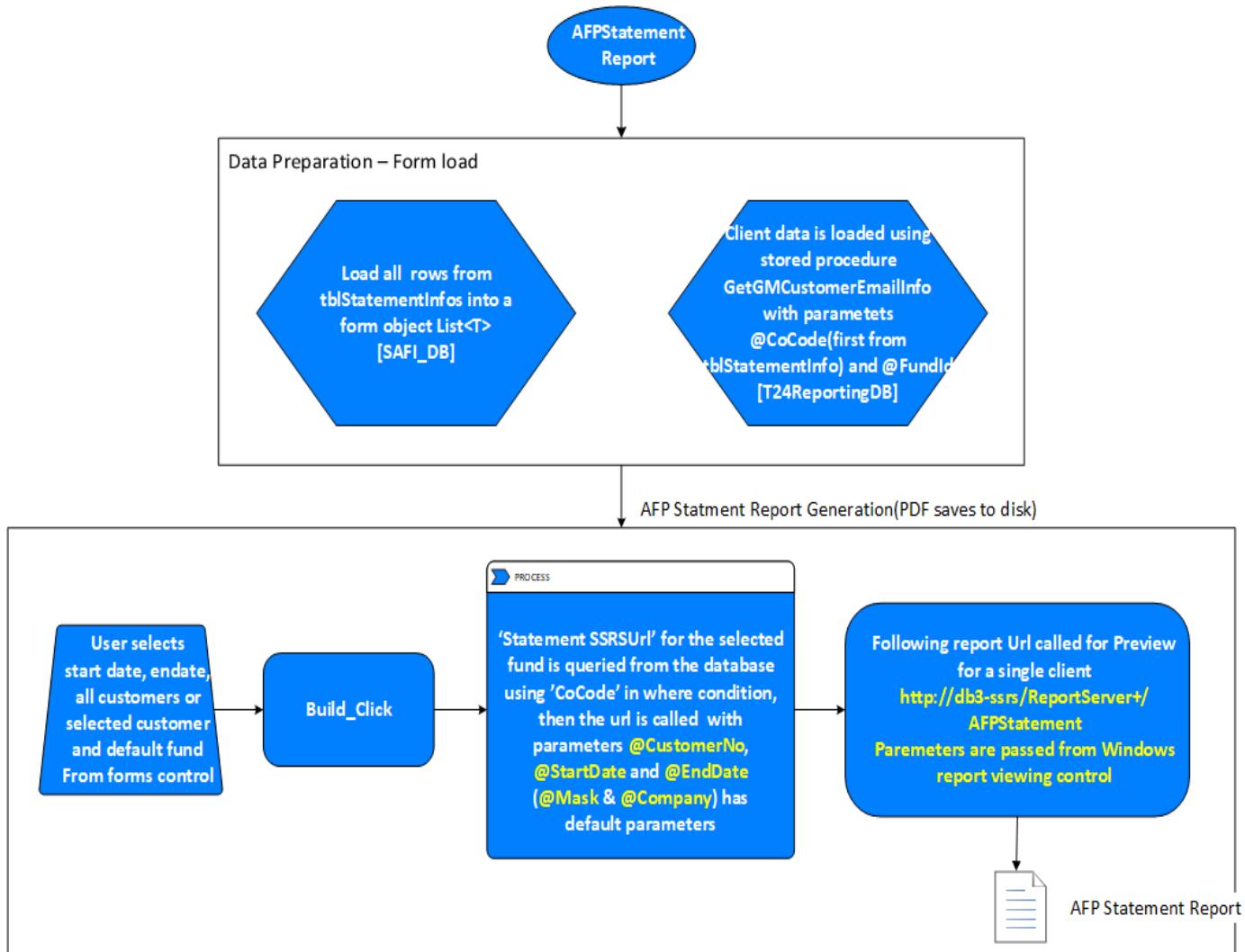


Figure 80: AFPStatement Report File 'Preview button'- Process flow diagram

UML class level diagram details of the classes involved in statement generation-

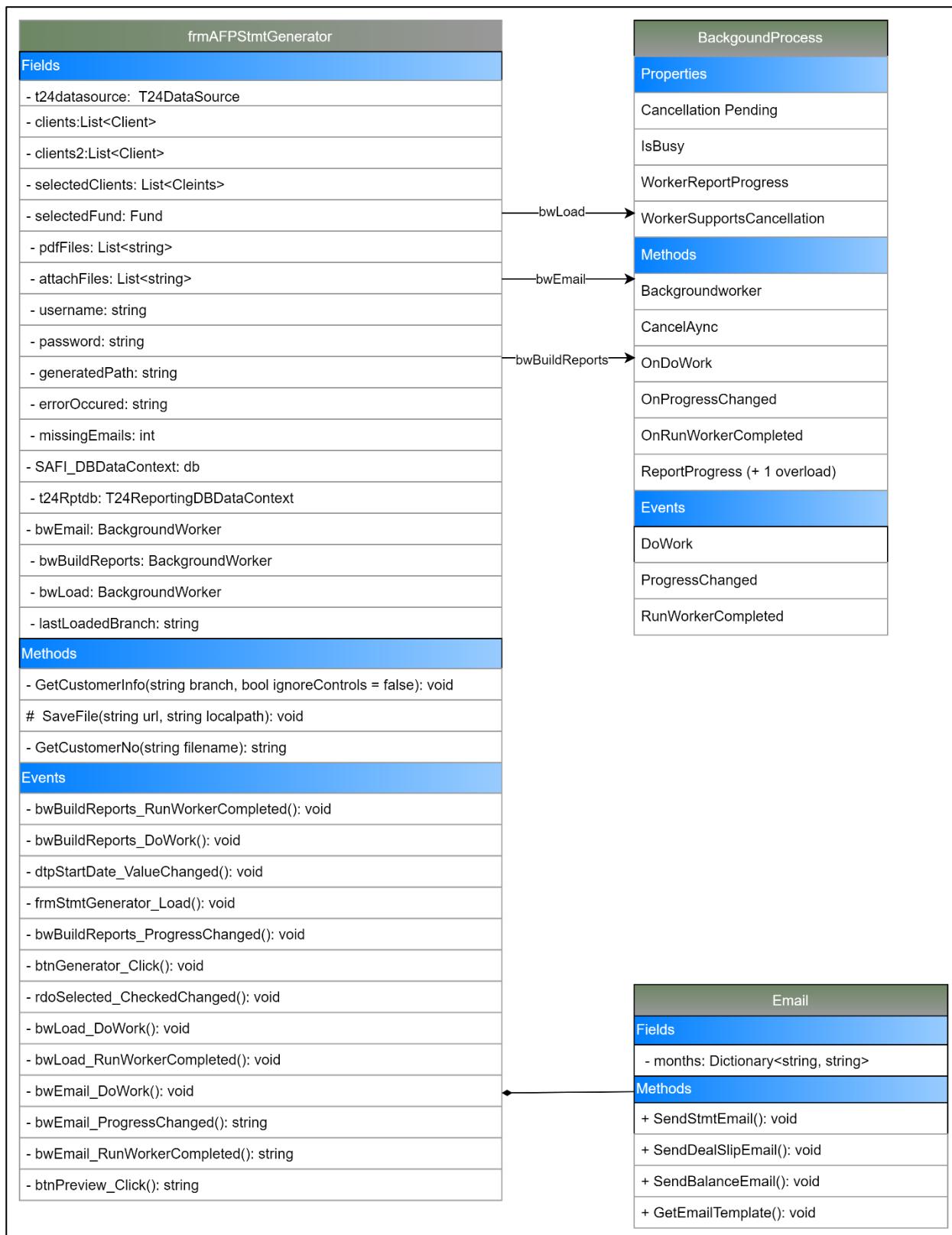


Figure 77:AFP Statements Generator – UML Class Diagram

ER Diagram for AFP Statements-

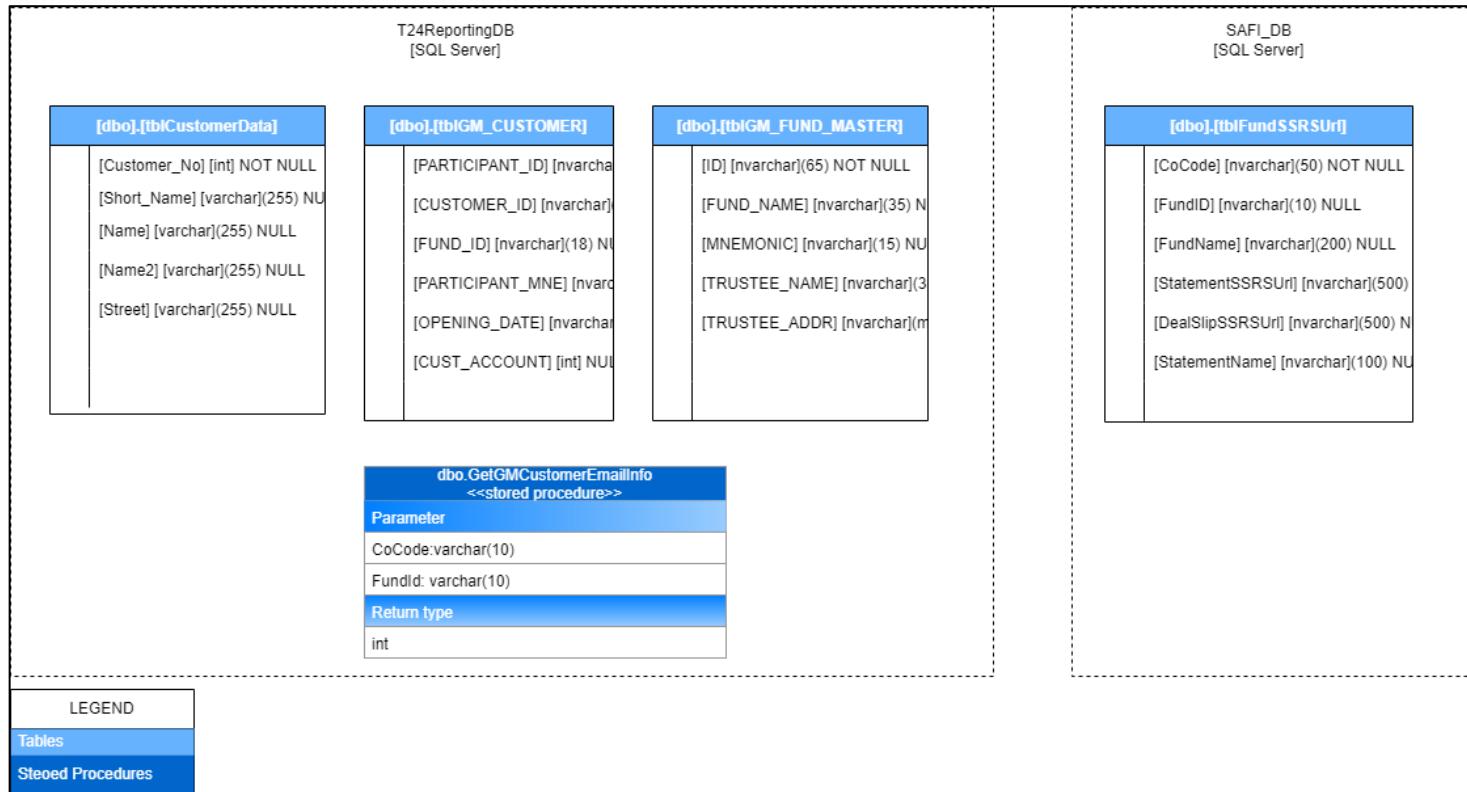


Figure 78: AFP Statements - ER Diagram

4.8 DR Reporting Application – REIT Upload Tool

Code level details for REIT Upload Tool

The frmREIT is one of the C4LEVEL4 (Code) Level, which is a drill down of the ‘DR Reporting App C# .Net Components’ and the below diagrams depicts the exact flow of frmREIT screen, validations and conditions that are being performed at the code level with the functionality.

Application View of REIT Upload Tool-

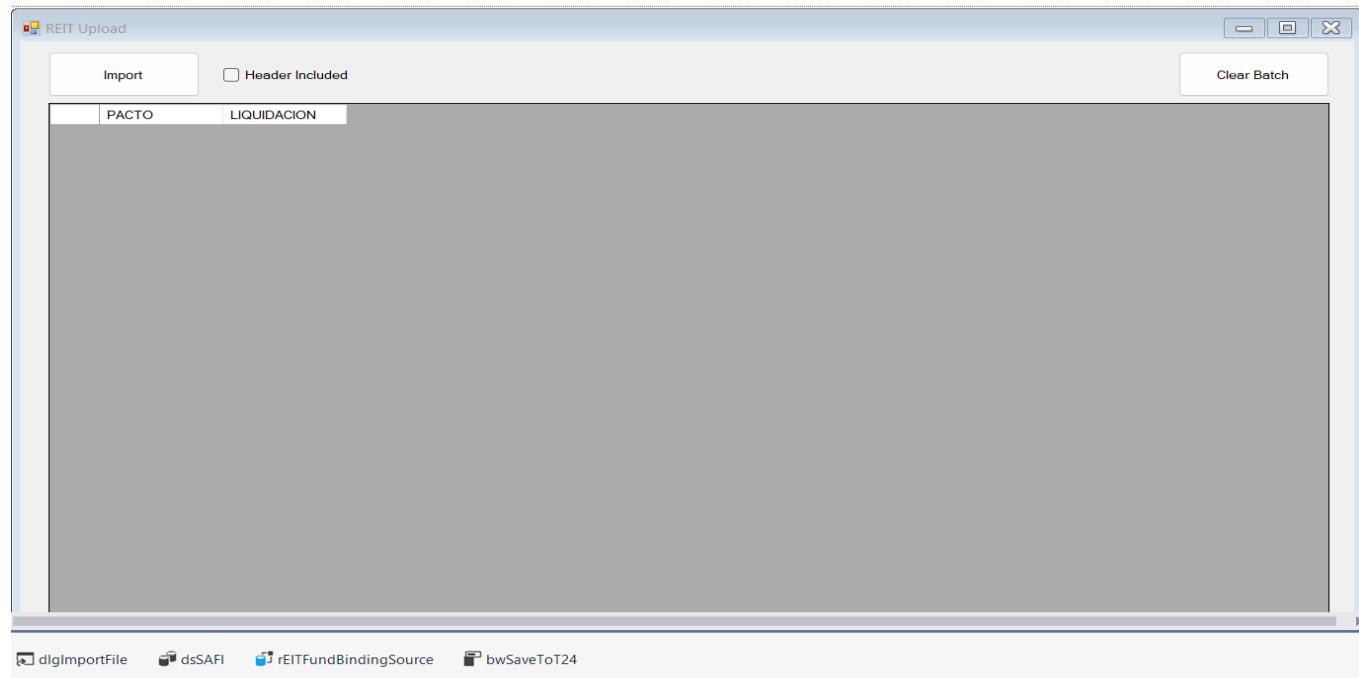


Figure 79: DRReporting REIT Windows - Form view

The REIT Upload tool form has four action methods:

1. Load Form
2. Import button click
3. Send File button click.
4. Clear Batch button click.

4.8.1 REIT Upload Tool – Load Form

Once the form gets loaded It just sets the Form Text to 'SAFI Utility System – User' with the User Domain Name and Environment UserName

Process flow diagram for REIT Upload tool – 'Load Form'

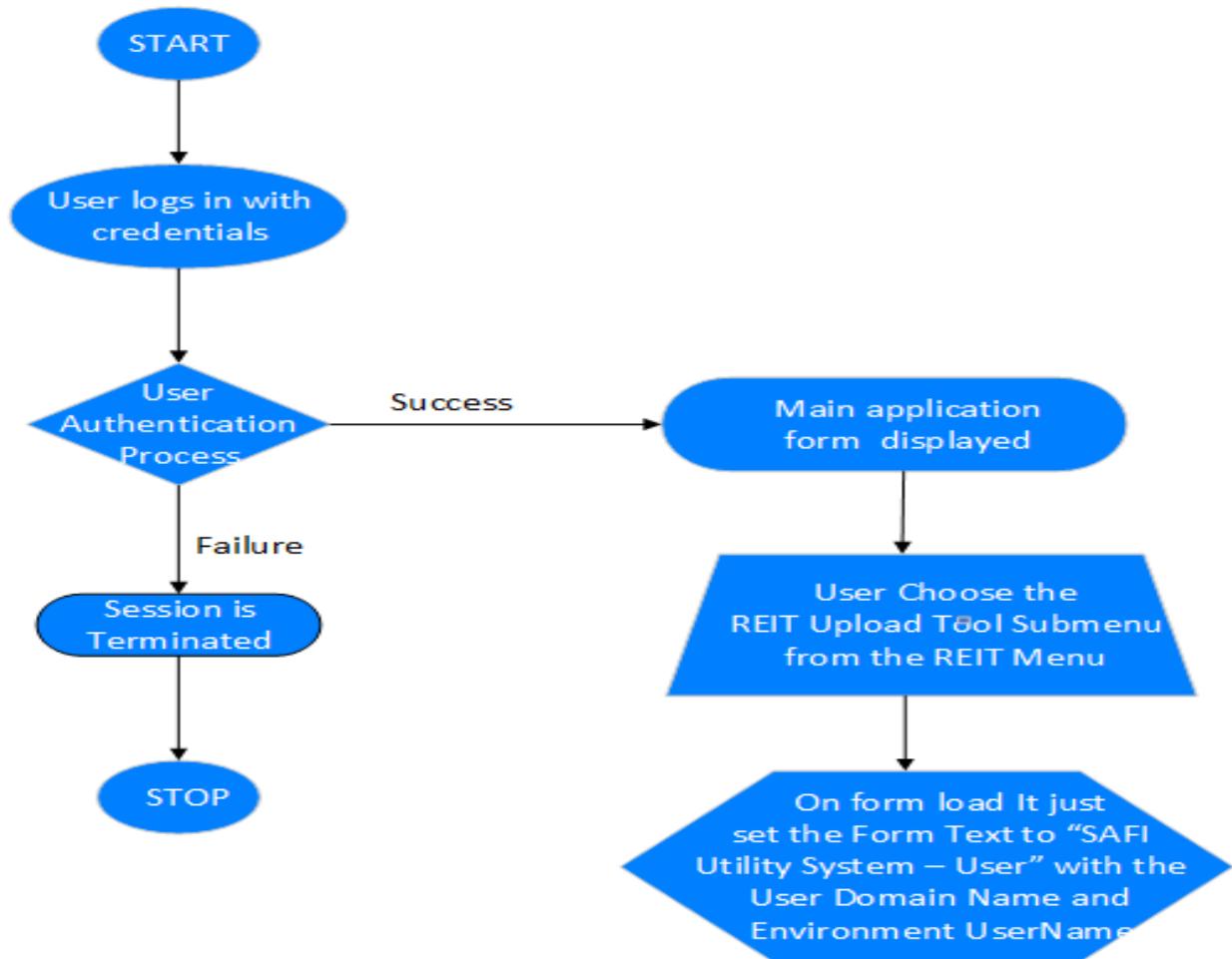


Figure 80: REIT Upload Tool 'Form load' – Process flow diagram

Sequence Diagram for REIT Upload Tool 'Form Load'

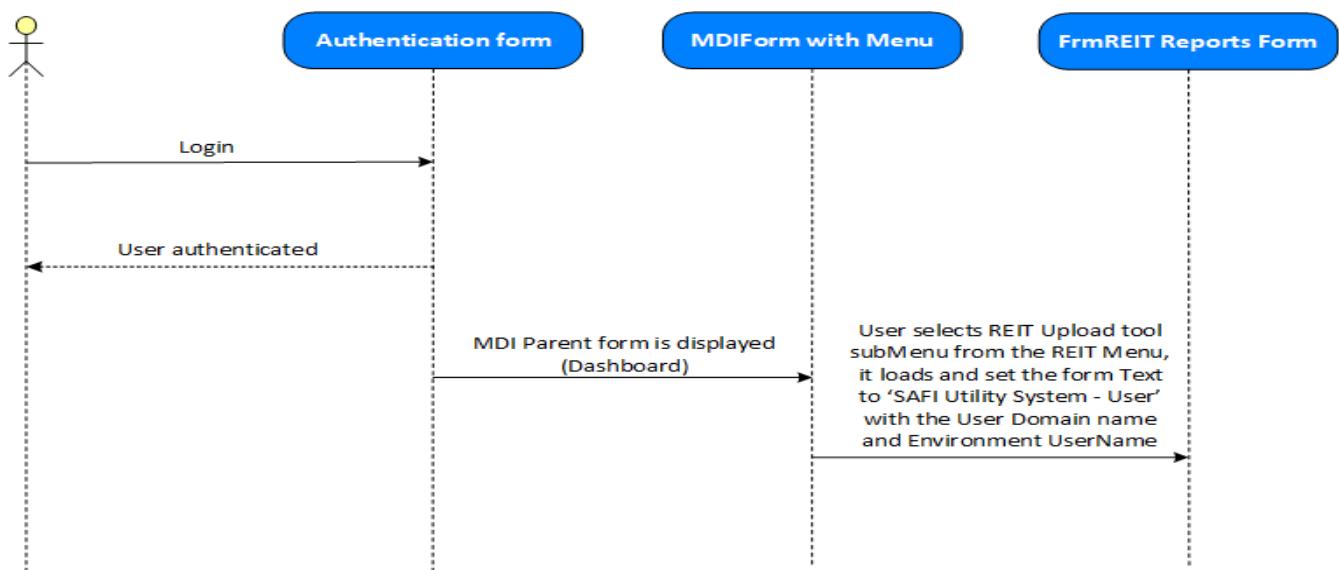


Figure 81: REIT Upload tool – 'Form Load' – Sequence flow diagram

4.8.2 REIT Upload Tool- Import Button Click

Once the button ‘btnImport’ is clicked, it checks whether the username and password are given correctly and then proceeds with the process flow. Here, It calls a ‘ProcessQuery’ method which opens up a Dialog box and it can be saved as a .CSV file or a .XLSX or .XLS file. The information retrieved from the method is saved in any one of the formats mentioned above and this file is parsed and the information is saved into a temporary fundRow Datatable. Then ProcessQuery method is called again with the parameter “UnitsEnquiry” to get the Units information, then assigns the row Count to a “sum” variable, and stores it in a temporary Datatable. It also assigns the first column value, second column value from the Datatable to a variable called “prevNumOfUnits” and “totalFundUnits” respectively.

Now it checks for the following conditions: -

- If sum < prevNumOfUnits
It assigns a Boolean variable called “Valid” to false and a message box is thrown saying “The total amount of units are less than the previous registration of customer participation.” As a warning.
- If (sum + prevNumOfUnits) > totalFundUnits
It assigns a Boolean variable called “Valid” to false and a message box is thrown saying “All units reported exceed the maximum amount of units for the entire fund.” As a warning.
- If Valid = true
btnSend will get Enabled else btnSend will be disabled.

Process flow diagram for REIT Upload Tool ‘Import Button Click’-

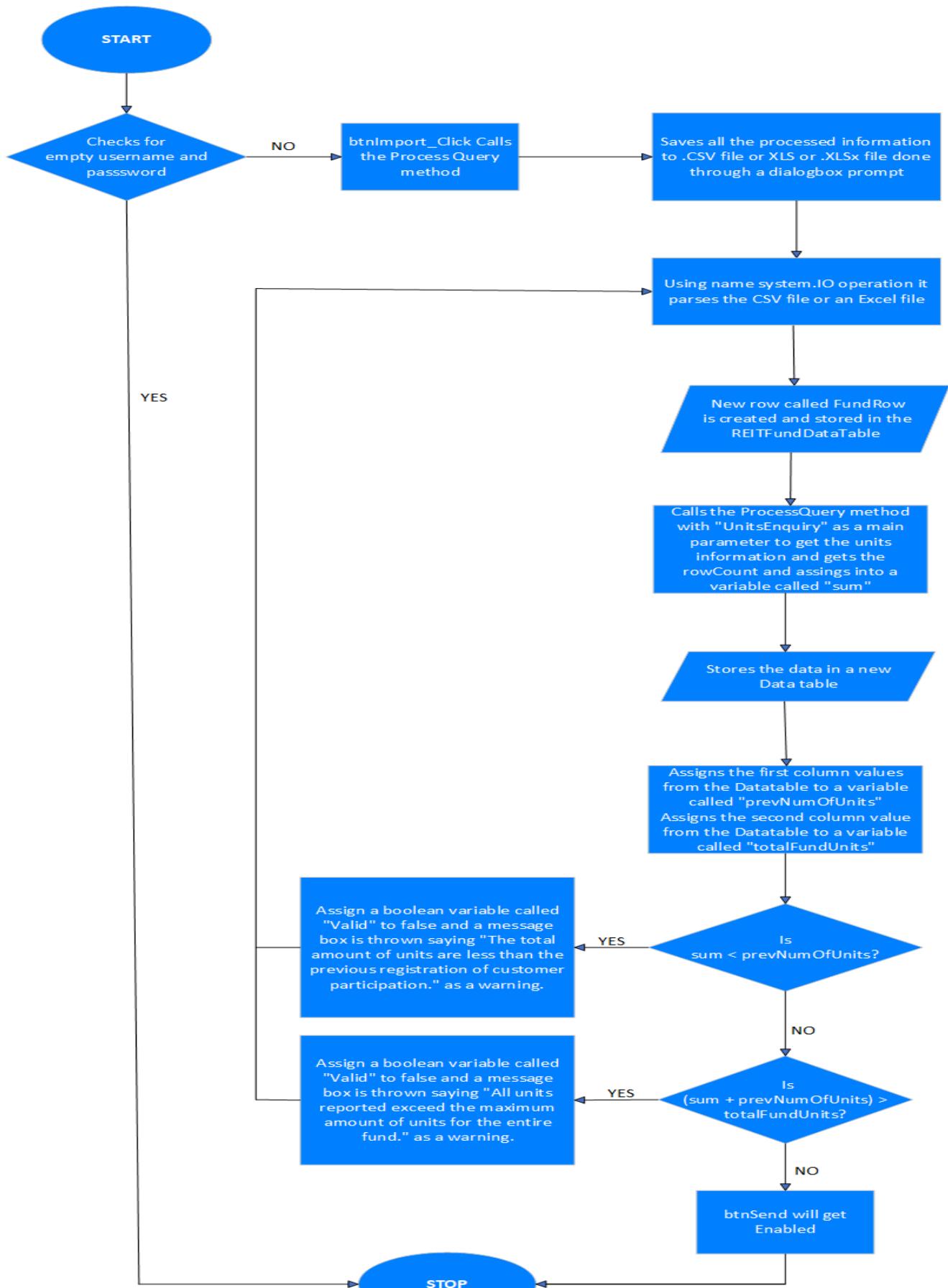


Figure 82: REIT Upload Tool 'Import Button Click' - Process Flow Diagram

Sequence Diagram for REIT Upload Tool 'Import Button Click'-

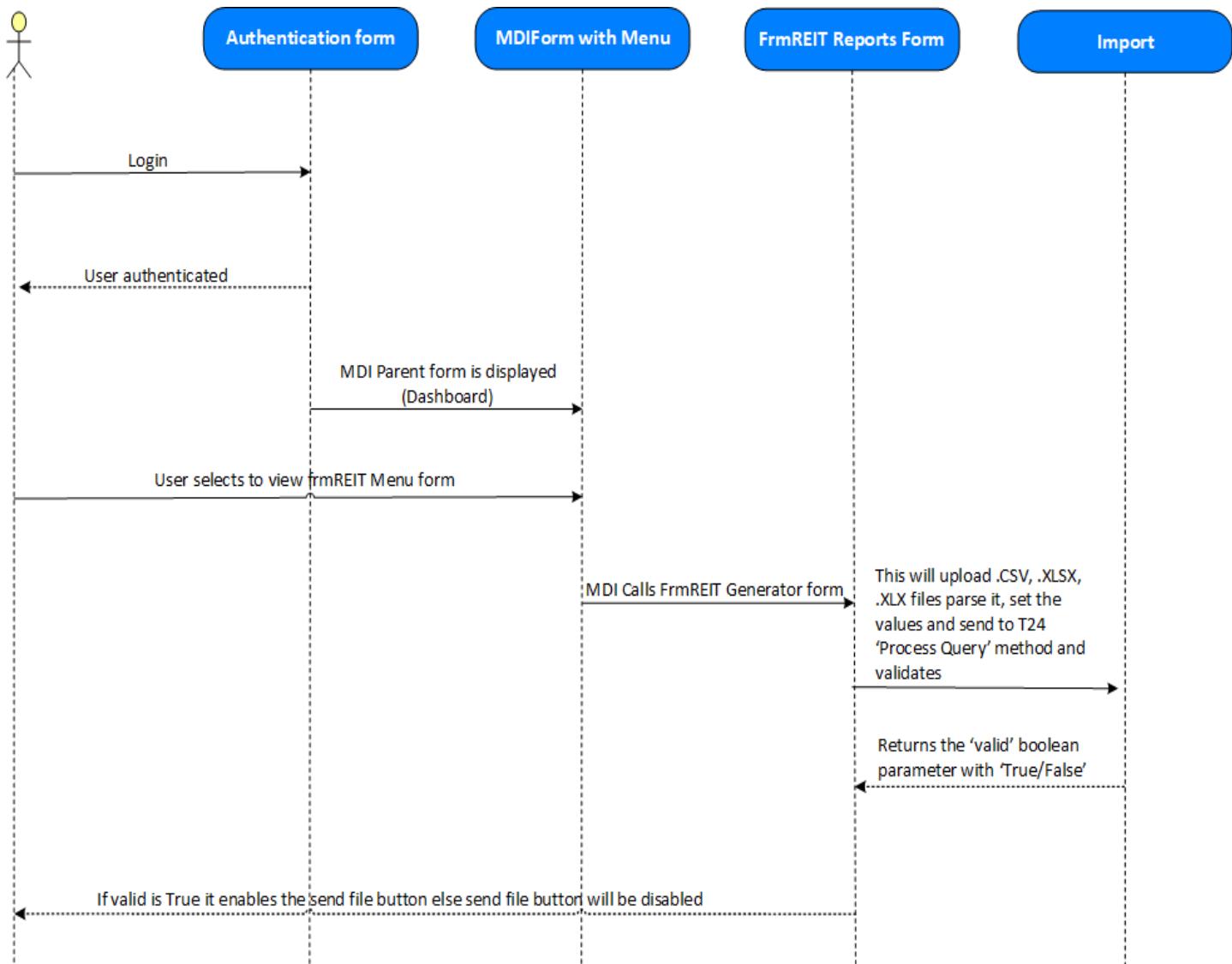


Figure 83: REIT Upload Tool 'Import Button Click' – Sequence flow diagram

4.8.3 REIT – Send File button click

After the Import click button process is completed, it checks whether the import process button succeeds after validation. The Send File button gets enabled or disabled after successful or failed validation. Once

it has succeeded, the user clicks on the Send File button, data is saved to Temenos T24 database. If it fails, it continues the same process to see if it gets validated and we are able to save it to T24 database

Process Flow Diagram for REIT 'Send File button Click'-

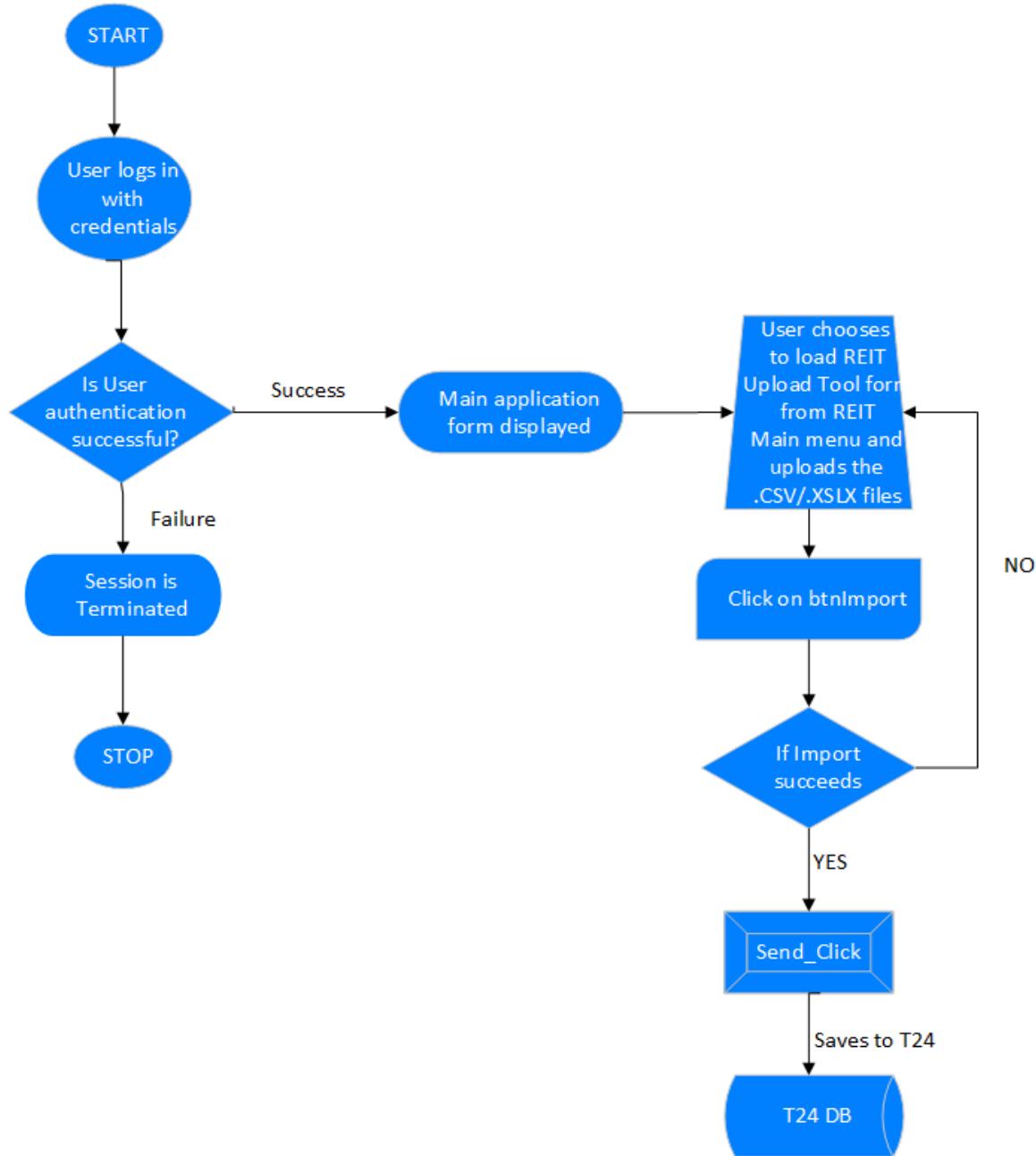


Figure 84: frmREIT 'Send Button Click' - Process Flow Diagram

Sequence flow diagram for REIT 'Send File button click'-

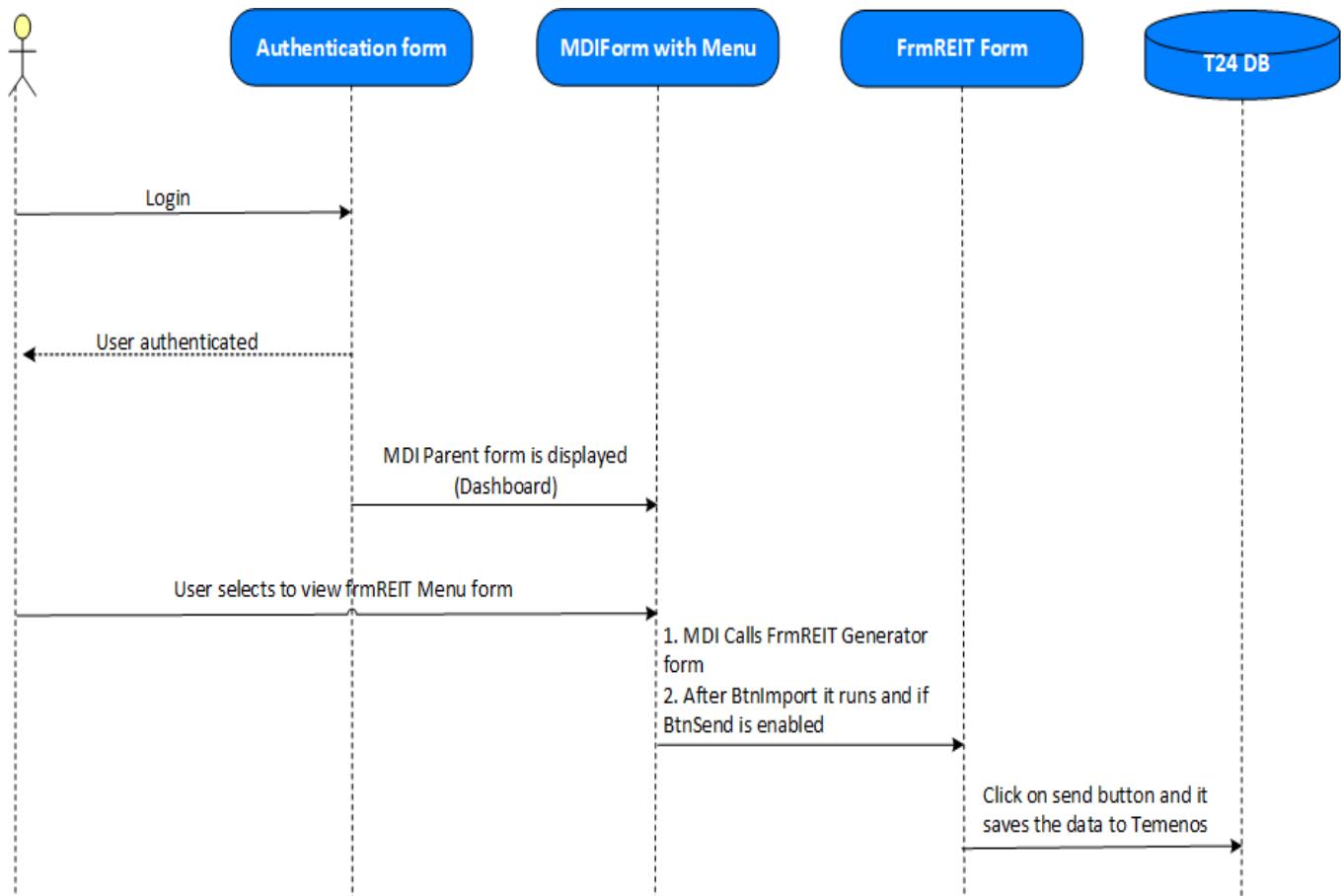


Figure 85: frmREIT 'Send File Button Click' - Sequence Diagram

4.8.4 REIT – Clear Batch button click

Once we click the Clear Batch button, it clears the Fund row dataset, makes the "NoOfUnit" label to show as Empty and makes the 'Send File button disabled.

Process Flow Diagram for 'Clear Batch button click'-

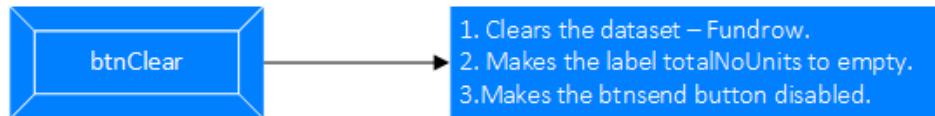


Figure 74: frmREIT 'Clear batch button click' - Process Flow Diagram

Class Diagram for frmREIT-

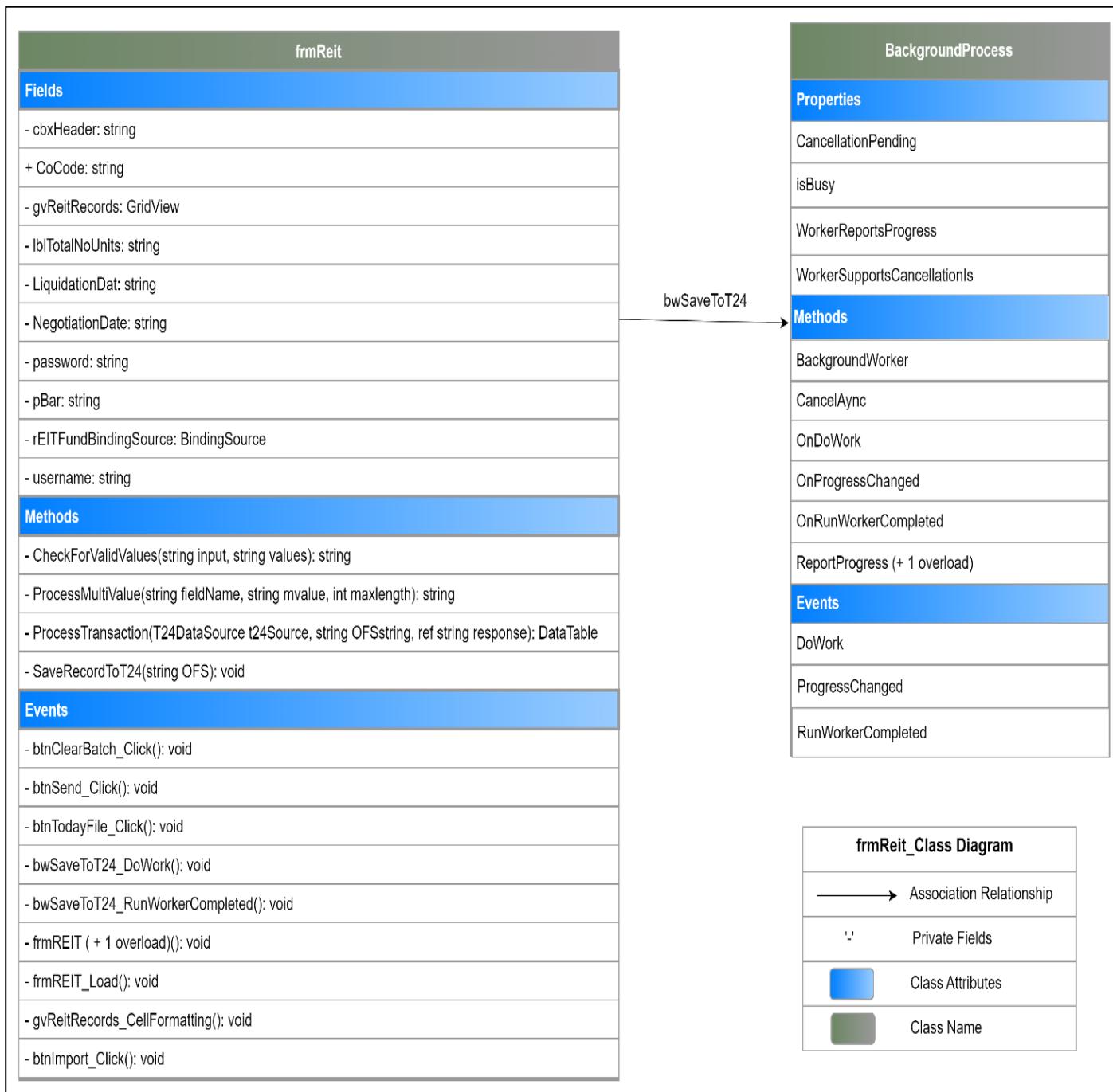


Figure 86: frmREIT – UML Class Diagram

4.9 DR Reporting Application – View Log

Code level details for View Log

While running the application, it writes the exception and information to the physical file .log in the directory of the server. The content of data from log file can be viewed on this screen. It has 2 methods:

1. Form Load
2. Refresh button click.

The View Log form view is below-

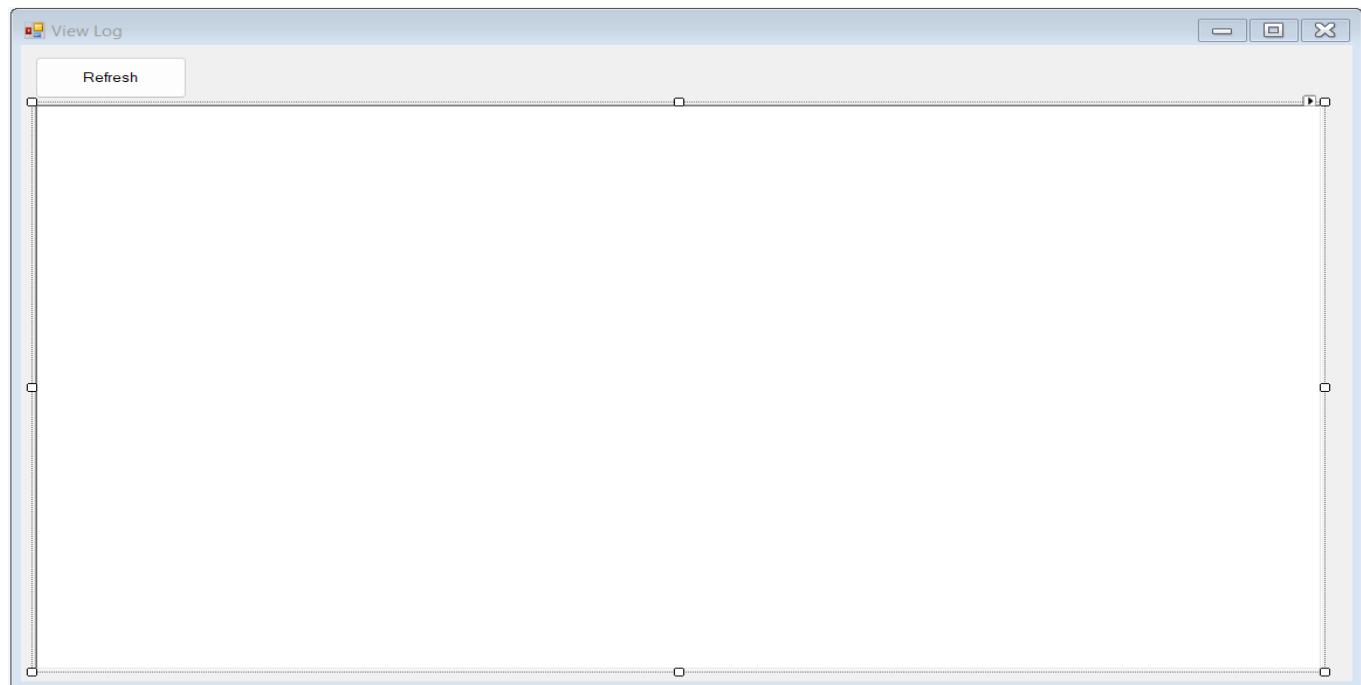


Figure 87: DRReporting View Log - Application View

4.9.1 ViewLog – Form Load action

During the form load process, it first checks if the current directory exists, if it exists then check if the log file exists or not, if the log file exists then read the content of the data from the file and load it into the ‘RichTextBox’ controller, then the user can view the data.

Process Flow Diagram for ViewLog ‘Form Load action’-

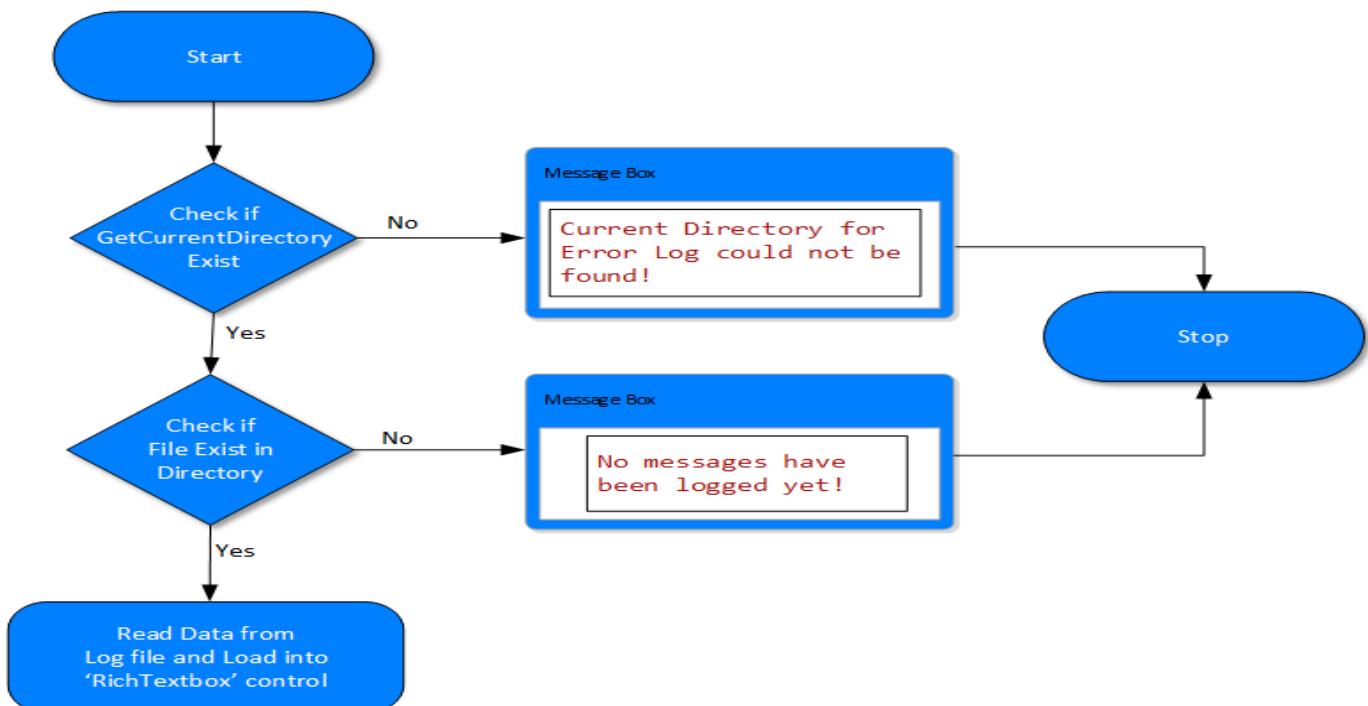


Figure 88: ViewLog ‘Form Load action’ - Process Flow Diagram

Sequence Diagram for ViewLog ‘Form Load action’-

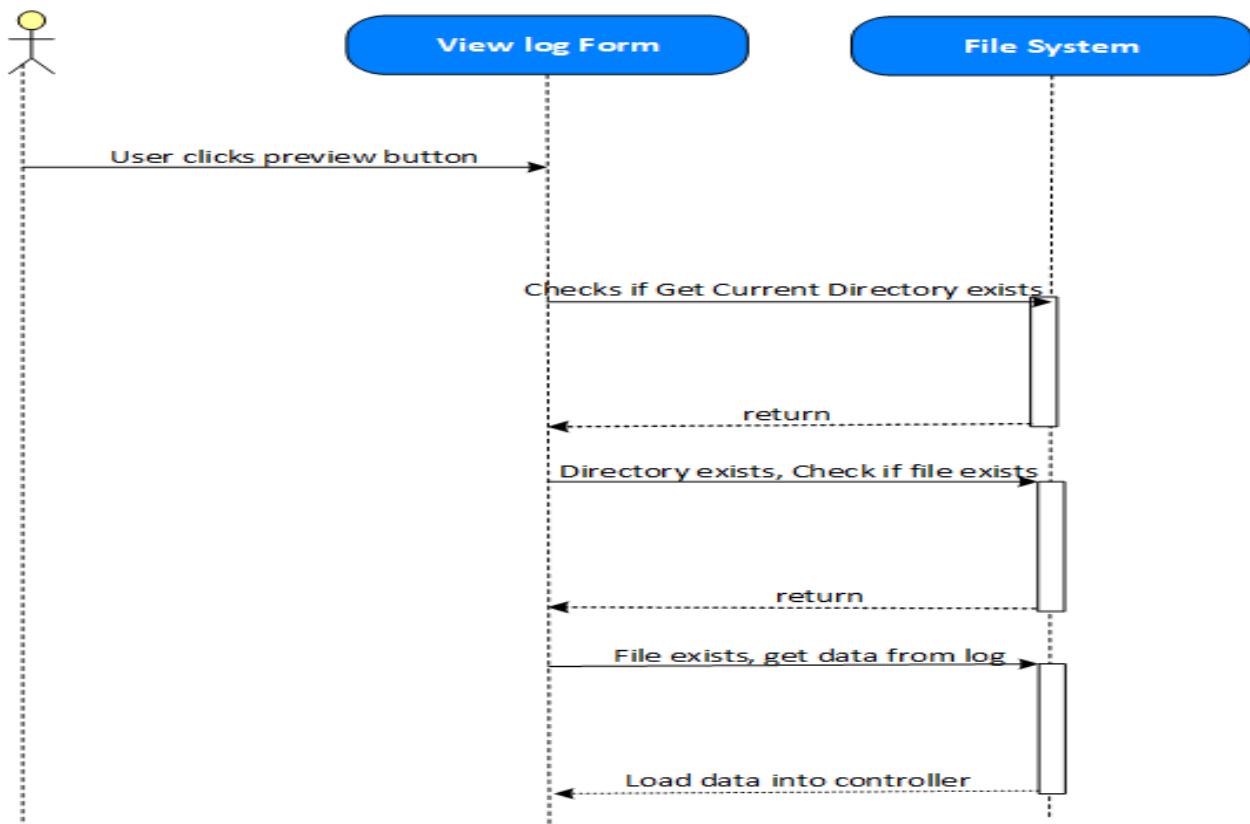
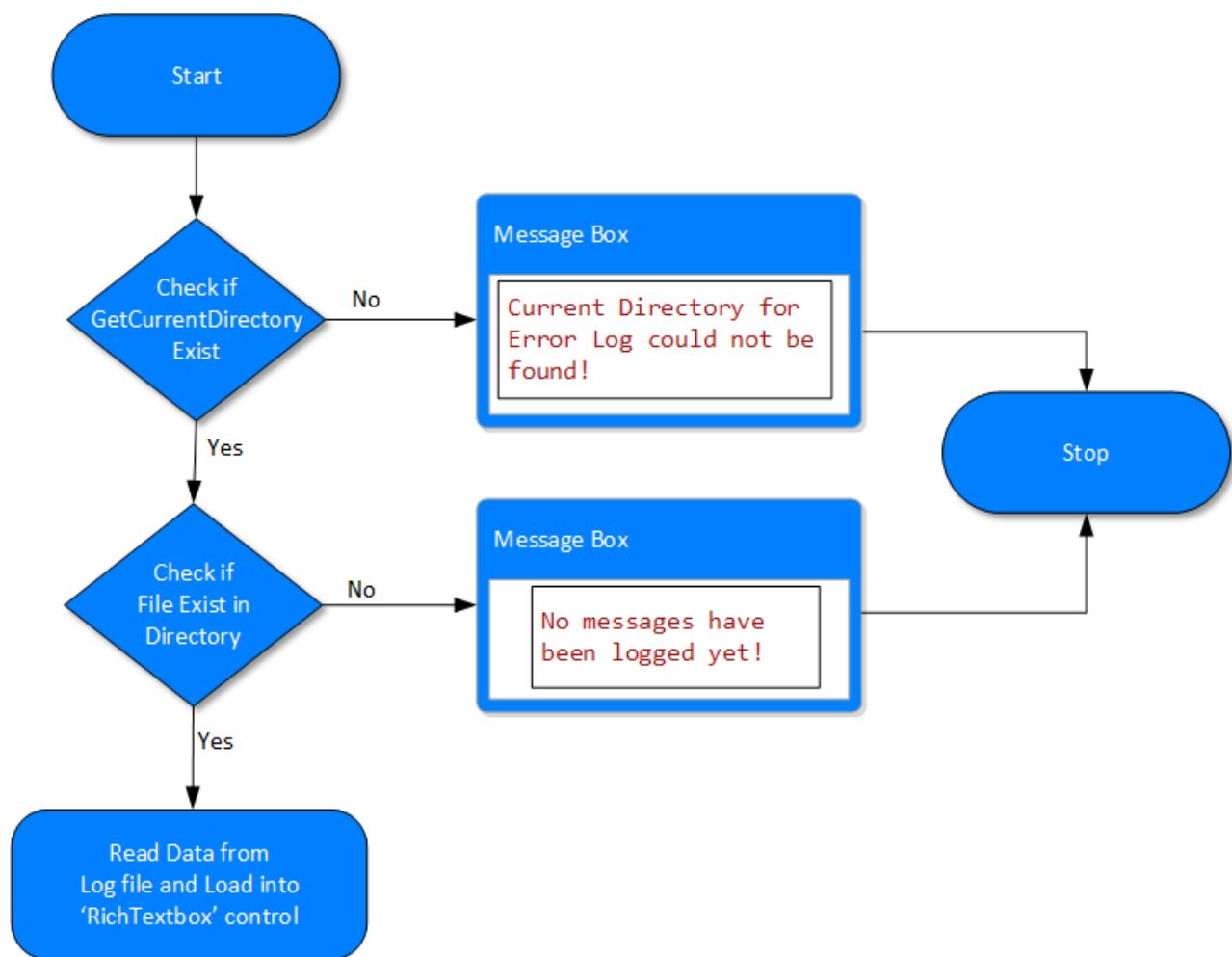


Figure 89: ViewLog 'Form Load action' - Sequence Diagram

4.9.2 ViewLog – Refresh button click

Upon clicking the “Refresh” button, first check if the current directory exists or not, if it exists then check if the log file exists, if log file exists, then read the content of the data from the file and load it into the “RichTextBox” controller, then the user can view the data.

Process Flow Diagram for ViewLog ‘Refresh Button Click’-



Sequence flow diagram Refresh Button - Sequence flow diagram

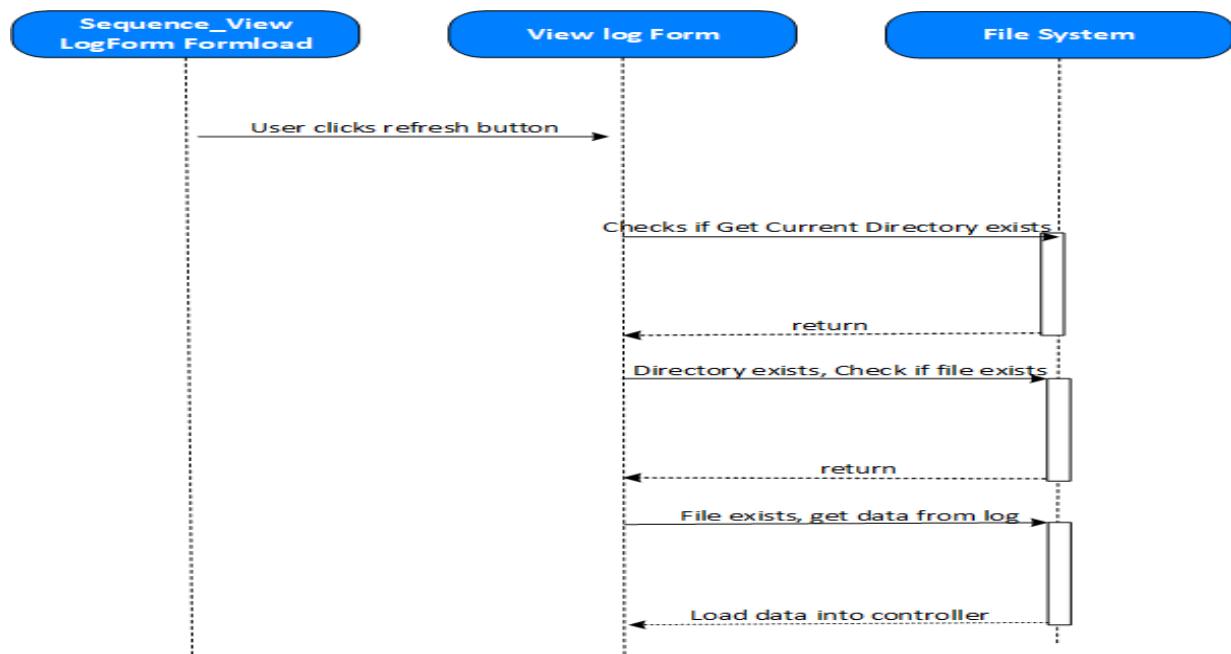


Figure 90: View Log 'Refresh Button Click' - Sequence Flow diagram

UML class level diagram details of the classes involved in View Log (frmViewLog)-

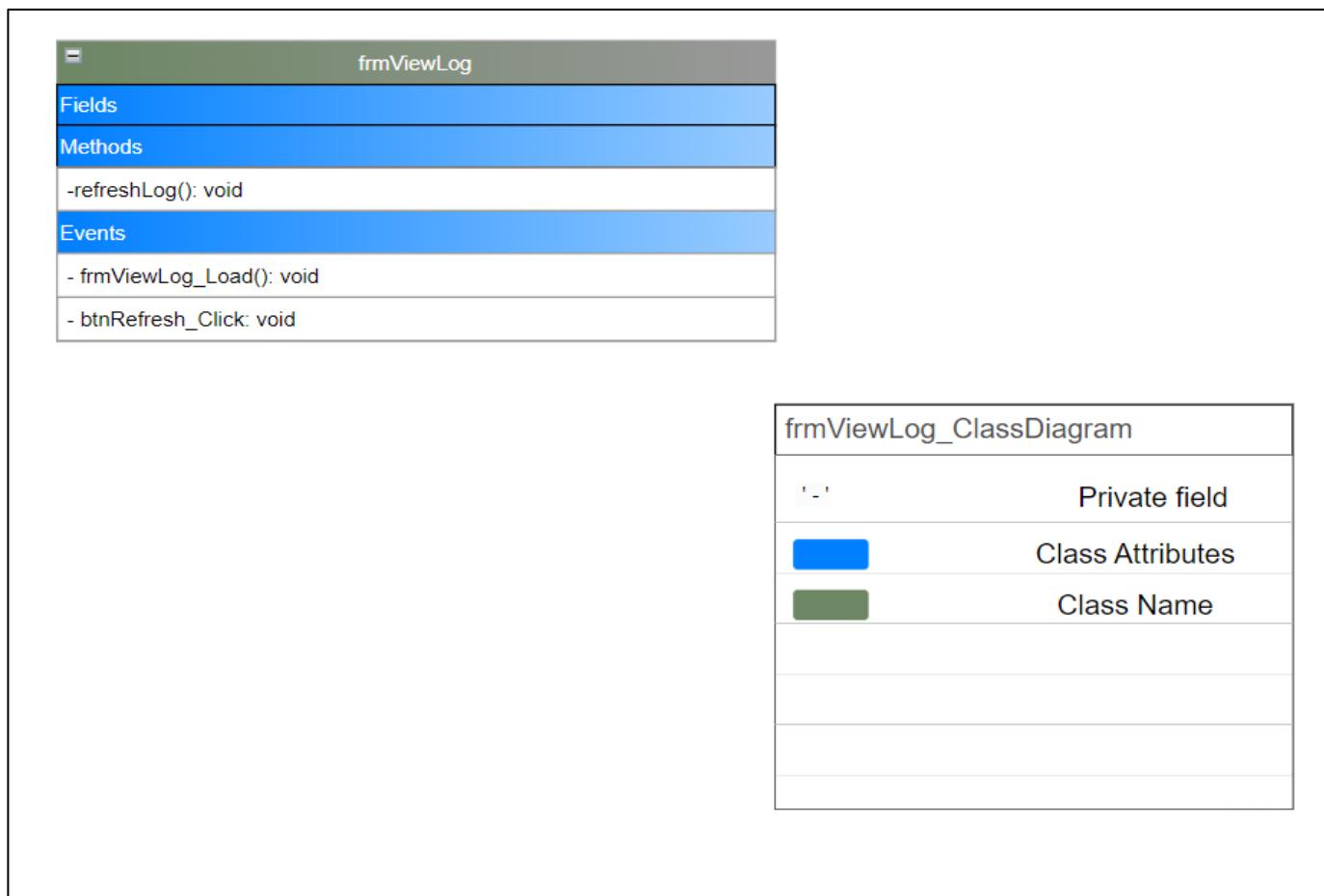


Figure 91: View Log – UML Class Diagram

5 DR Reporting Application – Email Services

An email web service typically refers to a set of APIs (Application Programming Interfaces) or web-based protocols that enable the sending, receiving, and management of emails over the internet.

Temenos T24 is widely used by financial institutions for their core banking operations. Temenos T24 provides a set of APIs – EMAIL that can be used for sending email integration purposes.

URL : <http://appsvr3-acs-dv.jmmb.com:8080/SendEmailWeb/SendEmailWSService?wsdl>

Code Level Details for Email Services

- Protocol: HTTP
- Communication: SOAP based
- Data Format: XML
- Interface- SendEmailWSChannel
- Interface- SendEmailWS
- SendEmailWS.sendMail(DRReporting.EmailWS.sendMailRequest)
- SendEmailWS.sendMailAsync(DRReporting.EmailWS.sendMailRequest)
- Class – SendEmailWSClient

METHODS:

- SendEmailWSClient.DRReporting.EmailWS.SendEmailWS.sendMail(DRReporting.EmailWS.sendMailRequest)
- SendEmailWSClient.DRReporting.EmailWS.SendEmailWS.sendMailAsync(DRReporting.EmailWS.sendMailRequest)
- SendEmailWSClient.SendEmailWSClient()
- SendEmailWSClient.SendEmailWSClient(string)
- SendEmailWSClient.SendEmailWSClient(string, string)
- SendEmailWSClient.SendEmailWSClient(string, System.ServiceModel.EndpointAddress)
- SendEmailWSClient.SendEmailWSClient(System.ServiceModel.Channels.Binding, System.ServiceModel.EndpointAddress)
- SendEmailWSClient.sendMail(string)
- SendEmailWSClient.sendMailAsync(string)

Class – sendMailRequest

- sendMailRequest.sendMailRequest()
- sendMailRequest.sendMailRequest(string)
- SendMailRequest.emailXML

Class – sendMailResponse

- sendMailResponse.sendMailResponse()
- sendMailResponse.sendMailResponse(DRReporting.EmailWS.result)
- SendMailResponse.return

Process flow diagram for Email Service-

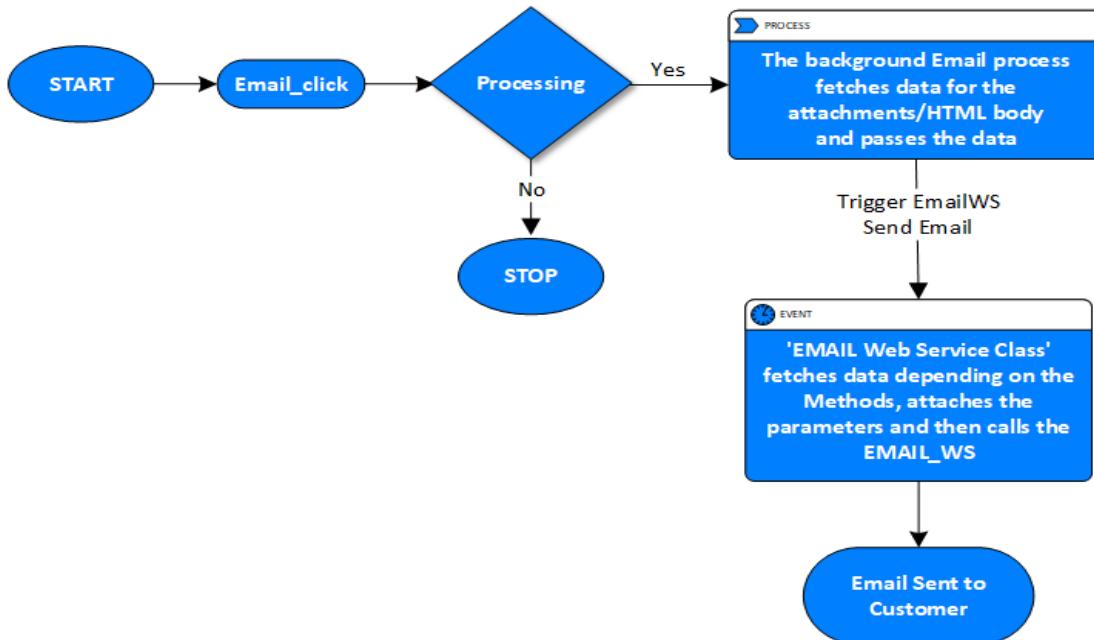


Figure 92: Email service - Process flow Diagram

Sequence Diagram for Email Service-

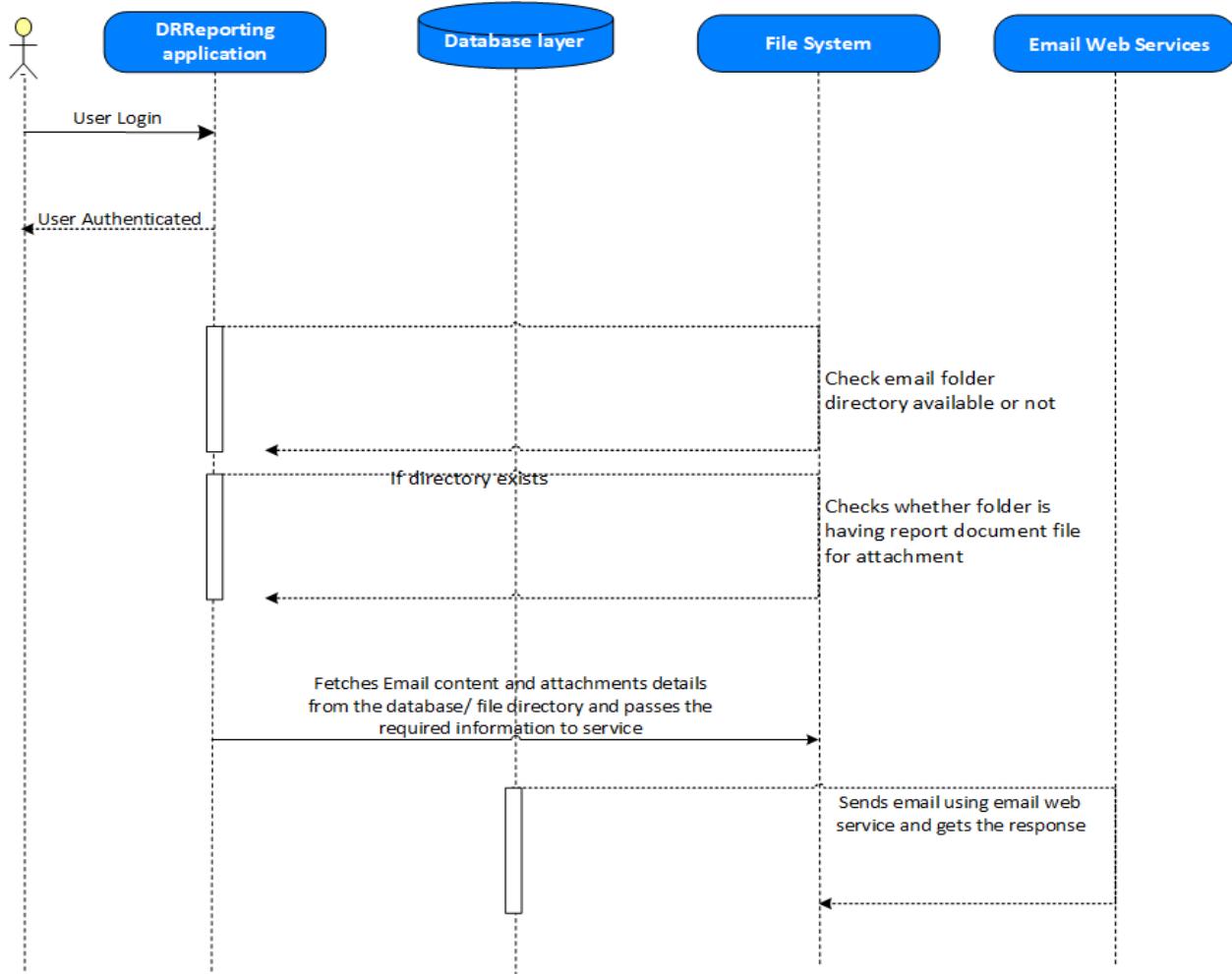


Figure 93: : Email service - Sequence Diagram

Class Diagram for Email Service

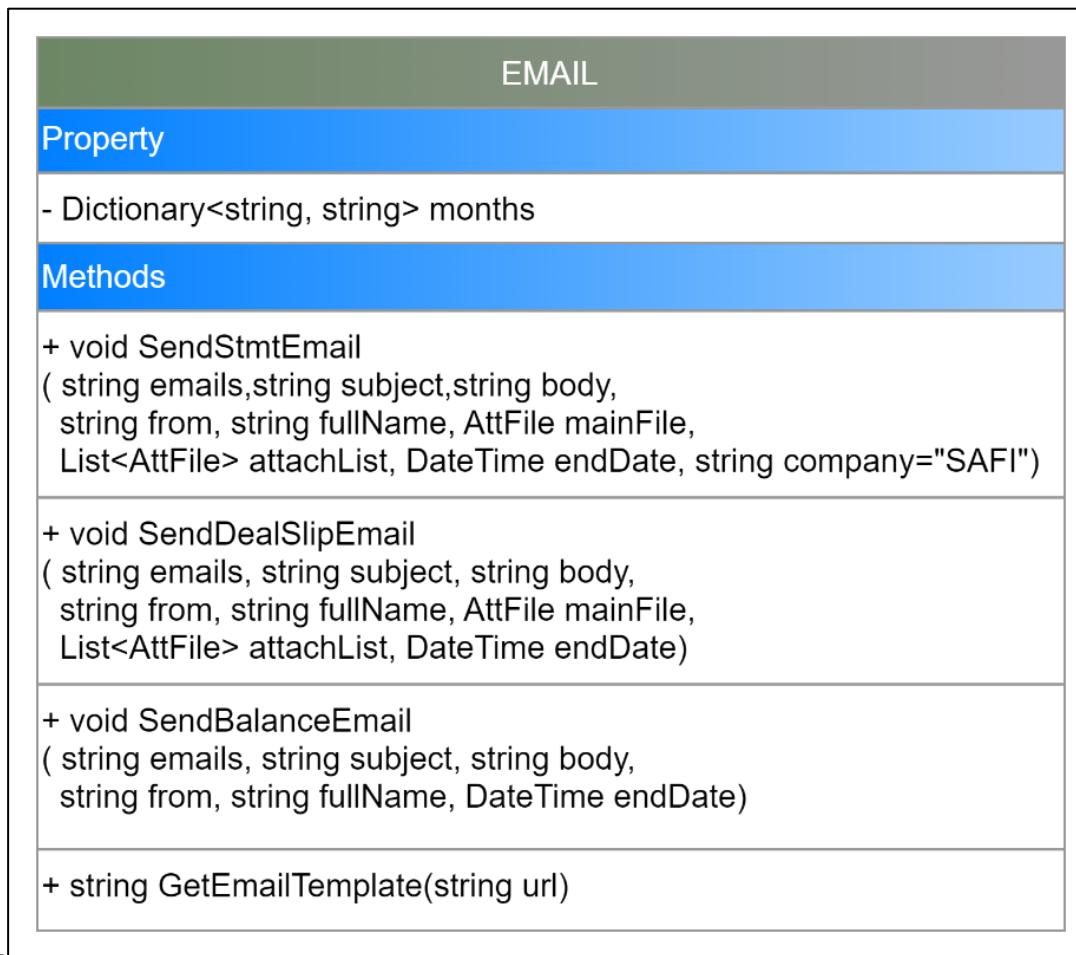


Figure 94: Email Service - UML Class Diagram

6 Data flow design

6.1 ER Diagrams

6.1.1 SAFI_DB ER Diagram



Figure 95: SAFI_DB ER Diagram

6.1.2 T24ReportingDB ER Diagram

ER Diagram – T24ReportingDB²

Note: The tables of T24ReportingDB have too many columns to fit in a word document legibly. The ER diagram Visio files have the tables in whole. The above image is a section table column.

[dbo].[tblCustomerData]	[dbo].[tblGM_CUSTOMER]	[dbo].[tblGM_FUND_MASTER]
PK [Customer_No] [int] NOT NULL [Short_Name] [varchar](255) NULL [Name] [varchar](255) NULL [Name2] [varchar](255) NULL [Street] [varchar](255) NULL [Parish] [varchar](255) NULL [Town] [varchar](255) NULL [Country] [varchar](255) NULL [Hold_Mail] [varchar](255) NULL [Postal_Code] [varchar](255) NULL [Sector] [varchar](255) NULL [Account_Officer] [varchar](255) NULL [Industry] [varchar](255) NULL [Target] [varchar](255) NULL [Status] [varchar](255) NULL [Nationality] [varchar](255) NULL [Residence] [varchar](255) NULL [Record_Status] [varchar](255) NULL [CPARTY] [varchar](255) NULL [TRN] [varchar](255) NULL [EMP_STATUS] [varchar](255) NULL [EMAIL] [nvarchar](1700) NULL [PROFESSION] [varchar](255) NULL	[PARTICIPANT_ID] [nvarchar](30) NOT NULL [CUSTOMER_ID] [nvarchar](10) NULL [FUND_ID] [nvarchar](18) NULL [PARTICIPANT_MNE] [nvarchar](15) NULL [OPENING_DATE] [nvarchar](11) NULL [CUST_ACCOUNT] [int] NULL [BRANCH_CODE] [nvarchar](4) NULL [ACCT_OFFICER] [int] NULL [CO_CODE] [nvarchar](11) NULL [DEPT_CODE] [nvarchar](4) NULL [T24DATE] [int] NULL [GM_STATUS] [nvarchar](10) NULL [COMPANYMNE] [nvarchar](3) NULL [REFERENCE_CCY] [nvarchar](3) NULL [CUST_DRDWN_ACCT] [nvarchar](12) NULL [BONUS_DIVIDEND] [nvarchar](8) NULL	[ID] [nvarchar](65) NOT NULL [FUND_NAME] [nvarchar](35) NULL [MNEMONIC] [nvarchar](15) NULL [TRUSTEE_NAME] [nvarchar](35) NULL [TRUSTEE_ADDR] [nvarchar](max) NULL [DEPOSITORY] [nvarchar](15) NULL [SEC_CODE] [nvarchar](10) NULL [FUND_FAMILY_ID] [nvarchar](15) NULL [CURRENCY] [nvarchar](3) NULL [FUND_TYPE] [nvarchar](12) NULL [DFUND_OR_PIP] [nvarchar](1) NULL [PIP_ALLOC_TBL] [nvarchar](11) NULL [START_DATE] [nvarchar](11) NULL [CLOSURE_DATE] [nvarchar](11) NULL [FUND_STATUS] [nvarchar](1) NULL [UNIT_FACE_VALUE] [nvarchar](10) NULL [MIN_INIT_SUBS] [nvarchar](16) NULL [MIN_INIT_UN_VAL] [nvarchar](5) NULL [MIN_ENSUING_SUBS] [nvarchar](16) NULL [MIN_ENS_UN_VAL] [nvarchar](5) NULL [MINIMUM_HOLDING] [nvarchar](10) NULL [MIN_HLDG_UN_VAL] [nvarchar](5) NULL [MIN_HOLD_PERIOD] [nvarchar](3) NULL

Figure 96: T24ReportingDB Process ER Diagram

7 Reports

7.1 Client Statement

² columns omitted for brevity

7.1.1 The SSRS data diagram of the Client Statement report

The SSRS data diagram of the Client Statement (ClientStatement.rdl) report describes datasets, parameters, images and datasource used by the report.

1. All parameters used by this report are listed.
2. All image names are listed.
3. Description of the datasource with connection properties are listed.
4. Dataset structure lists fields, datasource & query used. (DsMFStmt, Funds, DsCompany)

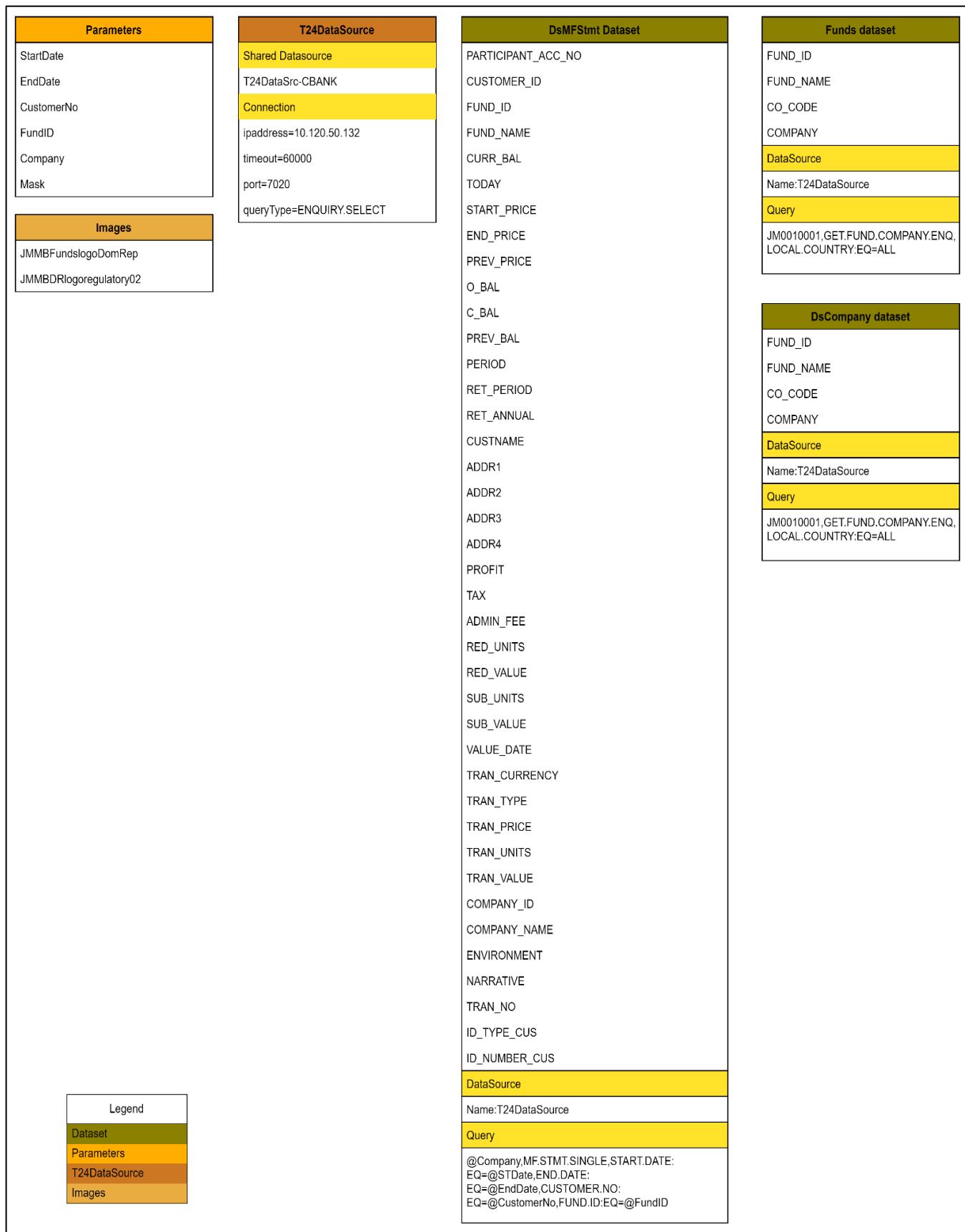


Figure 97: Client Statement report – SSRS Data Diagram

7.1.2 The SSRS Code diagram of the Client Statement report

The SSRS Code diagram of the Client Statement (ClientStatement.rdl) report describes the methods used by the report and lists the variables used by the report.

Description of methods-

1. SetVariableValue

Public function SetVariableValue has two input parameters and returns decimal value.

Input Parameter: Report variable(val) and decimal variable(newVal)

Function body: adds newVal value to val and returns the summation value

2. ReSetVariable

Public Function ReSetVariableValue has two input parameters and returns string value.

Input Parameters: Report variable (val) and decimal variable(newval).

Function body: assigns newVal decimal variable to report variable.

3. ConvertT24Date

Public Function ConvertT24Date has one input parameter and returns Datetime object.

Input Parameters: String variable T24Date.

Function body: Splits the date string using substring functions, concatenates it new date format and converts it to DateTime object using parse method and returns the new DateTiem object.

4. TranslateMonth

Public Function TranslateMonth has two string input parameters.

Input Parameters: String variable m and String variable input.

Function body: Creates a dictionary object **Months** of KeyValue pair with English name of the month as key and its Spanish counterpart as value. The Spanish month is found in dictionary using the m string parameter, Months(m) and this parameter is returned.

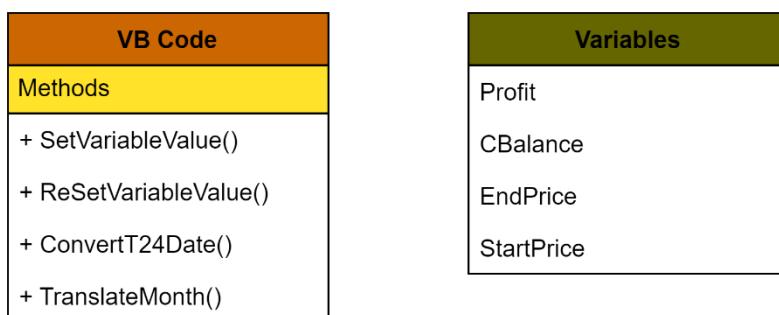


Figure 98: Client Statement report code diagram

7.2 Fondo Mutuo Rentabilidad

7.2.1 The SSRS data diagram of the Fondo Mutuo Rentabilidad report

The SSRS data diagram of the Fondo Mutuo Rentabilidad (FondoMutuoRentabilidad.rdl) report describes datasets, parameters, images and datasource used by the report.

1. All parameters used by this report are listed.
2. All image names are listed.
3. Description of the datasource with connection properties are listed.
4. Dataset's structure lists fields, datasource & query used. (Funds, DsCompany, DsInventory, DsMFStmt)

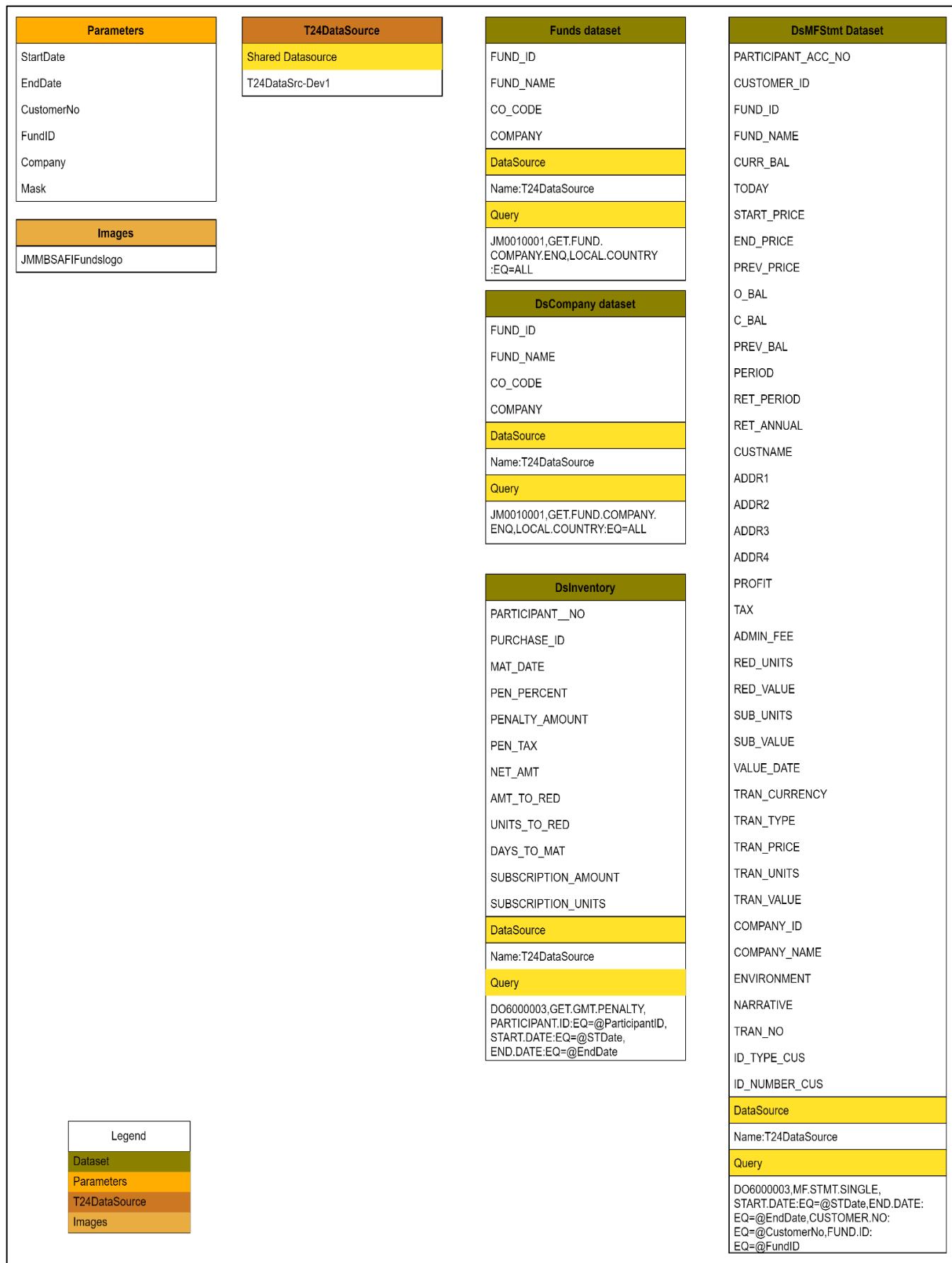


Figure 99: Fondo Mutuo Rentabilidad report – SSRS Data Diagram

7.2.2 The SSRS Code diagram of the Fondo Mutuo Rentabilidad report

The SSRS Code diagram of the Fondo Mutuo Rentabilidad (FondoMutuoRentabilidad.rdl) report describes the methods used by the report and list the variables used by the report. Description of methods-

1. SetVariableValue

Public function SetVariableValue has two input parameters and returns decimal value.

Input Parameter: Report variable(val) and decimal variable(newVal)

Function body: adds newVal value to val and returns the summation value

2. ReSetVariable

Public Function ReSetVariableValue has two input parameters and returns string value.

Input Parameters: Report variable (val) and decimal variable(newval).

Function body: assigns newVal decimal variable to report variable.

3. ConvertT24Date

Public Function ConvertT24Date has one input parameter and returns Datetime object.

Input Parameters: String variable T24Date.

Function body: Splits the date string using substring functions, concatenates it new date format and converts it to DateTime object using parse method and returns the new DateTiem object.

4. TranslateMonth

Public Function TranslateMonth has two string input parameters.

Input Parameters: String variable m and String variable input.

Function body: Creates a dictionary object Months of KeyValue pair with English name of the month as key and its Spanish counterpart as value. The Spanish month is found in dictionary using the m string parameter, Months(m) and this parameter is returned.

VB Code	Variables
Methods	
+ SetVariableValue()	Profit
+ ReSetVariableValue()	CBalance
+ ConvertT24Date()	EndPrice
+ TranslateMonth()	StartPrice

Figure 100: Fondo Mutuo Rentabilidad report code diagram

7.3 Fondo Plazo90 Statement

7.3.1 The SSRS data diagram of the Fondo Plazo90 Statement report

The SSRS data diagram of the Fondo Plazo90 Statement (FondoPlazo90Statement.rdl) report describes datasets, parameters, images and datasource used by the report.

1. All parameters used by this report are listed.
2. All image names are listed.
3. Description of the datasource with connection properties are listed.
4. Dataset's structure list fields, datasource & query used. (Funds, DsCompany, DsInventory, DsMFStmt)



Figure 101: Fondo Plazo90 Statement report – SSRS Data Diagram

7.3.2 The SSRS Code diagram of the Fondo Plazo90 Statement report

The SSRS Code diagram of the Fondo Plazo90 Statement (FondoPlazo90Statement.rdl) report describes the methods used by the report and list the variables used by the report.

Description of methods-

1. SetVariableValue

Public function SetVariableValue has two input parameters and returns decimal value.

Input Parameter: Report variable(val) and decimal variable(newVal)

Function body: adds newVal value to val and returns the summation value

2. ReSetVariable

Public Function ReSetVariableValue has two input parameters and returns string value.

Input Parameters: Report variable (val) and decimal variable(newval).

Function body: assigns newVal decimal variable to report variable.

3. ConvertT24Date

Public Function ConvertT24Date has one input parameter and returns Datetime object.

Input Parameters: String variable T24Date.

Function body: Splits the date string using substring functions, concatenates it new date format and converts it to DateTime object using parse method and returns the new DateTiem object.

4. TranslateMonth

Public Function TranslateMonth has two string input parameters.

Input Parameters: String variable m and String variable input.

Function body: Creates a dictionary object **Months** of KeyValue pair with English name of the month as key and its Spanish counterpart as value. The Spanish month is found in dictionary using the m string parameter, Months(m) and this parameter is returned.

VB Code	Variables
Methods	
+ SetVariableValue()	Profit
+ ReSetVariableValue()	CBalance
+ ConvertT24Date()	EndPrice
+ TranslateMonth()	StartPrice

Figure 102: Fondo Plazo90 Statement report code diagram

7.4 AFP Statement

7.4.1 The SSRS data diagram of the AFP Statement report

The SSRS data diagram of the AFP Statement (AFPSatement.rdl) report describes datasets, parameters, images and datasource used by the report.

1. All parameters used by this report are listed.
2. All image names are listed.
3. Description of the datasource with connection properties are listed.
4. Dataset's structure list fields, datasource & query used. (DsHeaderInfo, DsFondo, DsMovimientos, DsInvest, DsSemiAnnual, DsCommissions, DsSummary and DsAdicional)



Figure 103: AFP Statement report – SSRS Data Diagram

7.4.2 The SSRS Code diagram of the AFP Statement

The SSRS Code diagram of the AFP Statement (AFPStatement.rdl) report describes the methods used by the report and lists the variables used by the report.

Below .net refers to use for this report.

1. System.Drawing, Version=2.0.0.0
2. BarcodeLib, Version=1.1.0.0

Description of methods-

1. ConvertT24Date

This method attempts to parse the provided date string. If successful, it returns a DateTime object representing the parsed date. If there's an error during parsing for e.g., if the date string format doesn't match the expected format. it returns None and prints an error message.

2. ConvertPeriod

This method Converts the period using date part and returns the new date.

3. VBTTrim

This method takes an input string (original_string), an old substring to be replaced (old_substring) and a new substring (new_substring).It then uses the replace method available in the strings to perform the replacement. The modified string is then returned.

4. TranslateMonth

This method translates an input month value from English to Spanish. For this purpose, we use a dictionary to map English month names to their corresponding Spanish translations.

5. Convert

This method is used to convert string to bytes.

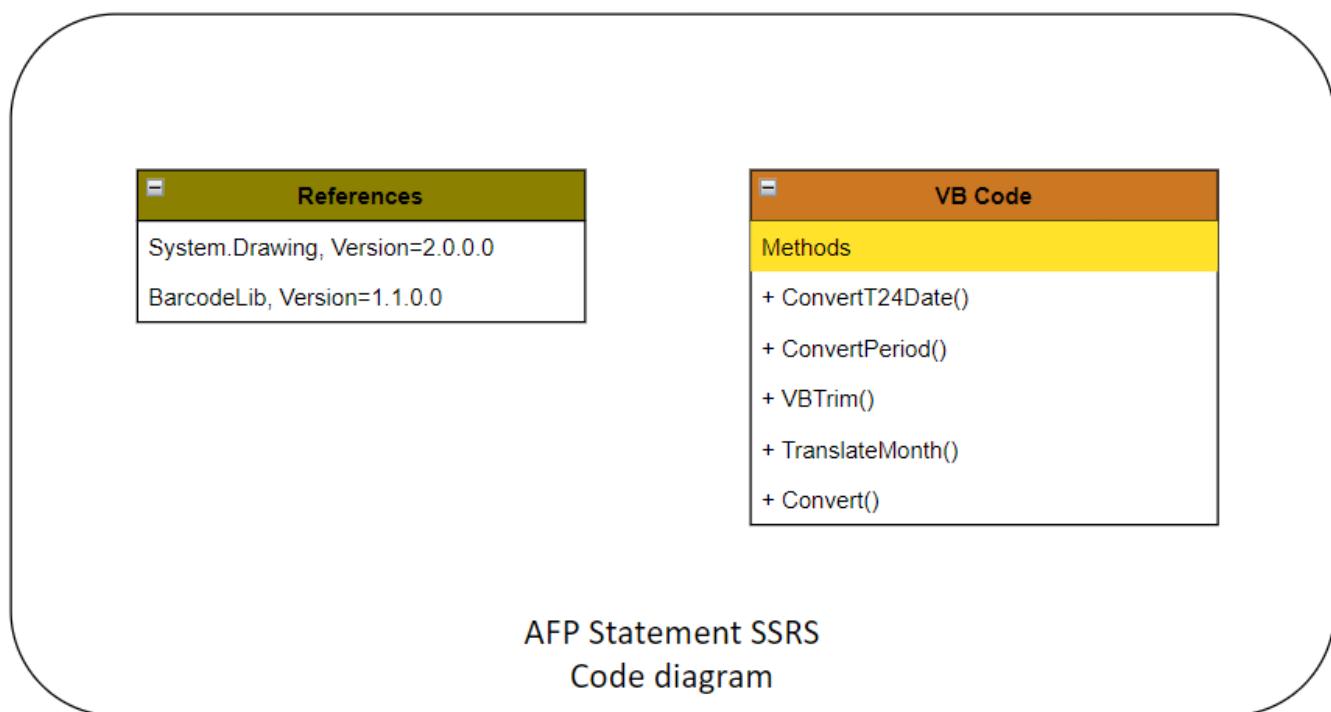


Figure 104: AFP Statement report – SSRS Code Diagram

7.5 InformacionDiariaDePublicar

7.5.1 The SSRS data diagram of the InformacionDiariaDePublicar report

The SSRS data diagram of the InformacionDiariaDePublicar (InformacionDiariaDePublicar.rdl) report describes datasets, parameters, images and datasource used by the report.

1. All parameters used by this report are listed.
2. All image names are listed
3. Description of the datasource with connection properties are listed.
4. Dataset's structure list fields, datasource & query used. (DsFunds, DsCompany, DsPublish)



Figure 105: InformacionDiariaDePublicar report – SSRS Data Diagram

7.5.2 The SSRS Code diagram of the InformacionDiariaDePublicar

The SSRS Code diagram of the InformacionDiariaDePublicar (InformacionDiariaDePublicar.rdl) report does not use any methods, references and variables.

7.6 Limitede Participaciony Relacionados

7.6.1 The SSRS data diagram of the Limitede Participaciony Relacionados report

The SSRS data diagram of the Limitede Participaciony Relacionados (LimitedeParticipacionyRelacionados.rdl) report describes datasets, parameters, images and datasource used by the report.

1. All parameters used by this report are listed.
2. All image names are listed
3. Description of the datasource with connection properties are listed.
4. Dataset's structure list fields, datasource & query used. (DsFunds, Company, DsCustomers,DsDate,)



7.6.2 The SSRS Code diagram of the Limitede Participaciony Relacionados report

The SSRS Code diagram of the Limitede Participaciony Relacionados (LimitedeParticipacionyRelacionados.rdl) report describes the methods used by the report and list the variables used by the report.

Description of methods-

1. ConvertT24Date

This method attempts to parse the provided date string. If successful, it returns a DateTime object representing the parsed date. If there's an error during parsing for e.g., if the date string format doesn't match the expected format, it returns None and prints an error message.

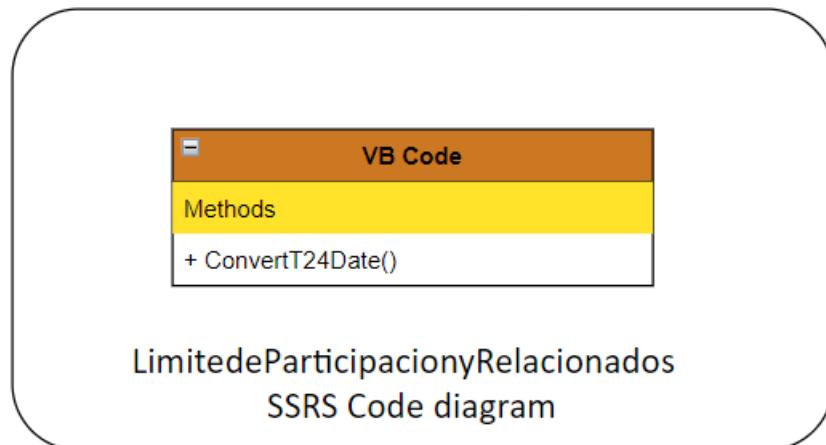


Figure 107: Limitede Participaciony Relacionados report – SSRS Code Diagram

7.7 Statement

7.7.1 The SSRS data diagram of the Statement report

The SSRS data diagram of the Statement (FundoMercadoDineroStatement.rdl) report describes datasets, parameters, images and datasource used by the report.

1. All parameters used by this report are listed.
2. All image names are listed.
3. Description of the datasource with connection properties are listed.
4. Dataset's structure list fields, datasource & query used. (Funds, DsCompany, DsInventory, DsMFStmt)

Parameters	T24DataSource	Funds Dataset	DsInventory Dataset	DsMFStmt Dataset
StartDate		FUND_ID	PARTICIPANT_NO	PARTICIPANT_ACC_NO
EndDate	Shared DataSource	FUND_NAME	PURCHASE_ID	CUSTOMER_ID
CustomerNo	/T24 Reports/Data Sources/ T24DataSrc-TRUMP	CO_CODE	MAT_DATE	FUND_ID
FundID		COMPANY	PEN_PERCENT	FUND_NAME
Company		DataSource	PENALTY_AMOUNT	CURR_BAL
Mask		Name: T24DataSource	PEN_TAX	TODAY
Images		Query	NET_AMT	START_PRICE
JMMSAFIFundslogo		JM0010001,GET.FUND. COMPANY,ENQ.LOCAL, COUNTRY:EQ=ALL	AMT_TO_RED	END_PRICE
			UNITS_TO_RED	PREV PRICE
			DAYS_TO_MAT	O_BAL
			SUBSCRIPTION_AMOUNT	C_BAL
			SUBSCRIPTION_UNITS	PREV_BAL
		DataSource	DataSource	PERIOD
		Name: T24DataSource	Name: T24DataSource	RET_PERIOD
		Query	Query	RET_ANNUAL
		JM0010001,GET.FUND.COMPANY. ENQ.LOCAL, COUNTRY:EQ=ALL	DO6000008,GET.GMT.PENALTY, PARTICIPANT.ID: EQ=@ParticipantID, START.DATE:EQ=@STDDate, END.DATE:EQ=@EndDate	CUSTNAME
				ADDR1
				ADDR2
				ADDR3
				ADDR4
				PROFIT
				TAX
				ADMIN_FEE
				RED_UNITS
				RED_VALUE
				SUB_UNITS
				SUB_VALUE
				VALUE_DATE
				TRAN_CURRENCY
				TRAN_TYPE
				TRAN_PRICE
				TRAN_UNITS
				TRAN_VALUE
				COMPANY_ID
				COMPANY NAME
				ENVIRONMENT
				NARRATIVE
				TRAN_NO
				ID_TYPE_CUS
				ID NUMBER CUS
Legend		DataSource		DataSource
DataSet		Name: T24DataSource		Name: T24DataSource
Parameters		Query		Query
T24DataSource				DO6000008,MF,STMT,SINGLE,START,DATE: EQ=@STDDate,END,DATE: EQ=@EndDate, CUSTOMER.NO:EQ=@CustomerNo, FUND.ID:EQ=@FundID
Images				

Figure 108: Statement report – SSRS Data Diagram

7.7.2 The SSRS Code diagram of the Statement

The SSRS Code diagram of the Statement (FundoMercadoDineroStatement.rdl) report describes the methods used by the report and list the variables used by the report.

Description of methods.

1. Set Variable

value1 and value2 are parameters representing the two values you want to add.

Inside this method, result is a variable storing the sum of value1 and value2.

The return statement is used to send the result back to the calling code.

2. Reset Variable

The ReSetVariable method takes the name of a variable (variable_name) and a new value (new_value).

It assumes the existence of a global storage mechanism for variables.

If the variable exists, its value is updated with the new value; otherwise, an error message is printed, and None is returned.

3. ConvertT24Date

The ConvertT24Date method attempts to parse the provided date string.

If successful, it returns a DateTime object representing the parsed date.

If there's an error during parsing for e.g., if the date string format doesn't match the expected format,

it returns None and prints an error message.

4. Translate Month

That translates an input month value from English to Spanish. For this purpose, we'll use a dictionary

to map English month names to their corresponding Spanish translations.

has context menu

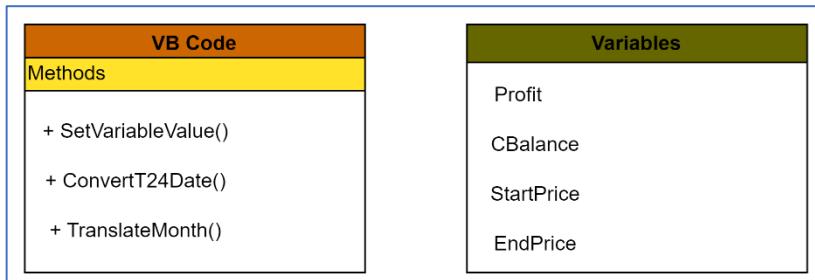


Figure 109: FundoMercadoDineroStatement report - Code Diagram

7.8 Dealslips

7.8.1 The SSRS data diagram of the Dealslips report

The SSRS data diagram of the Dealslips (FundoMercadoDineroDealSlip.rdl) report describes datasets, parameters, images and datasource used by the report.

1. All parameters used by this report are listed.
2. All image names are listed
3. Description of the datasource with connection properties are listed.
4. Dataset's structure list fields, datasource & query used. (Funds, DsDealSlip)

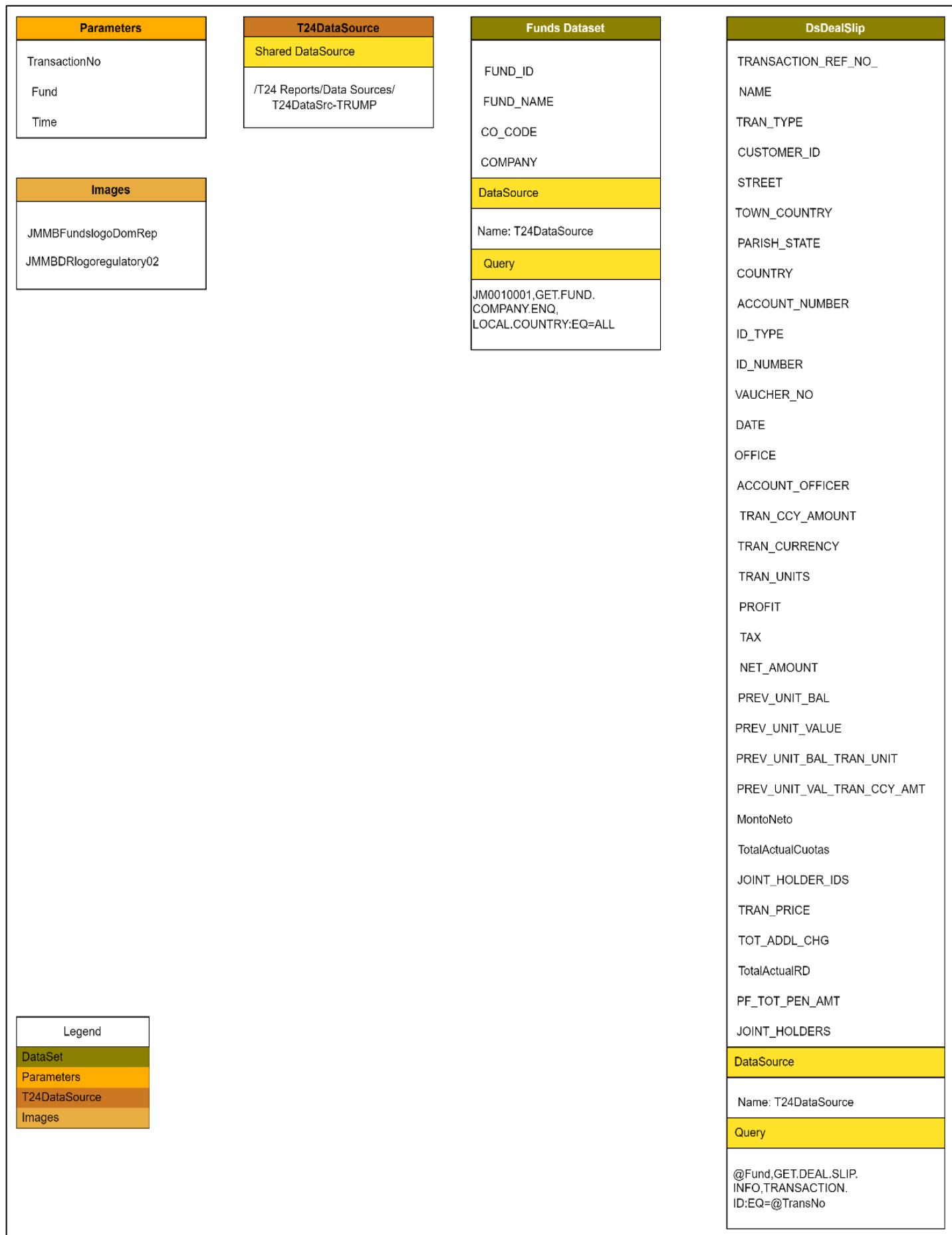


Figure 110: DealSlips report – SSRS Data Diagram

7.8.2 The SSRS Code diagram of the Dealslips

The SSRS Code diagram of the dealslips (FundoMercadoDineroDealSlip.rdl) report describes the methods used by the report.

Description of methods/function

VBTrim

The VBTrim takes an input string (original_string), an old substring to be replaced (old_substring), and a new substring (new_substring). It then uses the replace method available in the strings to perform the replacement. The modified string is then returned.

This function replaces the string with vbcr to "" and vbcrLF to "" from the given input string and returns the modified string

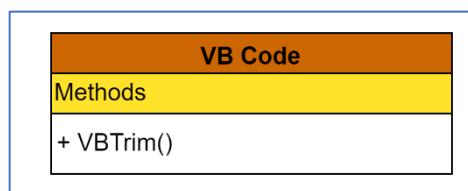


Figure 111: FundoMercadoDineroDealSlip report - Code diagram

7.9 Regulatory

7.9.1 The SSRS data diagram of the Regulatory report

The SSRS data diagram of the regulatory report (ValoracionDiariaDeLaCuota.rdl) report describes datasets, parameters, images and datasource used by the report.

1. All parameters used by this report are listed.
2. All image names are listed.
3. Description of the datasource with connection properties are listed.
4. Dataset's structure list fields, datasource & query used. (DsFunds, DsCompany, DsFundValuation)

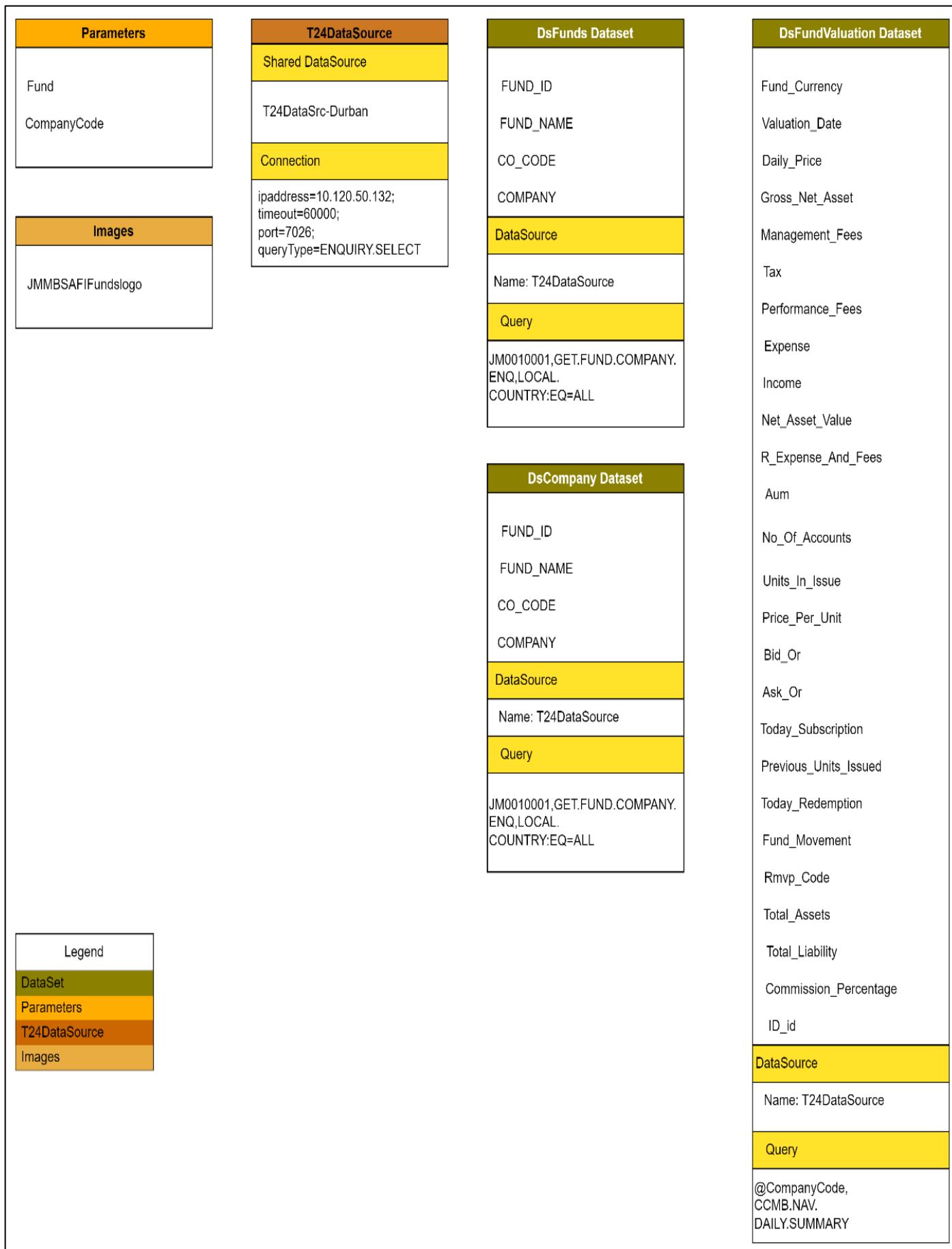


Figure 112: Regulatory report – SSRS Data Diagram

7.9.2 The SSRS Code diagram of the Regulatory report

The SSRS Code diagram of the regulatory (ValoracionDiariaDeLaCuota.rdl) report describes the methods used by the report and lists the variables used by the report.

Description of methods-

1. ConvertT24Date

The ConvertT24Date method attempts to parse the provided date string. If successful, it returns a DateTime object representing the parsed date. If there's an error during parsing for e.g., if the date string format doesn't match the expected format, it returns None and prints an error message.

This method parses date string in the required format and returns it as DateTime object.

```
NewDate = T24Date.Substring(0,4) + “-“ + T24Date.Substring(4,2) + “-“ + T24Date.Substring(6,2)
```

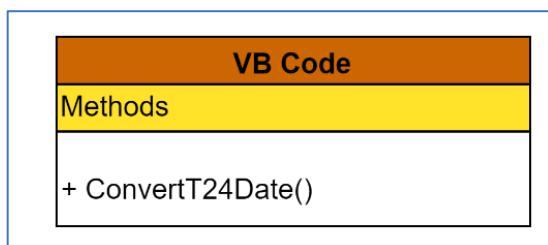


Figure 113: ValoracionDiariaDeLaCuota Report - Code Diagram

7.10 PortafolioDeInversiones

7.10.1 The SSRS data diagram of the PortafolioDeInversiones report

The SSRS data diagram of the PortafolioDeInversiones (PortafolioDeInversiones.rdl) report describes datasets, parameters, images and datasource used by the report.

1. All parameters used by this report are listed.
2. All image names are listed.
3. Description of the data source with connection properties are listed.
4. Dataset's structure list fields, datasource & query used. (DsFunds, DsCompany, DsInvestments)

Parameters	T24DataSource	DsFunds dataset	DSInvestments dataset
Fund	Shared Datasource	FUND_ID	Line_Type
CompanyCode	T24DataSrc-Durban	FUND_NAME	Issuer_Name
ReportDate	Connection	CO_CODE	Asset_Type_Desc
Images	ipaddress=10.120.50.132 timeout=60000 port=7026 queryType=ENQUIRY.SELECT	COMPANY	Sec_No
JMMBFundslogo		DataSource	Category
		Name:T24DataSource	Customer_Sector
		Query	Sec_Desc
		JM0010001,GET.FUND.COMPANY. ENQ,LOCAL.COUNTRY:EQ=ALL	Sec_Currency
			Isin
			Weight
Legend			Nominal
Dataset			Nominal_Lcy
Parameters			Days_To_Mature
T24DataSource			Duration_M
Images			Yield_To_Mature
			Market_Price
			Settlement_Date
			Sec_Mature_Date
			Sec_Int_Rate
			No_Of_Payments
			Int_Day_Basis
			Market_Value
			Last_Int_PayDt
			Next_Int_PayDt
			Accrued_Int
			Asset_Value
			Project_Acc_Int
			NAV_Date
			ID_id
			Acquire_Date
			AssetType
			No_of_Months
		DataSource	
		Name:T24DataSource	
		Query	
		@CompanyCode,CCMB. NAV.DAILY.DETAIL	

Figure 114: PortafolioDeInversiones report – SSRS Data Diagram

7.10.2 The SSRS Code diagram of the PortafolioDeInversiones

The SSRS Code diagram of the PortafolioDeInversiones (PortafolioDeInversiones.rdl) report describes the methods used by the report and lists the variables used by the report.

Description of methods-

1. SetVariableValue

This method sets the variable value to 0 if it is unassigned.

2. AddVariableValue

This methods add's two values passed to it and return's the new value.

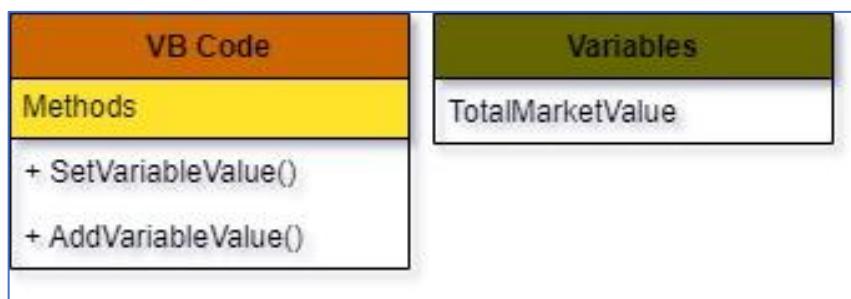


Figure 115: PortafolioDeInversiones report - Code Diagram

7.11 ReporteDeLiquidez

7.11.1 The SSRS data diagram of the ReporteDeLiquidez report:

The SSRS data diagram of the ReporteDeLiquidez (ReporteDeLiquidez.rdl) report describes datasets, parameters, images and datasource used by the report.

1. All parameters used by this report are listed.
2. All image names are listed.
3. Description of the datasource with connection properties are listed.
4. Dataset's structure list fields, datasource & query used. (DsFunds, DsCompany, DsLiquidity)

Parameters	T24DataSource	DsFunds dataset	DsLiquidity dataset
Fund	Shared Datasource	FUND_ID	Line_Type
CompanyCode	T24DataSrc	FUND_NAME	Issuer_Name
ReportDate	Connection	CO_CODE	Asset_Type_Desc
Images	ipaddress=10.120.50.130 timeout=60000 port=7025 queryType=ENQUIRY.SELECT	COMPANY	Sec_No
JMMBSAFIFundsLogo		DataSource	Category
		Name:T24DataSource	Customer_Sector
		Query	Sec_Desc
		JM0010001,GET.FUND.COMPANY. ENQ,LOCAL,COUNTRY:EQ=ALL	Sec_Currency
			Isin
			Weight
			Nominal
			Nominal_Lcy
			Days_To_Mature
			Duration_M
			Yield_To_Mature
			Market_Price
			Settlement_Date
			Sec_Mature_Date
			Sec_Int_Rate
			No_Of_Payments
			Int_Day_Basis
			Market_Value
			Last_Int_PayDt
			Next_Int_PayDt
			Accrued_Int
			Asset_Value
			Project_Acc_Int
			NAV_Date
			ID_id
		DataSource	
		Name:T24DataSource	
		Query	
		@CompanyCode,CCMB.NAV. DAILY.DETAIL	

Figure 116: ReporteDeLiquidez report – SSRS Data Diagram

7.11.2 The SSRS Code diagram of the ReporteDeLiquidez

The SSRS Code diagram of the ReporteDeLiquidez (ReporteDeLiquidez.rdl) report describes the methods used by the report and lists the variables used by the report.

Note: Code, References and Variables not Used for this report

7.12 SuscripcionRescatesCuotas

7.12.1 The SSRS data diagram of the SuscripcionRescatesCuotas report

The SSRS data diagram of the SuscripcionRescatesCuotas (SuscripcionRescatesCuotas.rdl) report describes datasets, parameters, images and datasource used by the report.

1. All parameters used by this report are listed.
2. All image names are listed.
3. Description of the datasource with connection properties are listed.
4. Dataset's structure list fields, datasource & query used. (DsFunds, DsCompany, DsCustomers)

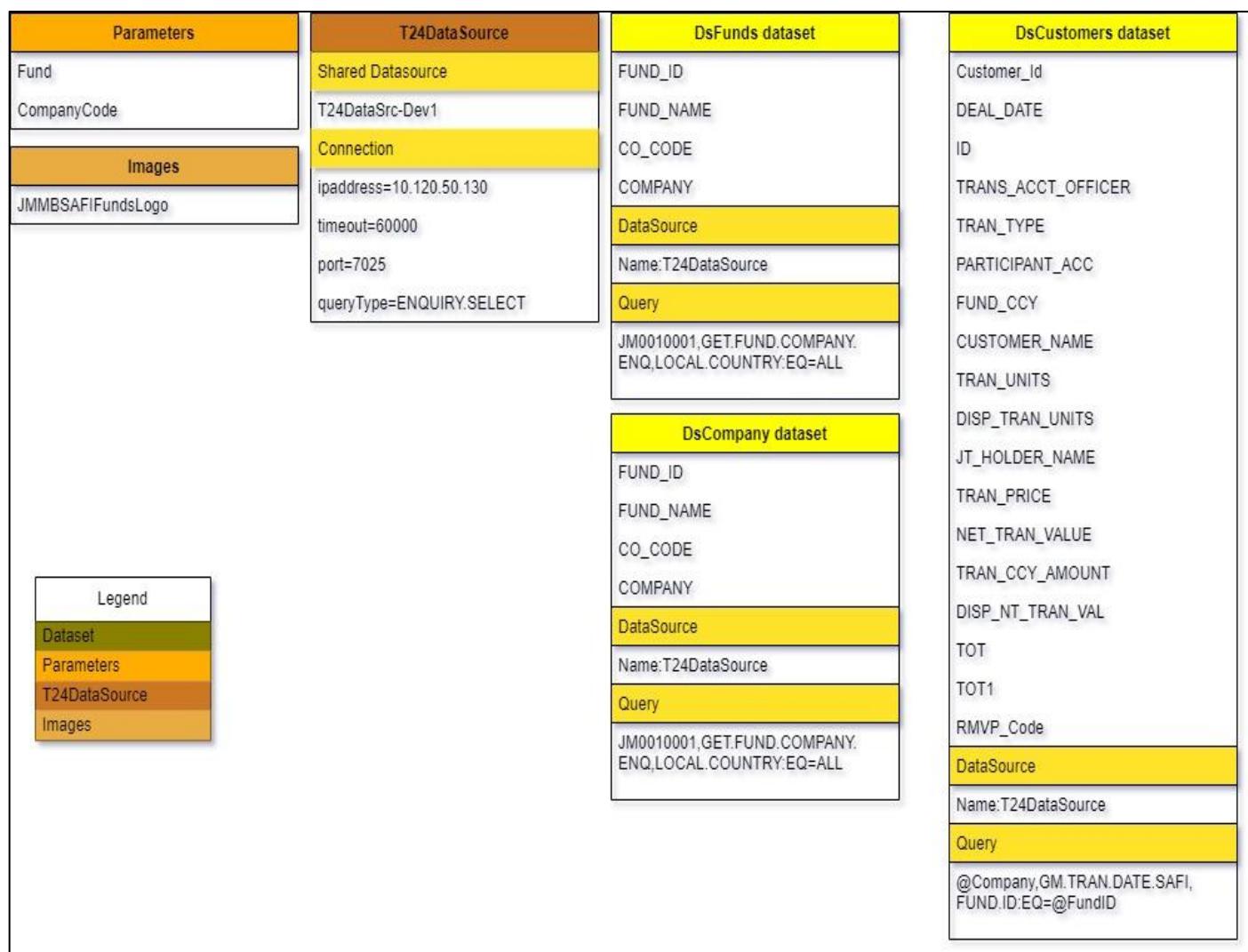


Figure 117: SuscripcionRescatesCuotas report – SSRS Data Diagram

7.12.2 The SSRS Code diagram of the SuscripcionRescatesCuotas

The SSRS Code diagram of the SuscripcionRescatesCuotas (SuscripcionRescatesCuotas.rdl) report describes the methods used by the report and lists the variables used by the report.

Description of methods-

1. ConvertT24Date

This method parses date string in the required format and returns it as DateTime object.

2. AddVariableValue

This method adds two values passed to it and returns the new value.

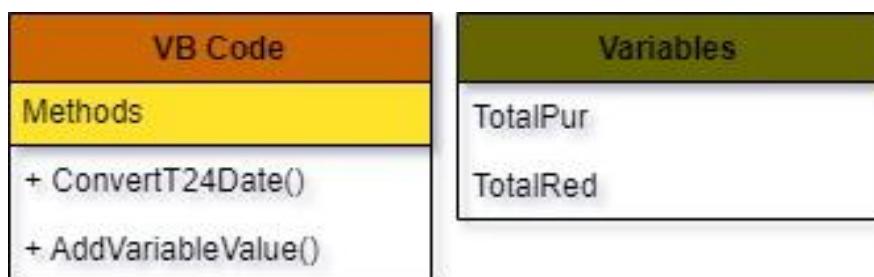


Figure 118: SuscripcionRescatesCuotas - Code Diagram

7.13 DOMRepDealSlip

7.13.1 The SSRS data diagram of the DOMRepDealSlip report

The SSRS data diagram of the Client DOMRepDealSlip (DOMRepDealSlip.rdl) report describes datasets, parameters, images and datasource used by the report.

1. All parameters used by this report are listed.
2. All image names are listed.
3. Description of the datasource with connection properties are listed.
4. Dataset's structure list fields, datasource & query used. (DsDealSlip, Funds)

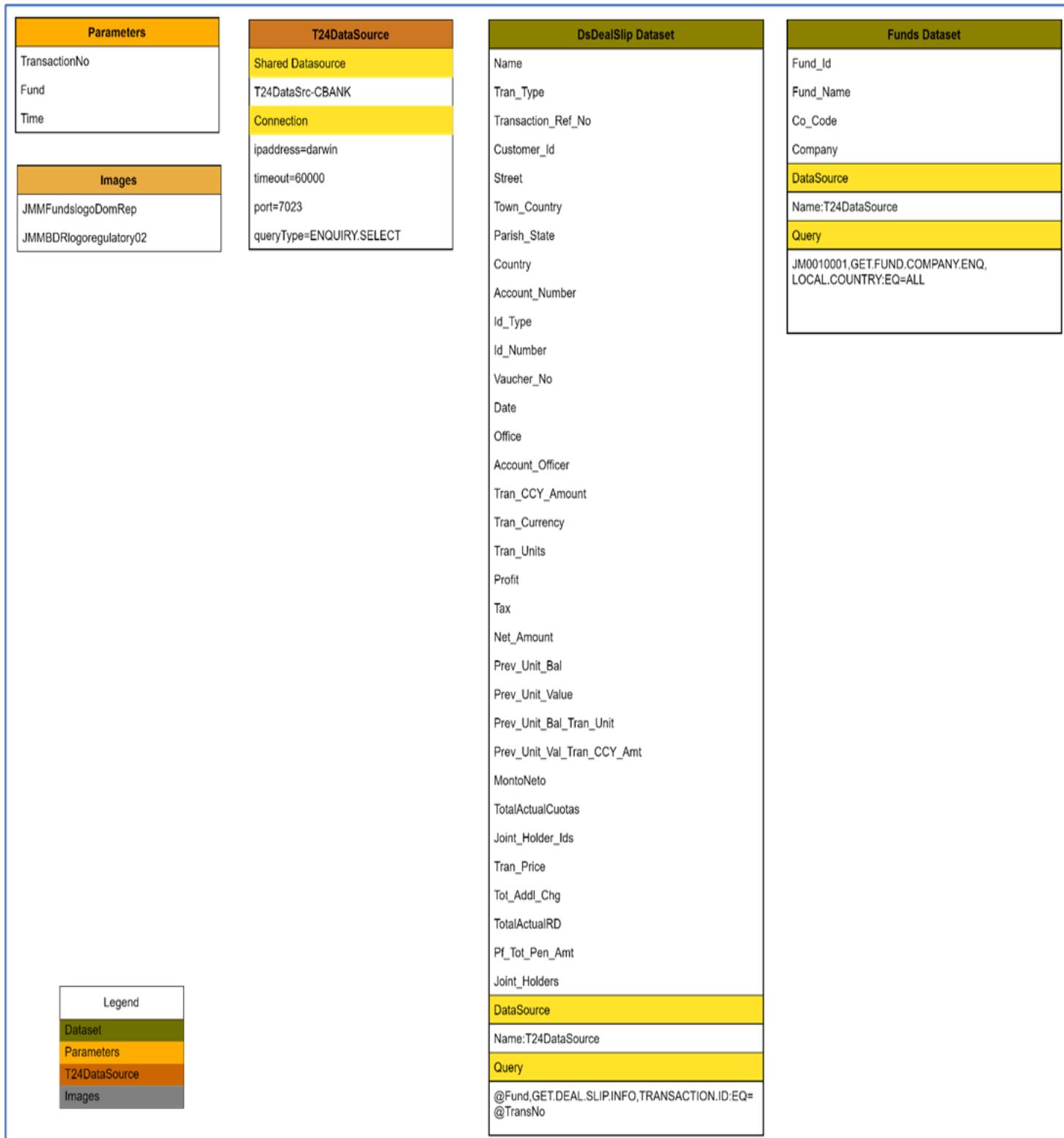


Figure 119: DOMRepDealSlip report – SSRS Data Diagram

7.13.2 The SSRS Code diagram of the DOMRepDealSlip

The SSRS Code diagram of the DOMRepDealSlip (DOMRepDealSlip.rdl) report describes the methods/functions used by the report and list the variables used by the report.

Description of methods/Functions-

1. VBTrim

This function removes/replaces the Carriage return and the Carriage Line feed with empty values and returns the string with the changes.

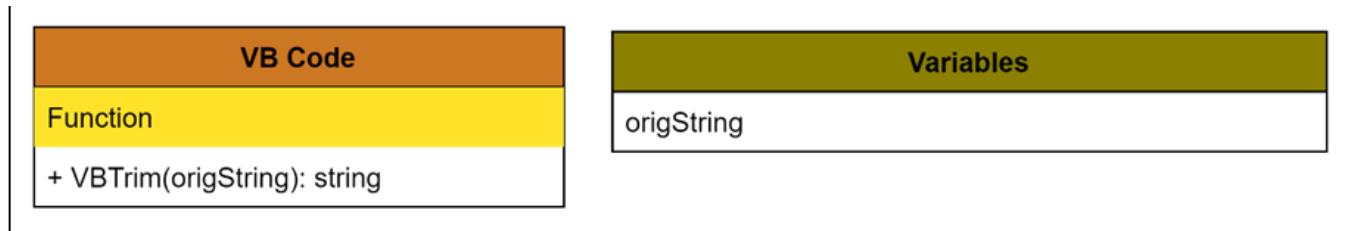


Figure 120: DOMRepDealSlip Report - Code Diagram

8 User Roles & Permission

<>

9 Exception Handling

<>

10 Assumptions

<>

11 Exception Handling:

<>

Logging:

12 Exclusion

Miscellaneous Files

<>

13 Libraries

13.1 Internal Libraries

Internal Libraries	
References	Description
T24Connector	This T24Connector dll is used for connecting database and to retrieve data from database

13.2 External Libraries

External Libraries	
References	Description
EPPlus	<p>EPPlus is a .NET Framework/.NET Core library for managing Office Open XML spreadsheets, distributed via Nuget . Version 5 supports .NET Framework from version 3.5 and .NET Core from version 2.0. It is currently a single DLL and has no dependencies on any other library such as Microsoft Excel.</p> <p>The API documentation provided is generated from the XML comments in the EPPlus source code. It is updated when we release new versions of the library</p>

13.3 .Net Libraries

.Net Libraries	
References	Description
Mscorlib	mscorlib.dll is a module associated with .NET Compact Framework from Microsoft Corporation. Non-system processes like mscorev.dll originate from software you installed on your system. Since most applications store data on your hard disk and in your system's registry, it is likely that your computer has suffered fragmentation and accumulated invalid entries which can affect your PC's performance.
System.Configuration	System.configuration.dll file is part of System.Configuration.dll product developed by Microsoft® .NET Framework. It is responsible for Microsoft Corporation functionality. This part is required for System.Configuration.dll to function. If you are planning to replace system.configuration.dll with a new version, make sure that you backup your old file. Replacing system.configuration.dll file with a different version can cause System.Configuration.dll to malfunction.
System.Core	System.core.dll file is part of .NET Framework product developed by Microsoft® .NET Framework. It is responsible for Microsoft Corporation functionality. This part is required for .NET Framework to function. If you are planning to replace system.core.dll with a new version, make sure that you backup your old file. Replacing the system.core.dll file with a different version can cause .NET Framework to malfunction.
System.Data	Provides access to classes that represent the ADO.NET architecture. ADO.NET lets you build components that efficiently manage data from multiple data sources.
System.Data.DataSetExtensions	System.Data.DataSetExtensions.dll' is a library module associated with the Microsoft® .NET framework, a product of Microsoft Corporation. The primary role of this particular DLL lies within data handling, specifically concerning Dataset and DataTable functionalities within .NET applications.

References	Description
System.Data.Linq	System.Data.Linq.dll is a library file associated with the Microsoft® .NET Framework, developed by Microsoft Corporation. Specifically, this dynamic link library (DLL) file pertains to LINQ to SQL, which is a component of the .NET Framework that provides a run-time infrastructure for managing relational data as objects.
System.Deployment	System.Deployment.dll is a Dynamic Link Library (DLL), designed as a repository for code, data, and resources among various programs. These DLL files enhance the efficiency of software applications and the operating system by providing shared access to a common set of functions. Instead of every application having its own copy of every function, common functions are kept in these DLL files so they can be shared among many applications. However, if a DLL file becomes corrupted or goes missing, it can cause issues with any program that relies on it.
DirectoryServices.AccountManagement	System.DirectoryServices.AccountManagement.dll is a Dynamic Link Library (DLL), designed as a repository for code, data, and resources among various programs. These DLL files enhance the efficiency of software applications and the operating system by providing shared access to a common set of functions. Instead of every application having its own copy of every function, common functions are kept in these DLL files so they can be shared among many applications. However, if a DLL file becomes corrupted or goes missing, it can cause issues with any program that relies on it.
System.Drawing	The System.Drawing.dll is a Dynamic Link Library (DLL) file that is part of the .NET Framework, a software development platform created by Microsoft. This particular DLL provides classes that enable developers to create graphical applications and manipulate images, fonts, colors, and other graphical elements. It serves as a wrapper around the Windows Graphics Device Interface (GDI) and GDI+, providing a more convenient and object-oriented way to handle graphics programming tasks in .NET applications. This library is commonly used in Windows Forms applications but can also be used in other types of .NET projects.
System.Runtime.Serialization	The System.Drawing.dll is a Dynamic Link Library (DLL) file that is part of the .NET Framework, a software development platform created by Microsoft. This particular DLL provides classes that enable developers to create graphical applications and manipulate images, fonts, colors, and other graphical elements. It serves as a wrapper around the Windows Graphics Device Interface (GDI) and GDI+, providing a more convenient and object-oriented way to handle graphics programming tasks in .NET applications. This library is commonly used in Windows Forms applications but can also be used in other types of .NET projects.
System.ServiceModel	The System.ServiceModel.dll is a key component of the Microsoft .NET Framework, specifically designed to support Windows Communication Foundation (WCF). WCF is a framework for building distributed, service-oriented applications that enable secure and reliable communication across different systems. This DLL provides the foundational classes, interfaces, and enumerations required for developing both WCF services and clients, encompassing functionalities like defining service contracts, operation contracts, and data contracts, as well as implementing various security protocols and transaction support.

References	Description
System.Web.Extensions	<p>The System.Web.Extensions.dll is a Dynamic Link Library (DLL) that is part of the Microsoft .NET Framework, specifically designed for web development features. This library primarily contains classes that support ASP.NET AJAX (Asynchronous JavaScript and XML) and ASP.NET Web Forms, extending the capabilities of the basic ASP.NET web framework.</p>
System.Web.Services	<p>System.Web.Services.Description.resources.dll is a Dynamic Link Library (DLL), designed as a repository for code, data, and resources among various programs. These DLL files enhance the efficiency of software applications and the operating system by providing shared access to a common set of functions. Instead of every application having its own copy of every function, common functions are kept in these DLL files so they can be shared among many applications. However, if a DLL file becomes corrupted or goes missing, it can cause issues with any program that relies on it.</p>
System.Windows.Forms	<p>The System.Windows.Forms.dll is a Dynamic Link Library (DLL) file that's part of the Microsoft .NET Framework and .NET Core/.NET 5+. This DLL is the backbone for Windows Forms, a graphical (GUI) class library that provides a platform to create desktop applications for the Windows operating system. Windows Forms allows for the creation of form-based applications with a wide range of controls like buttons, text boxes, labels, grids, and other UI elements.</p>
System.Xml	<p>System.Xml.XDocument.dll is a Dynamic Link Library (DLL), designed as a repository for code, data, and resources among various programs. These DLL files enhance the efficiency of software applications and the operating system by providing shared access to a common set of functions. Instead of every application having its own copy of every function, common functions are kept in these DLL files so they can be shared among many applications. However, if a DLL file becomes corrupted or goes missing, it can cause issues with any program that relies on it.</p>
System.Xml.Linq	<p>'System.Xml.Linq.dll' is a dynamic link library (DLL) included in the Microsoft® .NET Framework, a comprehensive platform used for developing a wide range of applications for Windows and other environments.</p>

14. App.Config file

14.1 Configurations

Name of Configuration	Details												
connectionStrings	<p>This section gives the details of the connection strings for data base used in this application Here it shows the 3 connection strings.</p> <table border="1"> <thead> <tr> <th>Connetion string</th><th>DB Server name</th><th>DB Name</th></tr> </thead> <tbody> <tr> <td>FundSystem.Properties.Settings.SAFI_DBConnectionString</td><td>basseterre-dv2</td><td>SAFI_DB</td></tr> <tr> <td>DRReporting.Properties.Settings.SAFI_DBConnectionString</td><td>basseterre-dv2</td><td>SAFI_DB</td></tr> <tr> <td>DRReporting.Properties.Settings.T24ReportingDBConnectionString</td><td>basseterre-dv2</td><td>T24ReportingDB</td></tr> </tbody> </table> <p>Below is the connectionstring from config file :</p> <pre><connectionStrings> <add name="FundSystem.Properties.Settings.SAFI_DBConnectionString" connectionString="Data Source=basseterre-dv2;Initial Catalog=SAFI_DB;Integrated Security=True" providerName="System.Data.SqlClient" /> <add name="DRReporting.Properties.Settings.SAFI_DBConnectionString" connectionString="Data Source=basseterre-dv2;Initial Catalog=SAFI_DB;Integrated Security=True" providerName="System.Data.SqlClient" /> <add name="DRReporting.Properties.Settings.T24ReportingDBConnectionString" connectionString="Data Source=basseterre-dv2;Initial Catalog=T24ReportingDB;Integrated Security=True" providerName="System.Data.SqlClient" /> </connectionStrings></pre>	Connetion string	DB Server name	DB Name	FundSystem.Properties.Settings.SAFI_DBConnectionString	basseterre-dv2	SAFI_DB	DRReporting.Properties.Settings.SAFI_DBConnectionString	basseterre-dv2	SAFI_DB	DRReporting.Properties.Settings.T24ReportingDBConnectionString	basseterre-dv2	T24ReportingDB
Connetion string	DB Server name	DB Name											
FundSystem.Properties.Settings.SAFI_DBConnectionString	basseterre-dv2	SAFI_DB											
DRReporting.Properties.Settings.SAFI_DBConnectionString	basseterre-dv2	SAFI_DB											
DRReporting.Properties.Settings.T24ReportingDBConnectionString	basseterre-dv2	T24ReportingDB											
Appsetting>> MainFolderPath	Folder path of the where the report files saved. Ex: c:\DRStatements\												
Appsetting>> T24IPAddress	Server name of Temenos 24. (value: darwin)												
Appsetting>> T24Port	Port name which use to connect the server (value: 7023)												
Appsetting>> T24Timeout	Timeout setting in seconds for connection to server(value: 60000)												
Appsetting>> EmailFrom	from Email setting (value: notificaciones@jmmb.com)												
Appsetting>> T24DefaultCompany , AFPCOMPANY	The setting for authentication details for the application.(value: D06000006)												
Appsetting>> UnitsEnquiry	The parameter value for Temenos 24 in REIT form to query to get the total number of units.												
Appsetting>> DateEnquiry	The parameter value for Temenos 24 in REIT form to query to get the current date info.												

system.serviceModel	<p>Client>> endpoint: This section is for the email webservice setup like URL, type of binding, binding configuration, what kind of contract is used and name of the web service etc.</p> <p>The Values are give below:</p> <p>Address:http://appsvr3-env1-dv:38080/SendEmailWeb/SendEMailWSService</p> <p>binding: basicHttpBinding bindingConfiguration: SendEmailWSPortBindin contract: EmailWS.SendEmailWS name: SendEmailWSPort</p>
---------------------	---