

## Project: **Whizzer 2018/115906**

### SRS - Software Requirements Specification

**Business Information Systems (Allevo) SRL**; VAT no: 6117169

Grant type: Romania Innovation EEA grants (ROM-EEA 1153)

October 2021

Business Information  
Systems (Allevo) SRL  
Sediu social:  
23 Coltei St, 030245  
Bucharest, Romania  
Sediu executiv:  
23C Calea Vitan, 031281  
Bucharest, Romania  
Tel / fax:  
+40212554577  
+40212554578  
+40212554579  
Website: [www.allevo.ro](http://www.allevo.ro)  
Capital social: 2.531.930 lei  
RC: J40/2067/94  
CIF: RO5258486  
SWIFT PIC: PTSAR0AA  
DUNS: 55-244-8078  
Certificat ISO 9001:2015



Certificat ISO/IEC 27001:2013



## Document Control

Title: Whizzer - SRS - Software Requirements Specification  
Project: **Whizzer 2018/115906**  
Version: 1.2  
Creation Date: Date: 5.03.2020

Author:



## Update history

\*A - Added M - Modified D - Deleted

Version	Date	A* M D	Short description	Authors	Contributors
1.0	05.03.2020	A	Original document		
1.1	30.04.2020	A	Requirements development		
1.2	30.04.2020	M	Requirements review		
1.3	30.04.2020	M	Requirements development – Standard system features		

Information  
(Allevo) SRL  
Address:  
Bulevardul St. 030245  
Bucuresti, Romania  
E-mail:  
alea.Vitan, 031281  
Bucuresti, Romania

2554577  
2554578  
2554579  
www.allevo.ro  
Phone: 2.531.930 lei  
067/94  
258486  
C: PTSARAAA  
+244-8078  
ISO 9001:2015



ISO/IEC 27001:2013



1.4	30.05.2020	M	Requirements development – External Interface Requirements
1.5	30.06.2020	M	Requirements development – Other Non-functional Requirements
1.6	30.07.2020	M	Requirements development - Accounts payable and accounts receivable - Flows
1.7	30.08.2020	M	Requirements development - Accounts payable and accounts receivable – Description

Information  
allevo) SRL  
t. 030245  
Romania  
iv:  
Vitan, 031281  
Romania  
4577  
4578  
4579  
w.allevo.ro  
l: 2.531.930 lei  
7/94  
486  
PTSAR0AA  
4-8078  
9001:2015



IEC 27001:2013

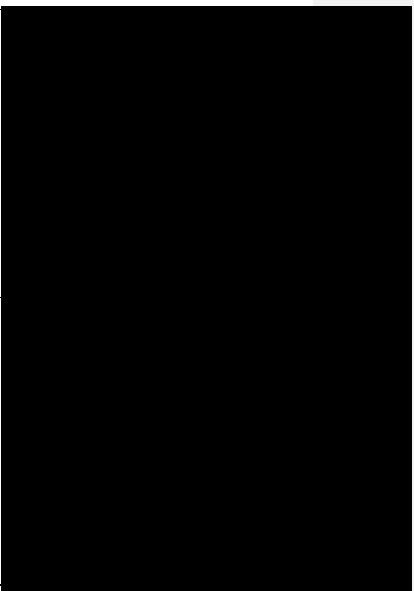


1.8	30.09.2020	M	Requirements development – Balance Sheet – General description
1.9	30.10.2020	M	Requirements development – Balance Sheet – Detailed Analysis
1.10	30.11.2020	M	Requirements development – Cash Reporting - General description
1.11	30.12.2020	M	Requirements development – Cash Reporting - Balance Sheet forecast

Information  
allevo) SRL  
al:  
i St, 030245  
st, Romania  
culiv:  
lea Vitan, 031281  
st, Romania  
  
554577  
554578  
554579  
www.allevo.ro  
cial: 2.531.930 lei  
067/94  
58486  
PTSAR0AA  
244-8078  
ISO 9001:2015



1.12	30.01.2021	M	Requirements development – Cash Reporting - Cash Flow forecast
1.13	30.02.2021	M	Requirements development – Common feature - Register Realised Balance Sheet (P&L)



Business Information  
Systems (Allevo) SRL  
Sediul social:  
23 Coltei St, 030245  
Bucharest, Romania  
Sediul executiv:  
23C Calea Vitan, 031281  
Bucharest, Romania  
Tel / fax:  
+40212554577  
+40212554578  
+40212554579  
Website: www.allevo.ro  
Capital social: 2.531.930 lei  
RC: J40/2067/94  
CIF: RO5258486  
SWIFT PIC: PTSAR0AA  
DUNS: 55-244-8078  
Certificat ISO 9001:2015



Certificat ISO/IEC 27001:2013



1	Introduction	7
1.1	PURPOSE	7
1.1.1	<i>Problem</i>	7
1.1.2	<i>Solution</i>	8
1.1.3	<i>Components</i>	8
1.1.4	<i>Target audience</i>	9
1.1.5	<i>Legal framework</i>	9
1.1.6	<i>Partner role – Bakken and Baeck</i>	9
1.1.7	<i>Performance and stress tests</i>	10
1.1.8	<i>Integration and deployment considerations</i>	10
1.1.9	<i>Requirements</i>	10
1.2	DOCUMENT CONVENTIONS	10
1.3	INTENDED AUDIENCE AND READING SUGGESTIONS	11
1.4	PROJECT SCOPE	12
1.5	REFERENCES	12
2	Overall Description	13
2.1	PRODUCT PERSPECTIVE	13
2.2	PRODUCT FEATURES	15
2.3	USER CLASSES AND CHARACTERISTICS	17
3	System Features	18
3.1.1	<i>Concepts</i>	18
3.2	STANDARD SYSTEM FEATURES	19
3.2.1	<i>Whizzer processing model</i>	20
3.2.2	<i>Standard components</i>	21
3.3	NEW SYSTEM FEATURES	23
3.3.1	<i>Accounts payable and accounts receivable</i>	23
3.3.2	<i>Balance Sheet</i>	41
3.3.3	<i>Cash Reporting</i>	45
3.3.4	<i>Register Realised Balance Sheet (P&amp;L) - Common feature</i>	51
4	External Interface Requirements	53
5	Other Non-functional Requirements	53
5.1	SOFTWARE AS A SERVICE REQUIREMENTS	53
5.1.1	<i>SAAS01 - Architecture requirements - Multi-tenancy</i>	53
5.1.2	<i>SAAS02 - Update/ upgrade policy</i>	55
5.1.3	<i>SAAS03 - Configurability</i>	55
5.1.4	<i>SAAS04 - Security and privacy requirements</i>	56
5.1.5	<i>SAAS05 - Centralized administration</i>	56
5.1.6	<i>SAAS06 - Predictable cost model</i>	56
5.1.7	<i>SAAS07 - Performance requirements</i>	57
6	Other Requirements	58
6.1	APPENDIX A: GLOSSARY	58

Business Information  
Systems (Allevo) SRL  
Sediu social:  
23 Coltei St, 030245  
Bucharest, Romania  
Sediu executiv:  
23C Calea Vitan, 031281  
Bucharest, Romania  
Tel / fax:  
+40212554577  
+40212554578  
+40212554579  
Website: www.allevo.ro  
Capital social: 2.531.930 lei  
RC: J40/2067/94  
CIF: RO5258486  
SWIFT PIC: PTSAR0AA  
DUNS: 55-244-8078  
Certificat ISO 9001:2015



Certificat ISO/IEC 27001:2013



## 1 Introduction

### 1.1 Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

#### 1.1.1 Problem

Financial inclusion for SME's in terms of access to finance and control of personal data and over administrative processes.

#### Connectivity

SMEs are not well connected to their partner banks. They each login to the internet banking interfaces made available by the banks where they hold bank accounts. They download or receive bank statements for each account. Most SMEs own several accounts, in multiple currencies (most common RON, EUR, USD), at more than one bank.

Disconnected data, data copied from one source to the other. No single source of truth.

#### Formats

Bank statements differ from bank to bank. Information about the history of an account is now available via APIs to fintechs authorized as AISP's under PSD2.

APIs published by banks follow a similar set of standards (Berlin Group etc). However each bank has implemented its own AISP onboarding mechanism.

A standard for electronic invoices is not being widely used. Formats differ across EU countries.

#### Financing

SMEs in need for a bank loan need to create a physical dossier and take it to the bank to hopefully get a loan collateralized with approved, not yet paid, invoices.

#### Cashflow

Cashflow reporting and management is difficult for SMEs who do not have a CFO on board. A cashflow reporting tool would show if the company will likely need cash in the upcoming period of time.

Invoices issued to customers usually have long payment deadlines. This creates a lack of cash for delivered services or products. Factoring services available on the market are slow and not really available to SMEs in need of a short source of cash.

SMEs are faced with late payments, with direct impact on cashflow.

#### Reconciliation

Business Information  
Systems (Allevio) SRL  
Sediu social:  
23 Coltei St, 030245  
Bucharest, Romania  
Sediu executiv:  
23C Calea Vitan, 031281  
Bucharest, Romania  
Tel / fax:  
+40212554577  
+40212554578  
+40212554579  
Website: www.allevio.ro  
Capital social: 2.531.930 lei  
RC: J40/2067/94  
CIF: RO5258486  
SWIFT PIC: PTSAR0AA  
DUNS: 55-244-8078  
Certificat ISO 9001:2015



Certificat ISO/IEC 27001:2013



There is lack of automation of matching the debt of a customer (e.g. invoice issued) with an incoming payment. It is common practice that the oldest debt is settled first.

Due amounts and incoming payments may differ, as customers can make one payment for multiple invoices. Settlement is mostly manual.

#### 1.1.2 Solution

An open source software solution developed by Allevo that offers **financial operations as a service** to SMEs, to help them automate and centralize common financial flows:

1. **Balance Sheet**
2. **Salary Payments**
3. **Invoicing**
4. **Money Flow Automation**
5. **Accounts payable** (amounts that a company owes to its suppliers) **and Accounts receivable** (amounts owed to a company by its customers).
6. **Cash Reporting.**

These 6 functionalities have been used when pricing the solution and making financial projections for Whizzer.

Data is created only once and all stakeholders and applications use this same data, without re-entering it manually, nor via usual export and import working procedures. This ensures consistency and accuracy of the data that is being processed.

Open source GPL v3 distribution via finftp.org and GitHub.

The customers get financial insights for predictive analysis, cash management options, cost saving. Harmonized communication over a well-built open banking layer is needed.

Matching payments and invoices:

- 1-1 for match on amount, currency, customer
- Suggested: oldest debt with incoming unmatched amount

#### 1.1.3 Components

**FinTP-Connect** offers a communications layer that manages requests coming from TPPs and responses from the bank's side. Allevo extends FinTP-Connect and builds an easy to use and cost effective solution that offers a single window for balance sheet, salary, invoicing, and money flow automation, accounts payable and receivable capabilities. This results in an **end-to-end integration of open source applications that provide this set of functionalities, with input at the source**. FinTP-Connect offers an open banking infrastructure that keeps in mind provisions of GDPR and puts the customer in control of their data (consent management). The component used is FinTPc-API, which ensures interfacing between data exposed by banks via Open Banking APIs and FinTPc functionalities, which centralize financial operations of a corporate.

**FinTPc** (the TOSS project) for basic functionality: duplicate detection, transaction validation, routing rules, user roles etc.

Business Information  
Systems (Allevo) SRL  
Sediul social:  
23 Coltei St, 030245  
Bucharest, Romania  
Sediul executiv:  
23C Calea Vitan, 031281  
Bucharest, Romania  
Tel / fax:  
+40212554577  
+40212554578  
+40212554579  
Website: www.allevo.ro  
Capital social: 2.531.930 lei  
RC: J40/2067/94  
CIF: RO5258486  
SWIFT PIC: PTSAR0AA  
DUNS: 55-244-8078  
Certificat ISO 9001:2015



Certificat ISO/IEC 27001:2013





**Whizzer** is a repackaging of the extended FinTPc and FinTP-Connect ready to be deployed in the cloud. The solution delivers financial operations as a service to SMEs. This allows SMEs to achieve a lean internal infrastructure, only using the service that sits in the cloud.

Whizzer extends the functionalities of FinTPc and FinTP-Connect, by providing multi-tenant capabilities necessary for delivering Software as a Service and financial reporting which are easy to understand and use by SME's Administrators without a financial background.

Whizzer uses the communication layer offered by FinTP-Connect to leverage data insights a bank holds in its systems.

Whizzer automates existing processes for financial transactions: salaries, advance payments, invoice payments, incoming payments. It can show the SME's financial position, do reconciliation services (e.g. issued invoices and incoming payments).

#### 1.1.4 Target audience

The entity that uses Whizzer to offer SMEs operations as a service can be Allevo or a third party: bank, clearing house, fintech.

The end users of the service can be: small and medium sized companies, financial departments of corporates.

Geography: European Union.

#### 1.1.5 Legal framework

This entity that deploys Whizzer and offers its functionalities as a service needs to be authorized as **AISP** (*Account Information Service Provider* as defined under PSD2) in order to access data available in a EU based bank and use it to consolidate financial information. Control over access to accounts and historic data is strictly managed by the SME.

The local legal framework that allows the Whizzer solution for SMEs to run is the **148/2012 Law**, which regulates the registration of commercial operations using electronic bills.

GDPR regulation is applicable for retrieving, storing, processing and sharing personal data.

Best practices for security of critical financial systems will be implemented. Example: two factor authentication, encrypted data, complex user names and passwords, protection against cyberattacks.

#### 1.1.6 Partner role – Bakken and Baeck

Bakken and Baeck develops AI capability resulting into a software component, tested and integrated with Allevo's software component. UX, design and front-end for AI results.

Bakken and Baeck builds a machine learning model for predictive analytics, using natural language generation models (robot journalism) and contributing to UX/design/front-end for presenting predictive analytics/AI results. This results in new service propositions combining predictive analytics, artificial intelligence, and financing to enhance consumer and business offerings.

Business Information  
Systems (Allevo) SRL  
Sediu social:  
23 Coltei St, 030245  
Bucharest, Romania  
Sediu executiv:  
23C Calea Vitan, 031281  
Bucharest, Romania  
Tel / fax:  
+40212554577  
+40212554578  
+40212554579  
Website: www.allevo.ro  
Capital social: 2.531.930 lei  
RC: J40/2067/94  
CIF: RO5258486  
SWIFT PIC: PTSAR0AA  
DUNS: 55-244-8078  
Certificat ISO 9001:2015



Certificat ISO/IEC 27001:2013



The UI will contain a dashboard, with reports that understand the customers and anticipate their needs.

#### 1.1.7 Performance and stress tests

Simulation of up to 100 SMEs connecting at the same time to consume financial operations offered as a service.

#### 1.1.8 Integration and deployment considerations

Allevo's environment will include a setup of the infrastructure required to run the software solution.

The tech teams working of the project will create a shared software repository where both Bakken and Baeck's and Allevo's teams contribute.

The component developed by Bakken and Baeck is developed independently and is integrated into the final solution before end of 2020.

Requirements, architecture and design documents are published into the open source space at the end of the project – finfp.org and GitHub.

#### 1.1.9 Requirements

**Business area:** cash reporting, cashflow predictions, interoperability with bank digital channels, balance sheet reporting, salary reporting, invoice reporting, accounts payable and receivable, reconciliation between invoices and payments.

**Operations area:** processing flows for financial instruments (credit transfer, direct debit, debit instruments and securities), multicurrency and operational risk containment functionalities (accounting reconciliation, duplicate detection, validation, filtering).

**IT area:** data quality, end-to-end persistent transactions, system resilience, native ISO 20022 support, flow configuration, API-REST interfaces, transaction history etc.

## 1.2 Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

Below you can find the name conventions used in this document, scenarios and functionalities starting with S or F:

- SRECONnn – reconciliation scenario number nn, mandatory
- FRECONnn – reconciliation functionality number nn, mandatory
- SBSnn – Balance Sheet scenario number nn, mandatory
- FBSnn – Balance Sheet functionality number nn, mandatory
- SCASHREPnn – Cash Reporting scenario number nn, mandatory
- FCASHREPnn – Cash Reporting functionality number nn, mandatory

Business Information  
Systems (Allevo) SRL  
Sediul social:  
23 Colteii St, 030245  
Bucharest, Romania  
Sediul executiv:  
23C Calea Vitan, 031281  
Bucharest, Romania  
Tel / fax:  
+40212554577  
+40212554578  
+40212554579  
Website: www.allevo.ro  
Capital social: 2.531.930 lei  
RC: J40/2067/94  
CIF: RO5258486  
SWIFT PIC: PTSAR0AA  
DUNS: 55-244-8078  
Certificat ISO 9001:2015



Certificat ISO/IEC 27001:2013



- SAASnn – Software as a service requirements

### 1.3 Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

This document is addressed the team involved in architecture, design, development, test and deployment:

- Business Analysts
- Solution Designers
- Software Architects
- Software Developers
- Testers
- Deployment Engineers
- Support Personnel
- Project Managers

This document is an artefact used to design, develop, test and document the project.

Business Information  
Systems (Allevo) SRL  
Sediu social:  
23 Coltei St, 030245  
Bucharest, Romania  
Sediu executiv:  
23C Calea Vitan, 031281  
Bucharest, Romania  
Tel / fax:  
+40212554577  
+40212554578  
+40212554579  
Website: www.allevo.ro  
Capital social: 2.531.930 lei  
RC: J40/2067/94  
CIF: RO5258486  
SWIFT PIC: PTSAR0AA  
DUNS: 55-244-8078  
Certificat ISO 9001:2015



Certificat ISO/IEC 27001:2013



## 1.4 Project Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here. An SRS that specifies the next release of an evolving product should contain its own scope statement as a subset of the long-term strategic product vision.>

**Whizzer** is the cloud deployment of a financial management solution, composed of FinTPc, FinTP-Connect and new functionalities developed on top. It enables the processing of invoices sent and received by a company, the processing of bank statements, manual and automatic matching of invoices with payments, cash reporting, cashflow predictions, draft balance sheet reporting, salary reporting, invoice reporting, integration and money flow automation, accounts payable (amounts that a company owes to its suppliers) and receivable (amounts owed to a company by its customers).

The connection between the company and their partner banks via API-REST follows the Berlin Group Next Gen standard, widely used by the banking community across the EU. The connectivity is ensured by FinTP-Connect and it provides:

- The connection via Open Banking mechanisms, using the OAuth2 authentication security protocol. API interfaces exposed by banks can be accessed to fetch transaction history data
- Real-time statistics and reports using data retrieved through API calls, providing an overall picture of a company's financial position
- Cash management reporting
- Aggregation of data available at partner banks.

## 1.5 References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

Whizzer project documentation:

- Approved business plan: *07\_business-plan-Allevo v34.docx*
- SWIFT Standards MT:  
[https://www2.swift.com/knowledgecentre/rest/v1/publications/mfvr\\_20191220/1\\_0/MFVR2020.pdf?logDownload=true](https://www2.swift.com/knowledgecentre/rest/v1/publications/mfvr_20191220/1_0/MFVR2020.pdf?logDownload=true)
- SWIFT Standards MX:

Business Information  
Systems (Allevo) SRL  
Sediul social:  
23 Colteii St, 030245  
Bucharest, Romania  
Sediul executiv:  
23C Calea Vitan, 031281  
Bucharest, Romania  
Tel / fax:  
+40212554577  
+40212554578  
+40212554579  
Website: [www.allevo.ro](http://www.allevo.ro)  
Capital social: 2.531.930 lei  
RC: J40/2067/94  
CIF: RO5258486  
SWIFT PIC: PTSAR0AA  
DUNS: 55-244-8078  
Certificat ISO 9001:2015



Certificat ISO/IEC 27001:2013



- [https://www2.swift.com/knowledgecentre/rest/v1/publications/stdsmx\\_pay\\_init\\_mdrs/2.0/SR2019\\_MX\\_PaymentsInitiation\\_MDR1\\_Standards.pdf?logDownload=true](https://www2.swift.com/knowledgecentre/rest/v1/publications/stdsmx_pay_init_mdrs/2.0/SR2019_MX_PaymentsInitiation_MDR1_Standards.pdf?logDownload=true)
- [https://www2.swift.com/knowledgecentre/rest/v1/publications/stdsmx\\_pay\\_init\\_mdrs/2.0/SR2019\\_MX\\_PaymentsInitiation\\_MDR2\\_Standards.pdf?logDownload=true](https://www2.swift.com/knowledgecentre/rest/v1/publications/stdsmx_pay_init_mdrs/2.0/SR2019_MX_PaymentsInitiation_MDR2_Standards.pdf?logDownload=true)

## 2 Overall Description

### 2.1 Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

A typical SME has accounts open with at least one commercial bank – typically local currency, EUR, USD.

Payment type such as salary can be fetched from internal applications.

Salaries are paid from a local currency account open at one or more commercial banks.

Invoices are usually paid from accounts holding the currency to be paid (accounts payable). Invoices issued to customers are usually paid in the account contractually agreed (accounts receivable) – preferably same currency, to avoid exchange rates.

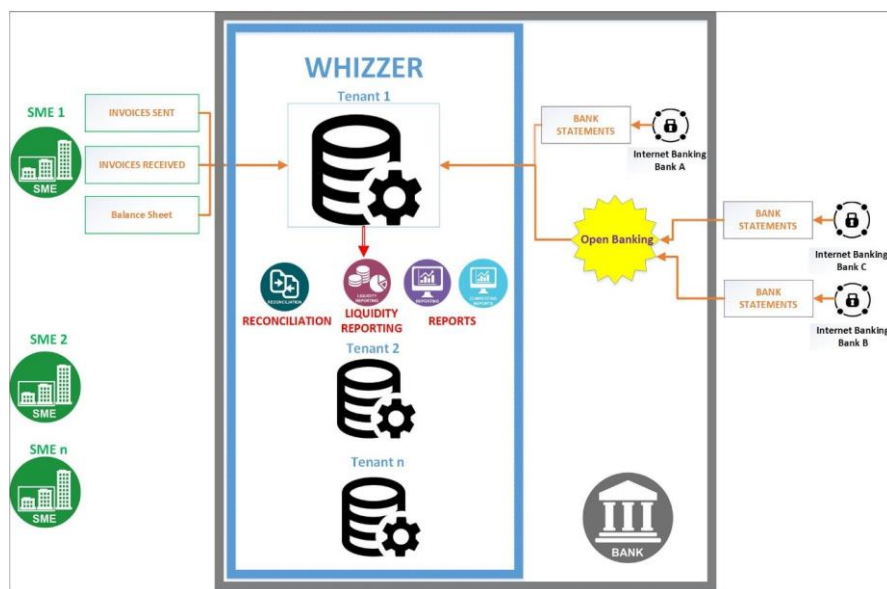
An SME connects to the internet banking interfaces of the banks to retrieve bank statements for each account and each currency. Frequency: ideally as often as possible. Real time data is most valuable.

Business Information  
Systems (Allevo) SRL  
Sediu social:  
23 Coltei St, 030245  
Bucharest, Romania  
Sediu executiv:  
23C Calea Vitan, 031281  
Bucharest, Romania  
Tel / fax:  
+40212554577  
+40212554578  
+40212554579  
Website: www.allevo.ro  
Capital social: 2.531.930 lei  
RC: J40/2067/94  
CIF: RO5258486  
SWIFT PIC: PTSAR0AA  
DUNS: 55-244-8078  
Certificat ISO 9001:2015



Certificat ISO/IEC 27001:2013





The key benefits provided by **Whizzer** are:

- Consolidation of all financial transactions of the company
- Streamlined operations by:
  - integrating payments from other applications
  - flow automation
  - automatic generation of consolidated reports
  - risk containment of duplicate or invalid payments
- Reduced fraud-related risks in connection with bank interface access
- Reports and statistics.

Business Information  
Systems (Allevo) SRL  
Sediu social:  
23 Coltei St, 030245  
Bucharest, Romania  
Sediu executiv:  
23C Calea Vitan, 031281  
Bucharest, Romania  
Tel / fax:  
+40212554577  
+40212554578  
+40212554579  
Website: [www.allevo.ro](http://www.allevo.ro)  
Capital social: 2.531.930 lei  
RC: J40/2067/94  
CIF: RO5258486  
SWIFT PIC: PTSAR0AA  
DUNS: 55-244-8078  
Certificat ISO 9001:2015



Certificat ISO/IEC 27001:2013



## 2.2 Product Features

<Summarize the major features the product contains or the significant functions that it performs or lets the user perform. Details will be provided in Section 3, so only a high level summary is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or a class diagram, is often effective.>

The main building blocks provided by **Whizzer** are:

- **Administration**: provides capabilities for defining **Users and Rights**, **Application Management** and configuration (including routing logic, parameters, validations) and **Business Management** configuration of various lists (partner banks, entities and accounts, partners and accounts, black lists and so on).
- **Application Interfaces**: represent the external interfaces to other applications, for importing financial transactions or sending financial transactions in various structured formats for the corresponding business flows (**payments and transaction reports**)
- **Reporting**: generating real-time statistical outcomes and reports for specific business or operational needs
- **Audit Trails**: ensure the basis for tracking workflows and exceptions or recording user actions within the application, necessary for performing investigations for exceptions or consistency checks

The diagram below presents the main features described above. The mandatory features for any specific configuration / deployment are written in **bold**, while features configured optionally are written in light grey.

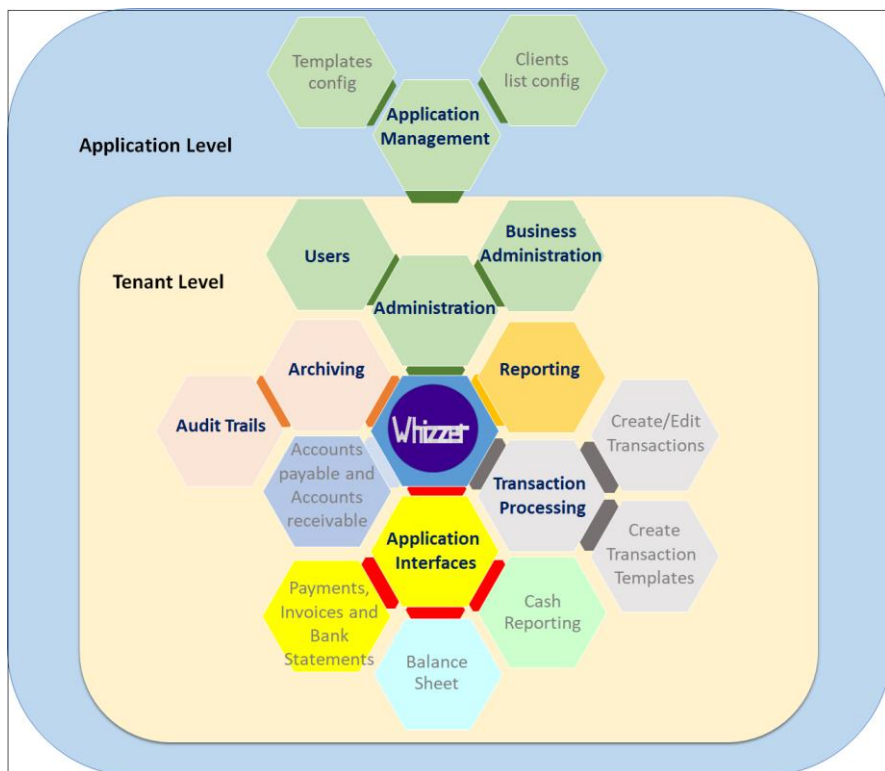
Commented [IG1]: To update

Business Information  
Systems (Allevo) SRL  
Sediu social:  
23 Coltei St, 030245  
Bucharest, Romania  
Sediu executiv:  
23C Calea Vitan, 031281  
Bucharest, Romania  
Tel / fax:  
+40212554577  
+40212554578  
+40212554579  
Website: www.allevo.ro  
Capital social: 2.531.930 lei  
RC: J40/2067/94  
CIF: RO5258486  
SWIFT PIC: PTSAR0AA  
DUNS: 55-244-8078  
Certificat ISO 9001:2015



Certificat ISO/IEC 27001:2013





Business Information  
Systems (Allevo) SRL  
Sediul social:  
23 Coltei St, 030245  
Bucharest, Romania  
Sediul executiv:  
23C Calea Vitan, 031281  
Bucharest, Romania  
Tel / fax:  
+40212554577  
+40212554578  
+40212554579  
Website: www.allevo.ro  
Capital social: 2.531.930 lei  
RC: J40/2067/94  
CIF: RO5258486  
SWIFT PIC: PTSAR0AA  
DUNS: 55-244-8078  
Certificat ISO 9001:2015



Certificat ISO/IEC 27001:2013





## 2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the favored user classes from those who are less important to satisfy.>

**FinTPc Interface** is used by the following user profiles:

- *Security Administrators* – ensure user management and access rights
- *Application Administrators* – ensure administration of FinTPc, providing support for managing specific business requests on the application configuration
- *Support and Monitoring Personnel* – provide continuous monitoring of FinTPc and initiate investigations when errors occur
- *Operators* – perform business operations provided by FinTPc, like actions on financial transactions (create, modify, reject, authorize), reconciliation, generate various reports and so on
- *Supervisors* – oversee the overall result of business processing, like generating reports on financial transactions

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

<Don't really say "System Feature 1." State the feature name in just a few words.>

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

Business Information  
Systems (Allevo) SRL  
Sediu social:  
23 Coltei St, 030245  
Bucharest, Romania  
Sediu executiv:  
23C Calea Vitan, 031281  
Bucharest, Romania  
Tel / fax:  
+40212554577  
+40212554578  
+40212554579  
Website: www.allevo.ro  
Capital social: 2.531.930 lei  
RC: J40/2067/94  
CIF: RO5258486  
SWIFT PIC: PTSAR0AA  
DUNS: 55-244-8078  
Certificat ISO 9001:2015



Certificat ISO/IEC 27001:2013



### 3 System Features

#### 3.1.1 Concepts

Nr crt.	Concept	Definition
1.	Whizzer	Whizzer is a new version or upgrade of FinTPc. Main components can be split into: <ul style="list-style-type: none"> <li>existing FinTPc features, pre-configured in a standardized way for easy onboarding</li> <li>new features related to management of financial operations of SMEs</li> <li>features from FinTP-Connect, that enable API integration with other systems</li> <li>architectural changes and non-functional features necessary to deploy and host FinTPc in cloud / data center, in a multi-tenancy mode that enables Software as a Service distribution</li> </ul>
2.	Application Interface	Represents the integration between Whizzer and other applications. It is defined by communication protocol and structured data format.  Application interface can be internal and external.  Internal application interface represents the integration between Whizzer and corporate's core applications (ERP, accounting, human resources).  External application interface represents the integration between Whizzer and applications that ensure connection to business partners.
3.	Financial Message	Represents a structured data set exchanged on an application interface. A message contains one or more financial transactions.
4.	Business Area	Is defined as a logical classification for different categories of financial transactions, such as : <ul style="list-style-type: none"> <li>Payments</li> <li>Invoices</li> <li>Statements</li> </ul> Classification is based on financial transactions that share a set of related information that meet a business purpose.  Whizzer design allows definition of new business areas.

Business Information  
Systems (Allevo) SRL  
Sediu social:  
23 Coltei St, 030245  
Bucharest, Romania  
Sediu executiv:  
23C Calea Vitan, 031281  
Bucharest, Romania  
Tel / fax:  
+40212554577  
+40212554578  
+40212554579  
Website: www.allevo.ro  
Capital social: 2.531.930 lei  
RC: J40/2067/94  
CF: RO5258486  
SWIFT PIC: PTSAR0AA  
DUNS: 55-244-8078  
Certificat ISO 9001:2015



Certificat ISO/IEC 27001:2013



Nr crt.	Concept	Definition
5.	Financial Transaction	Represents the information exchanged between counterparties part of a larger financial flow. It represents an element that defines an individual financial operation.  Financial transactions are logically grouped in business areas based on their predefined business purpose.  Financial transactions can also be referred to as transactions.
6.	Status	Represents a property of a financial transaction that indicates its processing phase.
7.	Action	Is a processing operation on financial transactions either: <ul style="list-style-type: none"> <li>• automatically, based on predefined processing configuration, referred as Routing Rule (RR), or</li> <li>• manually, based on user decisions, referred as User Action (UA).</li> </ul>
8.	Event	Represents auditing trails related to processing of financial transactions and actions performed in the user interface.
9.	Queue	Represents a number of financial transactions grouped together with the purpose of applying the same set of actions.
10.	Entity	Represents an organization with a number of bank accounts associated. A bank account can correspond only to one entity.  Entities in the standard configuration are internal.  Internal entities are the organizations for which Whizzer manages the financial flows.  Financial transactions from Payments Business Area are issued from internal entities accounts to other internal entities accounts.  Statements can be received and processed only for internal entities accounts.
11.	User Right	Represents access granted to users in the user interface for actions and data enquiry.
12.	Reconciliation	The matching process between financial transactions based on predefined criteria

Business Information  
Systems (Allevo) SRL  
Sediu social:  
23 Coltei St, 030245  
Bucharest, Romania  
Sediu executiv:  
23C Calea Vitan, 031281  
Bucharest, Romania  
Tel / fax:  
+40212554577  
+40212554578  
+40212554579  
Website: www.allevo.ro  
Capital social: 2.531.930 lei  
RC: J40/2067/94  
CF: RO5258486  
SWIFT PIC: PTSAR0AA  
BUNS: 55-244-8078  
Certificat ISO 9001:2015



Certificat ISO/IEC 27001:2013



### 3.2 Standard system features

There are some system features that enable the standard configuration of the application.

### 3.2.1 Whizzer processing model

**Whizzer** is the new version of FinTPc, a financial middleware allowing business users to define and control the process of transferring financial transaction between various corporate systems.

*Reports and statistics* provides needed insights on processed transactions for business operations.

*Transaction processing* develops according to the following stages:

1. Fetch financial messages from Application Interfaces
2. Format, Store and Route financial transactions
3. Publish financial messages to Application Interfaces

1. *Fetch stage* assumes the existence of predefined Application Interface. Whizzer can interface with Database Systems, File Systems, Messaging Systems and Web Services.

The purpose of this stage is to reliably interface with other financial application and perform preparation steps for next processing stage.

During this stage the main operations are:

- a. Backup fetched messages
- b. Split messages by transactions
- c. Transform transactions in easily parsing format
- d. Augment transaction with data collected by other application calls (WS, DB)
- e. Transfer transactions to the core application processor

2. *Format, Store and Route stage* is performed by both the core application processor and user interaction. The purpose of this stage is to offer processing control to the users and deliver business capabilities over processed transactions.

*Format and Parse.* By formatting, the financial transaction content is interpreted and transformed into an the native format of the application. Whizzer is required to fully understand financial transactions in order to deliver all the business capabilities. Interpreted information is stored in an easily accessible way. After being persisted, transactions are available at any moment for reporting.

*Route.* By routing, the users strictly control transactions transfers between application interfaces. The control is applied using:

- queues where transactions are segregated into bundles for each business operation need
- queue's predefined actions in order to distinctly operate in each queue

*So segregating transactions and operations, subject to the business operation requirement, is the main purpose of routing capability.*

Transactions delivered to the core processor are gathered in entry queues according to custom configuration of fetching stage. Actions performed by core application processor

Business Information  
Systems (Allevo) SRL  
Sediu social:  
23 Coltei St, 030245  
Bucharest, Romania  
Sediu executiv:  
23C Calea Vitan, 031281  
Bucharest, Romania  
Tel / fax:  
+40212554577  
+40212554578  
+40212554579  
Website: www.allevo.ro  
Capital social: 2.531.930 lei  
RC: J40/2067/94  
CIF: RO5258486  
SWIFT PIC: PTSAR0AA  
DUNS: 55-244-8078  
Certificat ISO 9001:2015



Certificat ISO/IEC 27001:2013



called *routing rules* take control and operate on delivered transactions. Routing rules are user defined, but sequentially executed by the core application processor. *Format and Store* is a mandatory routing rule on entry queues.

During routing, transactions can be held in queues for user audit and routing decisions. User decisions on held transactions are performed by **user actions**.

Transactions can be moved from entry queues to application predefined intermediate queues that deliver features like *duplicate detection* (standard configuration implies all fields duplicate detection). Other user defined intermediate queues are possible in order to perform more grained processing segregation on transactions.

In order to have transactions published on the application interface, they need to be previously collected in exit queues. Only exit queues have the capability to push transactions to the publishing stage processing. Each receiving application interface is bound to a publishing stage and usually requires a specific receiving transaction format. Transforming transactions to required format need to be among last actions performed on exit queues.

After being pushed to publishing stage, transactions are no longer available for other operations than correlation with related messages. Published transactions are still available for reporting.

To provide a higher quick overview of processed transactions, there is a status reported on each transaction. Status changes according to relevant actions performed during processing.

Modifying transaction information during routing can conflict with routing decisions made on previous values of transaction information. Such conflicts break the reliable and predictable transfer of financial transactions. Whizzer avoids these conflicts by providing limited and strictly controlled transactions modifications mechanisms.

**3. Publishing stage.** The purpose of this stage is to reliably interface with other financial applications and perform preparation steps according to application interface requirements

During this stage the main operations are:

- Group transactions into batches (if required by application interface)
- Sign messages (if required by application interface)
- Transfer message to existing predefined interfaces according to interface protocol and digital data format

### 3.2.2 Standard components

The following FinTPc components are included in the standard configuration:

#### 3.2.2.1 Data structures

This component covers the standard behavior: financial message structures for each type of transaction (payments, invoices, statements) and possible interfaces.

Business Information  
Systems (Allevo) SRL  
Sediu social:  
23 Coltei St, 030245  
Bucharest, Romania  
Sediu executiv:  
23C Calea Vitan, 031281  
Bucharest, Romania  
Tel / fax:  
+40212554577  
+40212554578  
+40212554579  
Website: www.allevo.ro  
Capital social: 2.531.930 lei  
RC: J40/2067/94  
CIF: RO5258486  
SWIFT PIC: PTSAR0AA  
DUNS: 55-244-8078  
Certificat ISO 9001:2015



Certificat ISO/IEC 27001:2013



### 3.2.2.2 Administration

Administration also covers standard features, in terms of User Management, Application configurations and Business management.

Additional architecture requirements in extension of the standard configuration are described in 5.1.1 Architecture requirements - Multi-tenancy.

### 3.2.2.3 Application interfaces

Both **internal** and **external** application interface methods are used in order to support transactions from all business areas (payments, invoices and statements) through multiple communication protocols and structured data formats: files, database tables and APIs.

Standard configuration implies fetching bank statements through APIs.

### 3.2.2.4 Transaction processing

Standard scenarios for transaction processing are configured for all business areas (payments, invoices, statements), to support **fetching**, **processing** and **publishing** financial messages.

Processing financial messages implies formatting and splitting them into single financial transactions according to scenario they belong to.

Publishing transactions is not a part of standard configuration.

The standard configuration of the application manages Salary Payments, that are fetched from internal applications

Transactions statuses are also covered in order to keep track of the flow.

### 3.2.2.5 Reporting

General transaction reports are delivered for each business area: payments business area, invoices business area and statements business area.

Events report is also delivered, in order to capture all recorded events in the application.

### 3.2.2.6 Audit trails

The standard configuration collects and records logs and events such as errors, informational events and warnings needed for further investigation.

Events can be traced and investigated running [Events Report, mentioned at 3.2.2.5 Reporting.](#)

Business Information  
Systems (Allevo) SRL  
Sediu social:  
23 Coltei St, 030245  
Bucharest, Romania  
Sediu executiv:  
23C Calea Vitan, 031281  
Bucharest, Romania  
Tel / fax:  
+40212554577  
+40212554578  
+40212554579  
Website: www.allevo.ro  
Capital social: 2.531.930 lei  
RC: J40/2067/94  
CIF: RO5258486  
SWIFT PIC: PTSAR0AA  
DUNS: 55-244-8078  
Certificat ISO 9001:2015



Certificat ISO/IEC 27001:2013

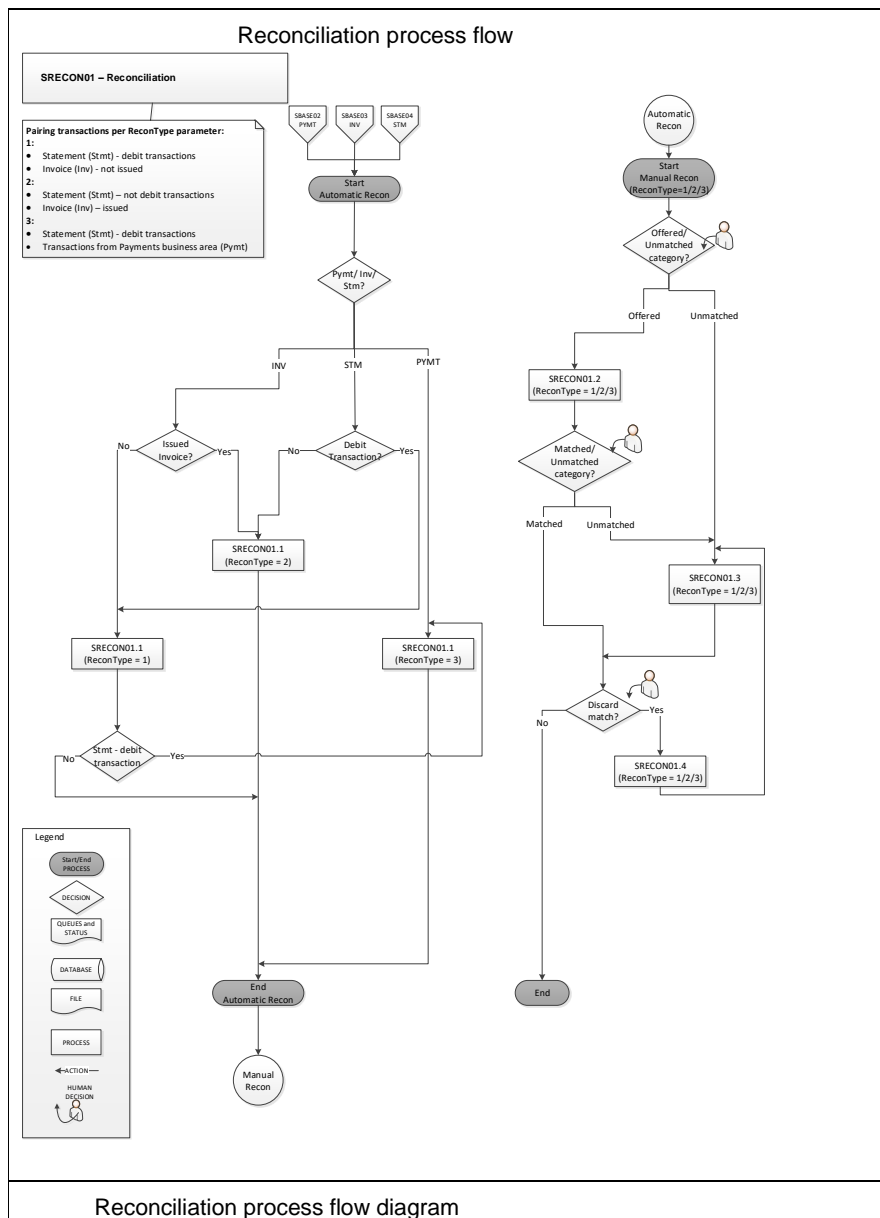


### 3.3 New system features

#### 3.3.1 Accounts payable and accounts receivable

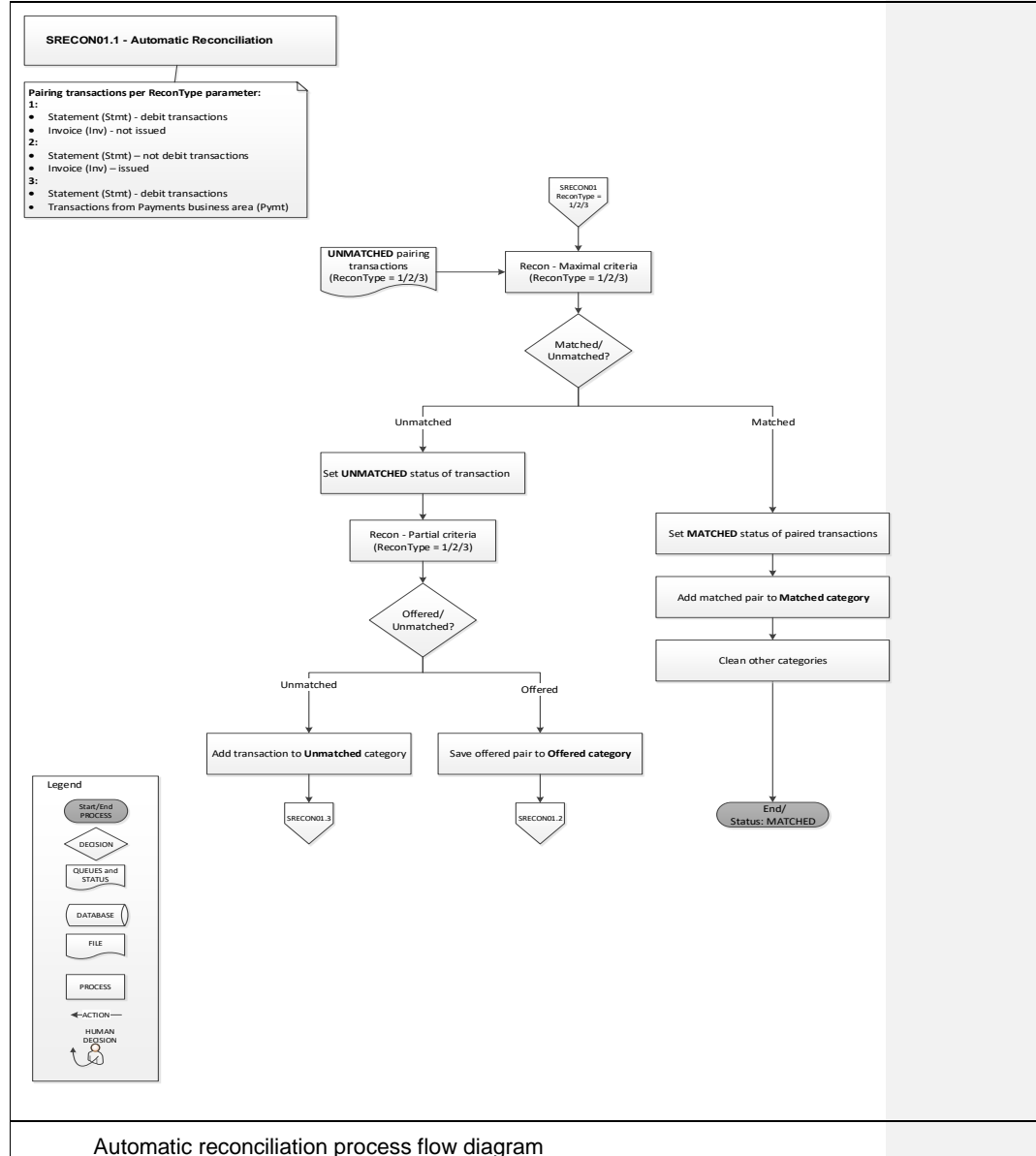
Accounts payable and accounts receivable can be used for the reconciliation process that can either be automatic or manual and it implies matching two or more financial transactions from different Business Areas, with the scope to provide consistency and accuracy in financial accounts.

##### *3.3.1.1 SRECON01 - Reconciliation process*

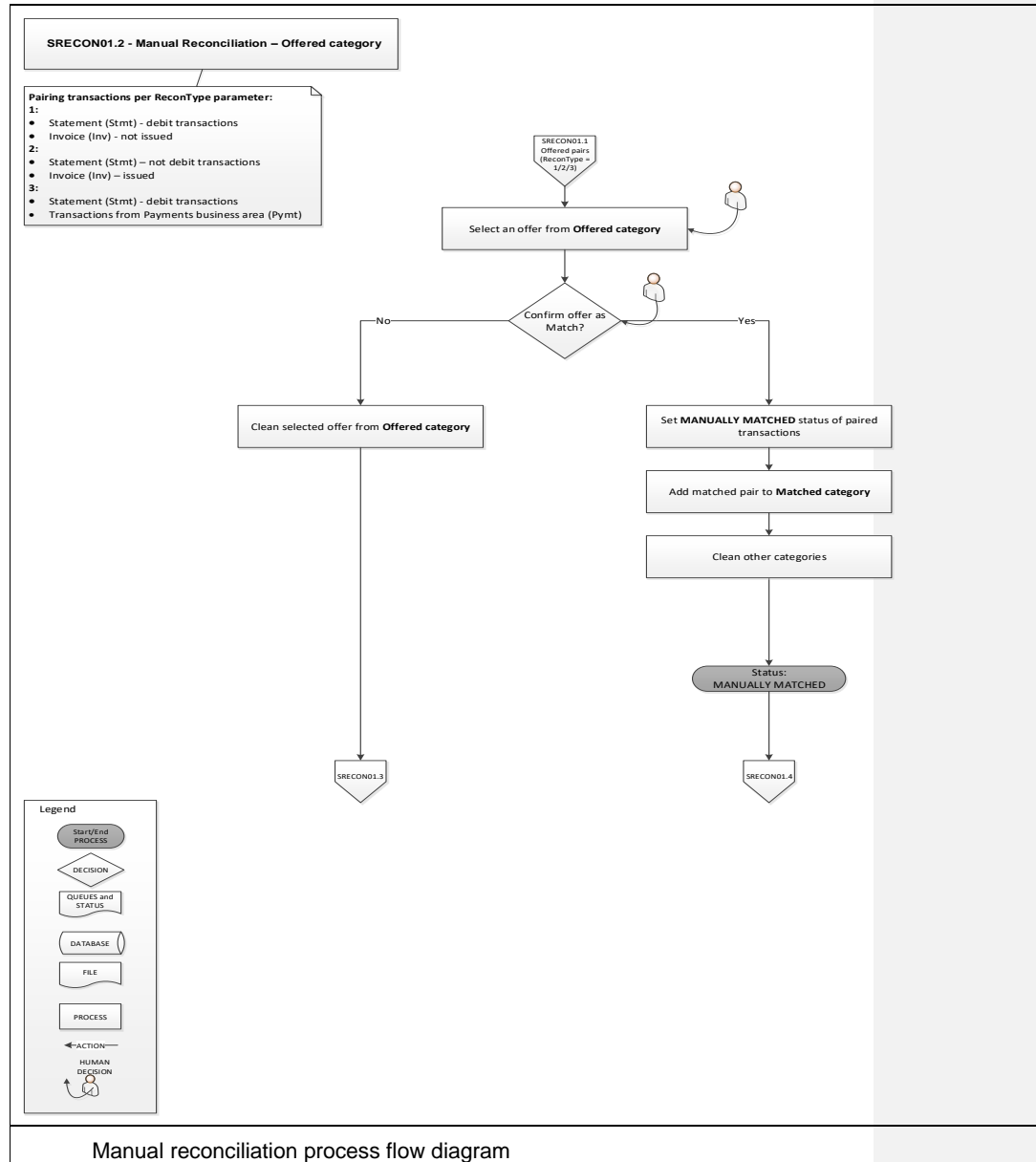




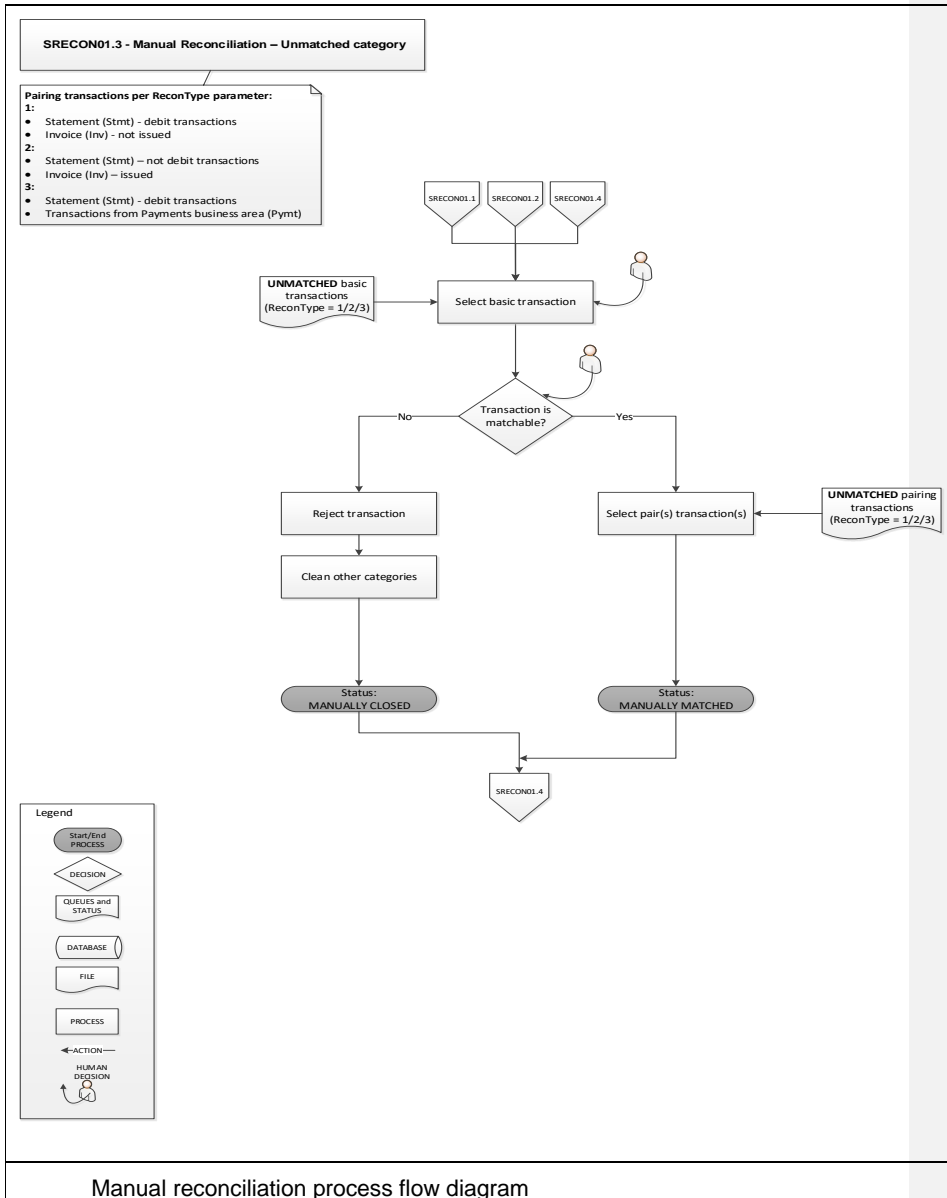
### 3.3.1.1.1 SRECON01.1 - Automatic Reconciliation



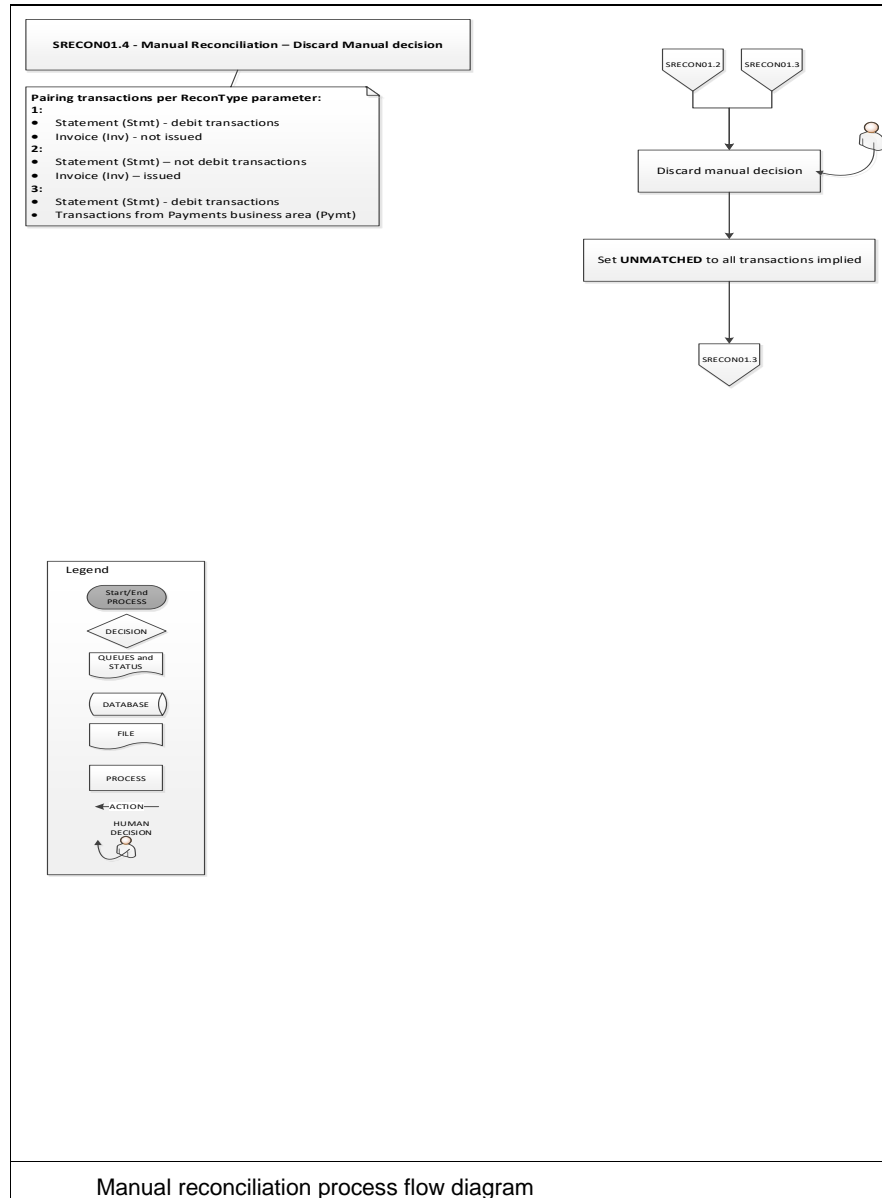
### 3.3.1.1.2 SRECON01.2.1 - Manual Reconciliation – Offered category



### 3.3.1.1.3 SRECON 01.3 - Manual Reconciliation – Unmatched category



### 3.3.1.1.4 SRECON 01.4 - Manual Reconciliation – Discard manual decision



### 3.3.1.2 FRECON01 Accounts payable and accounts receivable - description

#### 3.3.1.2.1 Reconciliation process

Reconciliation is an automatic or manual process of matching 2 or more financial transactions from different Business Areas. Unlike the automatic reconciliation where only 2 transactions are involved, in the manual reconciliation 2 or more transactions may be involved.

Automatic reconciliation is performed by perfectly matching specific fields from transactions; these fields are defined during the analysis period.

As a result, these financial transactions can be:

1. Matched – all the criteria are matched;
2. Partially matched – a subset of criteria is matched; in this case, the result is an offered match that needs to be manually confirmed.
3. Unmatched - none of the criteria matched.

In the manual process, the user can execute the following operations:

- Confirm an offered match
- Match 2 or more unmatched transactions
- Close an unmatched transaction. The transaction will not appear in any offer.
- Discard a previous user decision (match or close).

In order to decide whether to manually match transactions or not, the application displays relevant information of the transactions involved.

#### 3.3.1.2.2 Reconciliation status

As a result of reconciliation process, the transactions have one of the following **status**:

- **MATCHED** – final status, the transactions are automatically matched
- **MANUALLY MATCHED** – the transactions are manually matched
- **MANUALLY CLOSED** – the transaction is manually closed
- **UNMATCHED** – the transaction is unmatched or partially matched

Reconciliation process allows to unmatched a set of transactions with '**MANUALLY MATCHED**' or '**MANUALLY CLOSED**' status. After this operation the transactions have '**Unmatched**' status.

#### 3.3.1.2.3 Reconciliation categories

As a result of the reconciliation process, transactions are grouped in the following **categories**:

- *Matched:*

- pairs of 2 transactions automatically matched according to reconciliation criteria. These transactions have “**MATCHED**” status.
- Two or more transactions manually matched. These transactions have “**MANUALLY MATCHED**” status.
- **Offered:**
  - pairs of 2 transactions partially matched according to reconciliation criteria and offered for manual confirmation or rejection. These transactions have “**UNMATCHED**” status.
- **Unmatched:**
  - transactions that are not matched. These transactions have “**UNMATCHED**” status and they are submitted to a manually match process 1-n or n-1 or manually close operation.
- **Closed:**
  - transactions manually closed. These transactions have “**MANUALLY CLOSED**” status.

Resulted status (category) Initial status (category)	<b>MATCHED</b> (Matched category)	<b>UNMATCHED</b> (Offer category)	<b>UNMATCHED</b> (Unmatched category)	<b>MANUALLY MATCHED</b> (Matched category)	<b>MANUALLY CLOSED</b> (Closed category)
<b>MATCHED</b> (Matched category)		x	x	x	x
<b>UNMATCHED</b> (Offer category)	X		reject offer as Match	confirm offer as Match	x
<b>UNMATCHED</b> (Unmatched category)	X	x		create a Match	reject transaction
<b>MANUALLY MATCHED</b> (Matched category)	X	x	discard manual decision		x
<b>MANUALLY CLOSED</b> (Closed category)	X	x	discard manual decision	x	

Reconciliation process	Reconciliation process - result	Status	Category
Automatic	Matched	Matched	Matched
	Partially matched	Unmatched	Offered and Unmatched
	Unmatched	Unmatched	Unmatched
Manual	Manually matched	Manually matched	Matched
	Manually closed	Manually closed	Closed

#### 3.3.1.2.4 Reconciliation criteria

##### 3.3.1.2.4.1 Debit statement transactions vs invoices received from suppliers

Following information must match, in order to have the pair of transactions reconciled.

Debit statement transaction (findata.repstatstmt.trxmark='D')	Invoice received from suppliers (findata.repstatstmt.Invoicetype='Received')
Transaction reference (findata.repstatstmt.reference)	Reference (findata.repstatinvc.reference)
Entity (findata.repstatstmt.name)	Buyer (findata.repstatinvc.entity)
Amount (findata.repstatstmt.amount)	Amount (findata.repstatinvc.amount)
Currency (findata.repstatstmt.currency)	Currency (findata.repstatinvc.currency)

There are some transaction types that are unmatchable, which means that they do not have another transaction that can be linked to them, such as:

- Salary payments reflected in debit/credit statement transactions don't have a matching invoice
- Bank fees reflected in debit/credit statement transactions don't have a matching invoice
- STORNO payments

These types of transactions will be excluded from the reconciliation process, based on the criteria (e.g. Reference starting with "SAL") identified with the Client in the analysis period.

If no criteria is identified to exclude these transactions, they can be manually closed by the user, when they will appear unmatched.

#### 3.3.1.2.4.2 Credit statement transactions vs issued invoices

The following information must match, in order to have the pair of transactions reconciled.

Credit statement transaction (findata.repstatstmt.trxmark='C')	Issued invoices (findata.repstatstmt.Invoicetype='Issued')
Entity (findata.repstatstmt.name)	Seller (findata.repstatinvc.entity)
Amount (findata.repstatstmt.amount)	Amount (findata.repstatinvc.amount)
Currency (findata.repstatstmt.currency)	Currency (findata.repstatinvc.currency)

There are some transaction types that are unmatchable, which means that they do not have another transaction that can be linked to them, such as:

- Salary payments reflected in debit/credit statement transactions don't have a matching invoice
- Bank fees reflected in debit/credit statement transactions don't have a matching invoice
- STORNO payments

These types of transactions will be excluded from the reconciliation process, based on the criteria identified with the Client in the analysis period.

If no criteria is identified to exclude these transactions, they can be manually closed by the user, when they will appear unmatched.

#### 3.3.1.2.4.3 Debit statement transactions vs transactions from Payments business area

Following information must match, in order to have the pair of transactions reconciled.

Debit statement transaction (findata.repstatstmt.trxmark='D')	All financial transactions from Payments Business Area
Transaction reference	End to End Identification



Debit statement transaction (findata.repstatstmt.trxmark='D')	All financial transactions from Payments Business Area
(findata.repstatstmt.reference)	
Entity (findata.repstatstmt.name)	Debtor Name
Account Identification	Debtor Account
Amount (findata.repstatstmt.amount)	Amount (Amount+ccy)
Currency (findata.repstatstmt.currency)	Ccy (Amount+ccy)

### 3.3.1.3 FRECON02 - Reconciliation report

#### Filter criteria:

- *Reconciliation type*: drop-down list or radio boxes with *Recon supplier invoices*, *Recon issued invoice*, *Recon payments*
- *Categories*: drop-down list or radio boxes with *All*, *Matched*, *Offered*, *Unmatched*, *Closed*, for all categories, except *Recon payments* that will only have *Matched* and *Unmatched* category, since this type of reconciliation will only have the automatic flow
- *Status*: drop-down list with *All*, *UNMATCHED*, *MATCHED*, *MANUALLY MATCHED*, *MANUALLY CLOSED*, for all categories, except *Recon payments* that will only have *Matched* and *Unmatched* category, since this type of reconciliation will only have the automatic flow
- *Internal Entity*: drop-down list with *All*, *Name* field from Internal entities and accounts List. Transactions are filtered by Entity/ Buyer/ Seller / Payer
- *Currency*: drop down list with *All*, *Currency field from Internal Entities and accounts List*
- *Statement/Invoice Period*: selection from calendar. Transactions are filtered by Value date/ Payment Terms Maturity Date/ Execution Date

#### Report result:

The report contains all transactions corresponding to selected criteria.

The result of the report contains the following information:

Reconciliation Type	Report result fields	
<i>Recon supplier invoices</i>	<b>Statement transactions (debit)</b>	<b>Invoices received from suppliers</b>
	Reference Entity Amount (Amount+ccy) Value date Account Details	Reference Buyer Amount (Amount + ccy) Maturity Date Account Seller Number and Serial
<i>Recon issued invoices</i>	<b>Statement transactions (credit)</b>	<b>Issued invoice</b>
	Reference Entity Amount (Amount+ccy) Value date Account Details	Reference Seller Amount (Amount + ccy) Maturity Date Account Buyer Number and Serial
<i>Recon payments Payments Business Area vs. statement transactions (debit)</i>	<b>Statement transactions (debit)</b>	<b>Financial transactions(Payments Business Area)</b>
	Reference Entity Amount (Amount+ccy) Account Details Value date	Original ID Payer Amount (Amount+ccy) Payer Account Details Execution date Beneficiary Name Beneficiary Account

### 3.3.1.4 Database structure

#### 3.3.1.4.1 Statements

Statement transaction can be queried from the view `findata.repstatstmt`, having the following structure:

Field name	Description
correlationid	Uniquely identifies one transaction
messagetype	The type of transaction; "Statement transaction"
reference	The statement transaction reference
amount	The amount per transaction
currency	The currency used
valuedate	The value date
status	<p>The current state of the transaction</p> <p>'Completed - Rejected (reason: others)';</p> <p>'Completed - Suspended (reason: others)';</p> <p>'Completed - Rejected (reason: duplicate)';</p> <p>'Completed - Suspended (reason: others)';</p> <p>'Completed - Suspended (reason: flow)';</p> <p>'Completed - Sent';</p> <p>'Waiting for Authorization';</p> <p>'Waiting for Business Investigation';</p> <p>'Waiting for Technical Investigation';</p> <p>'Waiting for Duplicates Investigation';</p>

Field name	Description
	'In Process';  NOTE: Only transactions in 'Completed - Sent' state are to be eligible for reconciliation
sourcefilename	The file name of the original statement, provided by the bank
remittanceinfo	Transaction details
statementdate	The date interval of the transaction in statement; format: yyyy-mm-dd – yyyy-mm-dd  Open balance date – close balance date
statementnumber	The number of the statement, given by the bank
statementreference	The statement reference
name	Creditor/Debtor of the transaction
accountnumber	Creditor / Debtor IBAN (optional information)
trxmark	Creditor / Debtor indicator; "D" / "C"
insertdate	The timestamp of committing the transaction in the database
requestorservice	The application service fetching the statement to the database
queueenamel	The name of the queue currently holding the transaction  NOTE: empty for the state "Completed - Sent"



fintpc.findata.repstats  
tmt

### 3.3.1.4.2 Invoices

Invoice transactions can be queried from the view `findata.repstatinvc`, having the following structure:

Field name	Description
Correlationid	Uniquely identifies one transaction
Insertdate	The timestamp of committing the transaction in the database
Invoicetype	The type of the invoice "Received" / "Issued"
Reference	The payment reference
Cdtcustomername	Seller
Dbtcustomername	Buyer
Invoiceno	The number of the invoice
invoicserial	The serial of the invoice
Amount	The payment amount
Currency	The currency used
originalrefernce	The payment reference as assigned by the seller
cdtaccount	Seller account
maturitydate	Maturity date
Entity	Internal entity – seller/buyer



fintpc.findata.repstati  
nvc

### 3.3.1.5 SRECON01.2.2 - Manual reconciliation scenarios

When automatic reconciliation based on all reconciliation criteria cannot be done, the system will provide most likely matches, that will help manual reconciliation.

The offered matching is generated based on different scenarios where automatic reconciliation can fail, e.g.:

1. Same Invoice is paid twice; if the 2 different statements match the same invoice, one should be automatically matched and the other one should offered as a duplicate in order to be manually closed.
2. Single invoice being split into 2 credit or debit statement transactions.
3. An invoice is issued in a specific currency (e.g. EUR), and the payment is made in a different currency (e.g. RON), from an account with a different currency (also RON) and in this situation the statement would have the equivalent RON amount, calculated based on an national exchange rate (e.g. NBR), most probably locally stored.
4. An invoice is issued in a specific currency (e.g. EUR) and the payment is made also in EUR but from an account opened in a different currency (e.g. RON), this would be a situation in which the statement would also have some fees debited from the account, besides the invoice amount; in this case it is very important how the fees appear in the statement, if they are somehow linked to the transaction
5. Payment being made to a different account holder other than the one from the invoice.
6. Many invoices being grouped into one bigger payment transaction, that will be reflected into one bigger debit/credit statement transaction.

To provide the set of most suitable transaction for a given unmatched transaction a scoring process is needed.

There are some examples below, for manual reconciliation, both issued invoices vs. credit statement transactions and received invoices vs. debit statement transactions.

### 3.3.1.5.1 Examples

#### 3.3.1.5.1.1 ISSUED INVOICES vs STATEMENT TRANSACTIONS on CREDIT

Invoice					Statement								
<b>Direction (Issued/Received) (findata.repstatinv c.Invoicetype='Issued')</b>	Issued	Issued	Issued	Issued	<b>Direction (Debit/Credit) (findata.repstats tmt.trxmark='C')</b>	Credit	Credit	Credit	Credit	Credit	Credit	Credit	Credit
<b>Buyer (N/A)</b>	PAR1	PAR2	PAR1	PAR3	<b>Buyer (N/A)</b>	PAR1	PAR2	PAR1	PAR3	PAR1	PAR2	PAR1	PAR3
<b>Seller (findata.repstatinv c.entity)</b>	FINTPC 1	FINTPC 1	FINTPC2	FINTPC 2	<b>Seller (findata.repstats tmt.name)</b>	FINTPC1	FINTPC1	FINTP C2	FINTPC 2	FINTP C1	FINTP C1	FINTP C2	FINTP C2
<b>Amount (findata.repstatinv c.amount)</b>	10	5	10	25	<b>Amount (findata.repstats tmt.amount)</b>	10	5	10	25	14	7	21	28
<b>Maturity_date (findata.repstatinv c.maturitydate)</b>	15.04.20 20	15.04.2 020	15.04.20 20	15.04.20 20	<b>Balance Date (findata.repstats tmt.statemendate)</b>	11.04.20 20	12.04.20 20	11.04.2 020	13.04.20 20	11.04.2 020	12.04.2 020	11.04.2 020	13.04. 2020
<b>Invoice Reference (findata.repstatinv c.reference)</b>	APR000 1	APR00 02	FINTPC2 001	FINTPC 20002	<b>Transaction Reference (findata.repstats tmt.reference)</b>	UNIQ1	UNIQ2	UNIQ3	UNIQ4	UNIQ9	UNIQ1 0	UNIQ1 1	UNIQ 12

Matches: APR0001&UNIQ1 ; APR0002&UNIQ2 ; FINTPC2001&UNIQ3 ;FINTPC2002&UNIQ4

### 3.3.1.5.1.2 RECEIVED INVOICES vs STATEMENT TRANSACTIONS on DEBIT

Invoice					Statement								
<b>Direction (Issued/Received) (findata.repstat invc.Invoicetype='Received')</b>	Received	Received	Received	Received	<b>Direction (Debit/Credit) (findata.repstatst mt.trxmark='D')</b>	Debit	Debit	Debit	Debit	Debit	Debit	Debit	Debit
<b>Buyer (findata.repstat invc.entity)</b>	FINTPC1	FINTPC1	FINTPC2	FINTPC2	<b>Buyer (findata.repstatst mt.name)</b>	FINTP C1	FINTP C1	FINTP C2	FINTPC 2	FINTP C1	FINTP C1	FINTPC2	FINTPC2
<b>Seller (N/A)</b>	PAR9	PAR9	PAR7	PAR8	<b>Seller (N/A)</b>	PAR9	PAR8	PAR7	PAR8	PAR9	PAR8	PAR7	PAR8
<b>Amount (findata.repstat invc.amount)</b>	25	20	15	20	<b>Amount (findata.repstatst mt.amount)</b>	45	20	8	20	49	49	7	20
<b>Maturity_date (findata.repstat invc.maturitydate)</b>	15.04.2020	15.04.2020	15.04.2020	15.04.2020	<b>Balance Date (findata.repstatst mt.statemendate)</b>	10.04.2020	12.04.2020	17.04.2020	16.04.2020	10.04.2020	12.04.2020	17.04.2020	16.04.2020
<b>Invoice Reference (findata.repstat invc.reference)</b>	PR0001	PAR8001	PT0001	PAR8002	<b>Transaction Reference (findata.repstatst mt.reference)</b>	UNIQ5	UNIQ6	UNIQ7	UNIQ8	UNIQ 13	UNIQ 14	UNIQ15	UNIQ16

Matches: (PR0001+PR8001)&UNIQ5 ; PT0001&(UNIQ7+UNIQ15) ; PAR8002&UNIQ8

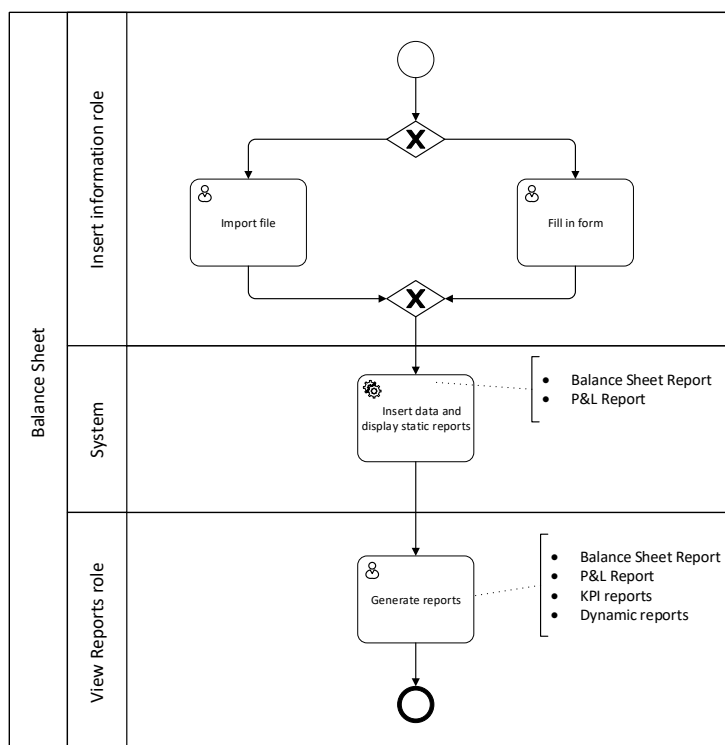


### 3.3.2 Balance Sheet

This feature uses Balance Sheet document information and generates very structured, easy to understand reports such as Balance Sheet Report and Profit & Loss Report, both numerical and graphical view.

#### 3.3.2.1 SBS01- Balance Sheet

Balance Sheet scenario is described below.



### 3.3.2.2 FBS01- Balance Sheet

This functionality allows users to generate different reports useful in a company, using both present indicators and historical information.

The input information can be fetched either through user interface or through an excel file import.

In case of a group of companies the user must first select the company (Internal Entity) that he would like to analyze.

Usually, these reports are generated monthly, in order to have a very up to date picture.

The system displays a graphical representation of the annual evolution, filtering can be done on every category, e.g. the annual evolution of Personnel Expenditure - YTD compared to previous years, the annual evolution of the turnover - YTD compared to previous years.

Data can be registered monthly (manual input in UI or file import) and it represents the value of the indicator in that month - the YTD value.

Data is stored in the system for every year, according to the following business rules.

#### **Business Rules:**

BR1: YTD values are registered in the system, monthly values are not registered in the system

BR2: YTD values (e.g. monthly registered) overwrite old YTD values

BR3: YTD values become annual values, once the year has ended

BR4: Annual new values overwrite annual old values - second confirmation needed (e.g. "Are you sure?" Or a 4 eyes supervision)

### 3.3.2.3 FBS02 - User Management

There are new specific roles that give the right to manage all information related to Balance Sheet feature.

The roles (user rights) specific to this feature are:

-View Balance Sheet Reports

-Insert Balance Sheet information: import (xls file) or manual input (fill in information in UI form) and generate static reports (i.e. Balance Sheet Report and P&L Report)

Both roles are able to generate static reports (i.e. Balance Sheet Report and P&L Report) but only "View Balance Sheet Reports" role is able to generate all types of reports including KPI reports and Dynamic reports.

The roles (user rights) should be assigned per tenant (Internal Entity), in order for a user from a specific tenant (Internal Entity) to have access only to Balance Sheet information related to his entity.

The CRUD matrix is described in the below attached file.

#### 3.3.2.4 FBS03 - Configurations

The configurations specific to this feature are:

- Company type: 3 possible values, one of them is implemented
- Maximum number of indicators that can be selected for the dynamic reports (Balance Sheet Report and P&L Report), that display the annual evolution of the selected indicators

#### 3.3.2.5 FBS04 - Reports

The two static reports that can be generated:

- P&L Report
- Balance Sheet Report

The business rules used to extract and calculate the information that is displayed in the reports are described in the file attached below.

The reports can be generated independently, if all mandatory fields for a static report (Balance Sheet Report or P&L Report) are provided by the user by manual input or by file import, that report will be generated. If at least one mandatory field for that specific report is not provided the report won't be generated and an error message will be displayed.

If information for both reports is inserted (Balance Sheet Report and P&L Report) irrespective of the method (import file or manual input in UI), both reports can be generated simultaneously.

Based on the data inserted in the system, the user can also generate some KPIs reports, specific to Balance Sheet Report, P&L Report and both reports (see file attached below)

Dynamic reports can also be generated, in order to analyze the annual evolution of the chosen Balance Sheet Report or P&L Report indicators and also of related KPIs, if necessary.

The dynamic reports will be generated for the period selected by the user, and for the selected indicators:

- for Balance Sheet Report and P&L Report indicators annual evolution, the dynamic report and related chart are generated together
- for KPIs annual evolution, the dynamic report and related chart are generated separately for each selected KPI

The reports displayed can also be exported as pdf/ Excel files.

### 3.3.2.6 FBS05 - Archiving

All data will be stored in the live data base for this feature, no data will be archived.

### 3.3.2.7 FBS06 - Audit Trails

The following specific **events** are also recorded:

- The action decisions performed by each user that result in modifying features' content, i.e. events related to Create (import file or fill in UI), Edit or Delete Balance Sheet information.

### 3.3.2.8 Application interface

There is no interfacing between applications for this feature: the input information is received through user interface (manual input or file upload) and the output information is kept in the system, reports can only be exported as pdf/ Excel file through user interface.

### 3.3.2.9 Data structures

Data structures specific to Balance Sheet feature and other specific information are described in the "SaaS\_BS\_data structures-mapping-calc.xls" file.

Before importing the file or filling in the information manually in UI, the user can select the year for which the insert is done, the default value is the current year.

### 3.3.3 Cash Reporting

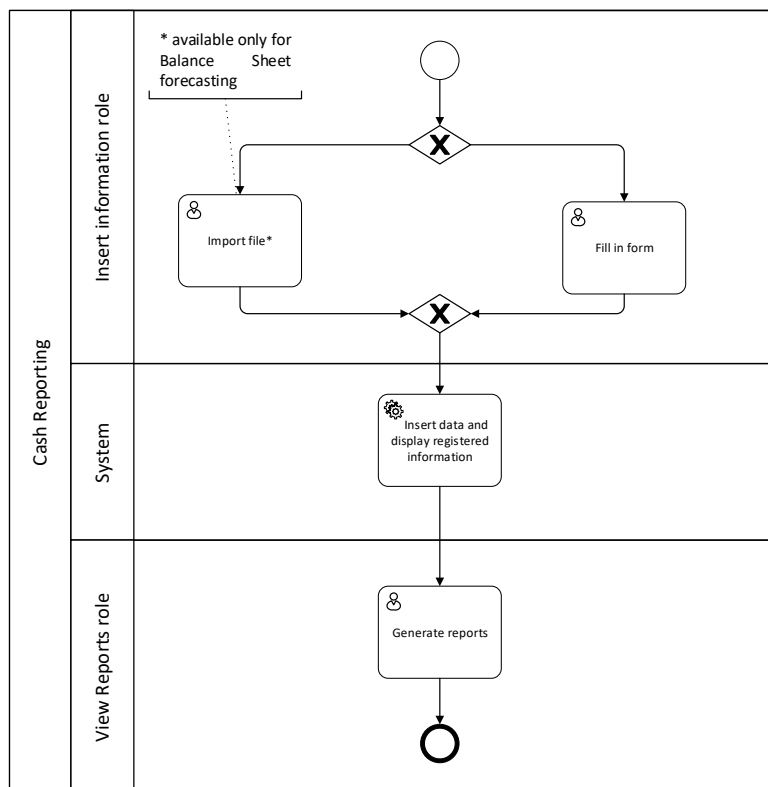
This feature offers forecasting for a future period of time, based on current information and on the estimated information.

There are 2 scenarios for forecasting:

- Cash flow forecasting
- Balance Sheet forecasting

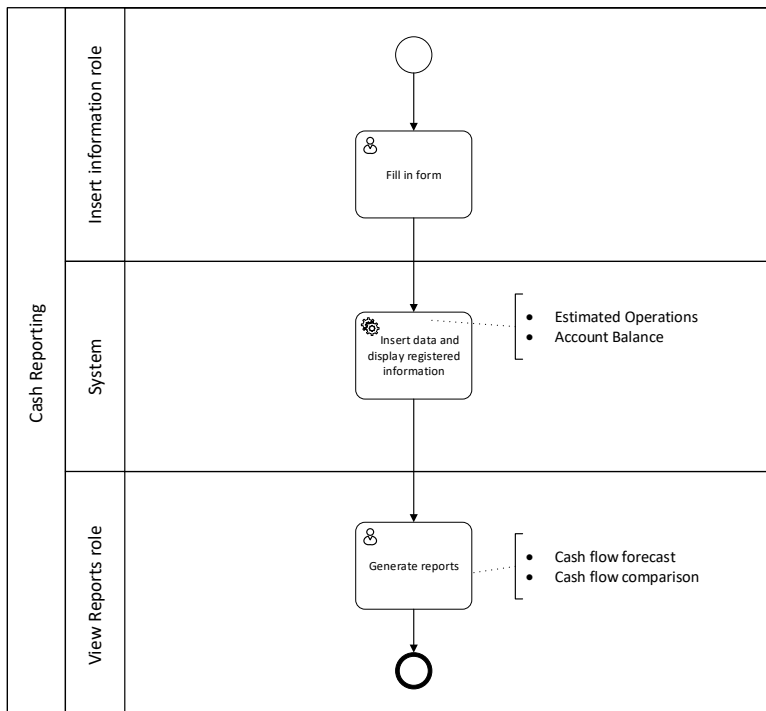
#### 3.3.3.1 SCASHREP01- Cash Reporting

Cash Reporting scenario is described below.



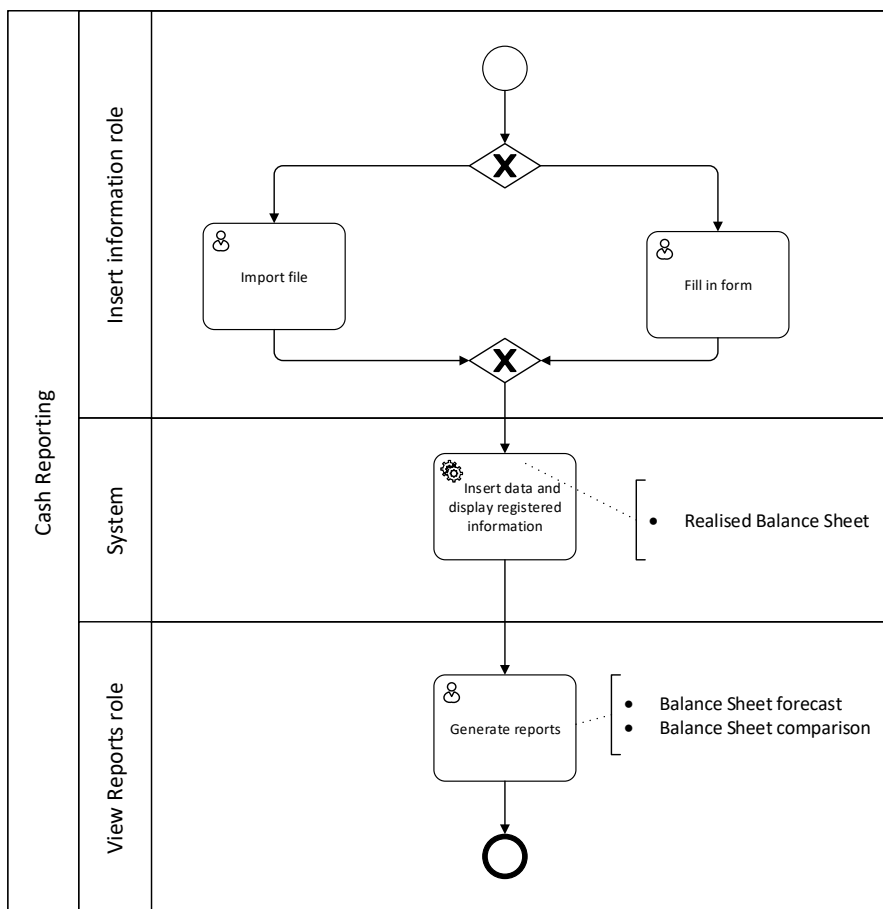
### 3.3.3.2 SCASHREP01.1 - Cash flow forecasting

Cash flow forecasting scenario is described below.



### 3.3.3.3 SCASHREP01.2 - Balance Sheet forecasting

Balance Sheet forecasting scenario is described below.



### 3.3.3.4 FCASHREP01 - Cash flow forecasting

The cash flow forecasting offers an estimation for the future days/ months, based on current status (balance of accounts) and on the estimated expenses (based on balance sheet data) and on the invoices (issued and not paid, received and not paid).

In case of a group of companies the user must first select the company (Internal Entity) that he would like to analyze.

The cash flow forecasting is made annually and is done at the beginning of every year but it can be updated if necessary, most probably monthly.

The current status for the cash flow represents the sum of the account balance of every account the company has at the banks it works with.

There are 2 options for this functionality:

- The cash flow forecasted for a future date
- The comparison between forecasted cash flow and actual cash flow

The input information is fetched from user interface.

The user can also set a default account (IBAN) for each partner.

The user can insert operations in the cash flow forecast, for both expenses and revenues, and he must insert an initial account balance for every account (IBAN).

Existing operations (expenses or revenues) can be edited, deleted or new ones can be added in the forecast section.

The cash flow forecast will be calculated for the inserted forecast date: only transactions with maturity date between (initial account balance date, forecast date) will be considered.

For the comparison between forecasted cash flow and actual cash flow, account balance amount will be inserted by the user for every account that needs comparison.

The comparison between forecasted cash flow and actual cash flow is presented both in numerical and graphical forms.

### **3.3.3.5 FCASHREP02 - Balance Sheet forecasting**

This Balance Sheet forecasting offers an estimation for the future Balance Sheet indicators based on realised Balance Sheet information and on the estimated modification rate.

In case of a group of companies, the user must first select the company (Internal Entity) that he would like to analyze.

There are 2 options for this functionality:

- The forecasted Balance Sheet for a future date
- The comparison between forecasted Balance Sheet and realized Balance Sheet

The Balance Sheet forecasting is annual but it will be updated if necessary.

The user has to first select the Realised Balance Sheet Date.



If no Realised Balance Sheet data for the selected Date exists in the system then a message will be displayed to state the fact that the user has to Register the Realised Balance Sheet P&L first, just how it is described in chapter 3.3.4 [Register Realised Balance Sheet \(P&L\) - Common feature](#)

If Realised Balance Sheet data for the selected Date exists in the system, the data will be fetched in Balance Sheet Amount fields.

The forecasted values for Balance Sheet are calculated using the proposed modification rate (percent %) calculated based on historical information.

The proposed modification rate (percent %) calculated by the system can be manually edited by the user.

For the comparison between forecasted Balance Sheet and realized Balance Sheet, a comparison date must be selected.

If no Realised Balance Sheet data for the selected Comparison Date exists in the system then a message will be displayed to state the fact that the user has to Register the Realised Balance Sheet P&L first (sheet "Realised BS P&L")

If Realised Balance Sheet data for the selected Comparison Date exists in the system, the data will be fetched in Realised Amount fields.

If no Forecasted Balance Sheet data for the selected Comparison Date exists in the system then a message will be displayed to state this fact.

If Forecasted Balance Sheet data for the selected Comparison Date exists in the system, the data will be fetched in Forecasted Amount fields.

The comparison between forecasted Balance Sheet and actual Balance Sheet is presented in both numerical (as difference between them) and graphical forms.

### **3.3.3.6 FCASHREP03 - User Management**

There are new specific roles that give the right to manage all information related to Cash Reporting feature.

The roles (user rights) specific to this feature are:

- View Reports
- Insert information: import (xls file-only for Balance Sheet forecasting) or manual input (fill in information in UI form) and view inserted information.

Both roles are able to view inserted data (i.e. Realised Balance Sheet and estimated operations) but only "View Reports" role is able to generate forecasting reports and comparison reports.

The roles (user rights) should be assigned per tenant (Internal Entity), in order for a user from a specific tenant (Internal Entity) to have access only to Cash Reporting information related to his entity.

The CRUD matrix is described in the below attached file.

#### **3.3.3.7 FCASHREP04 - Reports**

The reports that can be generated are:

- Balance Sheet forecast
- Balance Sheet comparison
- Cash flow forecast
- Cash flow comparison

The business rules used to extract and calculate the information displayed in the reports are described in the file attached below.

If all mandatory fields for a report are provided by the user by manual input or by file import, that report will be generated. If at least one mandatory field for that specific report is not provided the report won't be generated and an error message will be displayed.

The reports displayed can also be exported as pdf/ Excel files.

#### **3.3.3.8 FCASHREP05- Archiving**

All data will be stored in the live data base for this feature, no data will be archived.

#### **3.3.3.9 FCASHREP03 - Audit Trails**

The following specific **events** are also recorded:

- The action decisions performed by each user that result in modifying features' content, i.e. events related to Create (import file or fill in UI), Edit or Delete Cash Reporting information.

#### **3.3.3.10 Application interface**

There is no interfacing between applications for this feature: the input information is received through user interface (manual input or file upload) and the output information is kept in the system, reports can only be exported as pdf/ Excel file through user interface.

#### **3.3.3.11 Configurations**

There are no configurations specific to this feature.

### 3.3.3.12 Data structures

Data structures specific to Cash Reporting feature and other specific information are described in the "SaaS\_CR\_data structures-mapping-calc" file.

When necessary, before importing the file or filling in the information manually in UI, the user can select the date for which the insert is done, the default value is the current date.

### 3.3.4 Register Realised Balance Sheet (P&L) - Common feature

This feature is common for both [3.3.2 Balance Sheet](#) and [3.3.3 Cash Reporting](#) features and it collects input data used as:

- information to generate P&L Report (Balance Sheet Feature)
- a baseline in forecasting Balance Sheet Indicators (Cash Reporting feature)
- comparison information for calculating the difference between realised Balance Sheet and Forecasted Balance Sheet (Cash Reporting feature)

#### 3.3.4.1 SCOMM01- Register Balance Sheet (P&L)

The input information is fetched either from user interface or from an imported excel file. Information collected in this step (sheet "Realised BS P&L", file [http://tfs/sites/FinOps/Requirements/FinTPc/SaaS\\_CASHREP\\_data%20structures-mapping-calc.xlsx](http://tfs/sites/FinOps/Requirements/FinTPc/SaaS_CASHREP_data%20structures-mapping-calc.xlsx)) is common for these two features [3.3.2 Balance Sheet](#) and [3.3.3 Cash Reporting](#), and the information inserted will be used for both features, following the business rules described below:

**BR12:** If no Balance Sheet data for the selected date (Realised Balance Sheet Date) exists in the system then the 2 input options will be displayed: "Fill in UI" and "Import excel file with Standard Balance Sheet".

**BR13:** If "Fill in UI" option is selected then the user will manually insert the information in Balance Sheet Amount fields (which represent a limited number of fields necessary for Cash Reporting feature). The user will also have the option to **extend** the list of fields that he/she would like to fill in, in order to have the full list of fields that is necessary to generate the P&L Report from Balance Sheet feature.

**BR14:** If "Import excel file with Standard Balance Sheet" option is selected then the user will import the file and the specific information will be extracted from the file and displayed in **extended** list of Balance Sheet fields (necessary for both Balance Sheet feature and Cash Reporting feature).

---

**BR15:** If **extended\*** Balance Sheet data for the selected date (Realised Balance Sheet Date) exists in the system, the data will be fetched in the extended list of Balance Sheet fields.

**BR16:** If limited Balance Sheet data for the selected date (Realised Balance Sheet Date) exists in the system, the data will be fetched in the limited list of Balance Sheet fields, that will only be used by Cash Reporting feature.

**\*extended** - the extended list of Balance Sheet fields is mandatory to generate the P&L Report from Balance Sheet Feature.

For Cash Reporting feature only the limited list of Balance Sheet is mandatory, the extended list of Balance Sheet fields is optional.

## 4 External Interface Requirements

Standard interface requirements and new interface requirements are mentioned below, in order to capture all external interfaces.

**User interfaces** characteristics are covered by the standard configuration.

**Hardware interfaces** are out of the scope of the standard configuration.

**Software interfaces** specific to standard configuration are between the standard configuration components, between the standard configuration components and other applications and also between the standard configuration components and users.

Since new components are delivered, similar to standard configuration, there is also: an interface between the new system components (if necessary), an interface between new system components and standard system components, between new system components and other applications and between new system and users. Details about these interfaces are to be found in the dedicated chapters.

**Communication interfaces** described for the new system components are in addition of all communication interfaces described for standard configuration.

## 5 Other Non-functional Requirements

All Non-functional requirements regarding **performance, safety, security** and **software quality**, that are related to the standard configuration, are also valid for new system features.

Besides these requirements, there are also some specific SaaS requirements that are described below.

### 5.1 Software as a Service Requirements

#### 5.1.1 SAAS01 - Architecture requirements - Multi-tenancy

Multi-tenancy ensures that every client is on the same version of the software and that he/she benefits from the new features introduced by each software update.

Because every company uses the same instance of the application, there is a segregation for any type of data in the application. Users from a company have access only to data related to the company they belong to: transactions, flows, lists, functionalities, contracts, users, etc.

Multi-tenancy implies that besides the business separation, there is also an administrative separation. Before having the data separation, that is made using entity and message type and that isolates data between the users of the same company, a superior layer is

created: an organizational structure that is configured before anything else, that isolates data between the users of different companies.

Each user, flow, transaction and all other functionalities are linked to the organizational structure.

The organizational structure is populated with all Clients names, that form the tenants.

The SaaS can be offered by a Provider (e.g. a bank) to its Clients, so the Provider will have one instance of the application and multiple tenants (Companies)

There are 2 levels in a multi-tenant architecture:

- Application level
- Tenant Level

Users and some configurations are configured at the Tenant Level, that means that the configurations are made for each tenant (Company) separately.

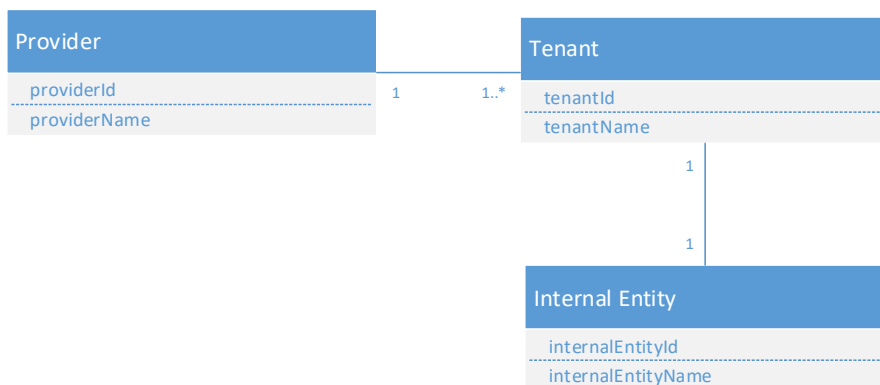
Still, there are some configurations that are not made at the Tenant Level, but at the Application Level (e.g. Routing Schemas) since they are the same for every client, and if for example a new Routing Schema will be created, that routing schema can be later on used for every client.

There are 2 type of administrators that cover multi-tenancy:

- Application Administrator: creates configurations for all tenants (general configurations common for all user or specific configurations created for each tenant separately)
- Business Administrator: creates specific business configurations just for one tenant (company)

While the Business Administrator can only make configurations on the Tenant Level, the Application Administrator can make configurations both at the Application Level and Tenant Level, according to the configurations described below.

The diagram below displays the high level relationships between the classes in multi-tenancy architecture.



1 Provider has 1 or many Clients/ Companies/ Tenants.

1 Tenant has 1 Internal Entity.

The following tables will help better express how these concepts act:

Concept	is operated at	Level
Provider		Application level
Tenant=Internal Entity		Application level

Concept	visible in UI	Yes/No
Provider		No
Tenant=Internal Entity		Yes

#### 5.1.2 SAAS02 - Update/ upgrade policy

The application itself is a single version of the software that is regularly updated. Any upgrade of the application, either initiated by the application provider or by a specific client, should be non-disruptive, so that noninvolved clients are not impacted.

#### 5.1.3 SAAS03 - Configurability

In this application a user is able to choose from multiple types of configurations. The flows and the interfaces that are responsible for fetching messages are designed as configurable features, in order to be easily personalized for each client, by a regular administrator (such as a business person), without the need to involve the development and the testing team.

Configurable features:

- Routing schemas (configured by Application Administrator – can create either a general configuration common for all users or specific configurations for each tenant separately)
- Internal Entities - this is actually the tenants list (configured by Application Administrator); a checklist is necessary in order for the Application Administrator to make the desired configurations, especially in terms of Roles and separation between internal entities and message types
- Routing Rules (configured by Application Administrator - can create either a general configuration common for all users or specific configurations for each tenant separately)

- Routing Schema Time Limits (configured by Application Administrator - can create either a general configuration common for all users or specific configurations for each tenant separately)
- Users (configured by Application Administrator, but for each tenant separately)  
The roles that can be assigned to users are standard configuration roles (e.g. Queues) and new roles corresponding to the new features (e.g. Balance Sheet and Cash Reporting roles)
- Banks (configured by Application Administrator)
- Accounts (configured by Business Administrator)
- Services (configured by Application Administrator - can do either a general configuration common for all users or specific configurations for each tenant separately)
- Configurations: Duplicate detection period (configured by Application Administrator - can create either a general configuration common for all users or specific configurations for each tenant separately)
- Queues (configured by Application Administrator - can create either a general configuration common for all users or specific configurations for each tenant separately)
- Connectivity management with the bank (configured by Application Administrator, but for each tenant separately)

#### 5.1.4 SAAS04 - Security and privacy requirements

Multi-tenancy described above requires additional security and privacy besides what is already enabled with the standard configuration.

All data segregated by the multi-tenancy is also kept private between different tenants and secured in order to not be accessed from a non-authorized tenant or even non-authorized external entities.

#### 5.1.5 SAAS05 - Centralized administration

The application can be easily and quickly configured by the application provider in order to get it running as fast as possible. Every parametrization requested by the Client (e.g. new flow, new user) is done by the application provider and for that he/she needs to have all the necessary tools to diagnose exceptions.

Additional information (i.e. Client name) is filled in when a new user is created, in order to configure it under the proper Client.

Events report and Audit Trails are also enhanced with information about the Client's name.

#### 5.1.6 SAAS06 - Predictable cost model



Implementation costs are predictable, subscription-based pricing have no hidden fees, and no investments are required for hardware and software license fees. Centralized reports can be configured in order to be used for calculating each Client's subscription, based on the total number of eligible transactions and on the functionalities offered.

#### 5.1.7 SAAS07 - Performance requirements

Multi-tenancy implies that the standard configuration performance requirements will be approximately multiplied with the maximum number of tenants supported.

In terms of concurrent user access the system has to be able to handle requests from an estimated number of 50 concurrent users/tenant?

In terms of user interface response each request should be resolved in a reasonable timeframe – for normal navigation, users should not wait longer than 1 second, while for complex processing (generating complex reports with specific criteria) responses should not take longer than 5 seconds.

In terms of performance process transactions the system has to be able to comfortably process daily volumes estimated by corporations (usually around 10 000 daily transactions/tenant?).

## 6 Other Requirements

### 6.1 Appendix A: Glossary

AISP = Account Information Service Provider

API = Application Programming Interface

PSD2 = Payment Service Direct 2

SMEs = Small and Medium-sized Enterprises

TOSS = Treasure Open Source Software

TPP = Third Party Provider

SaaS= Software as a Service

RTO=Recovery Time Objective

RPO= Recovery Point Objective

BR= Business Rules

YTD= Year to date

CRUD matrix = Create/ Read/ Update/ Delete user rights matrix