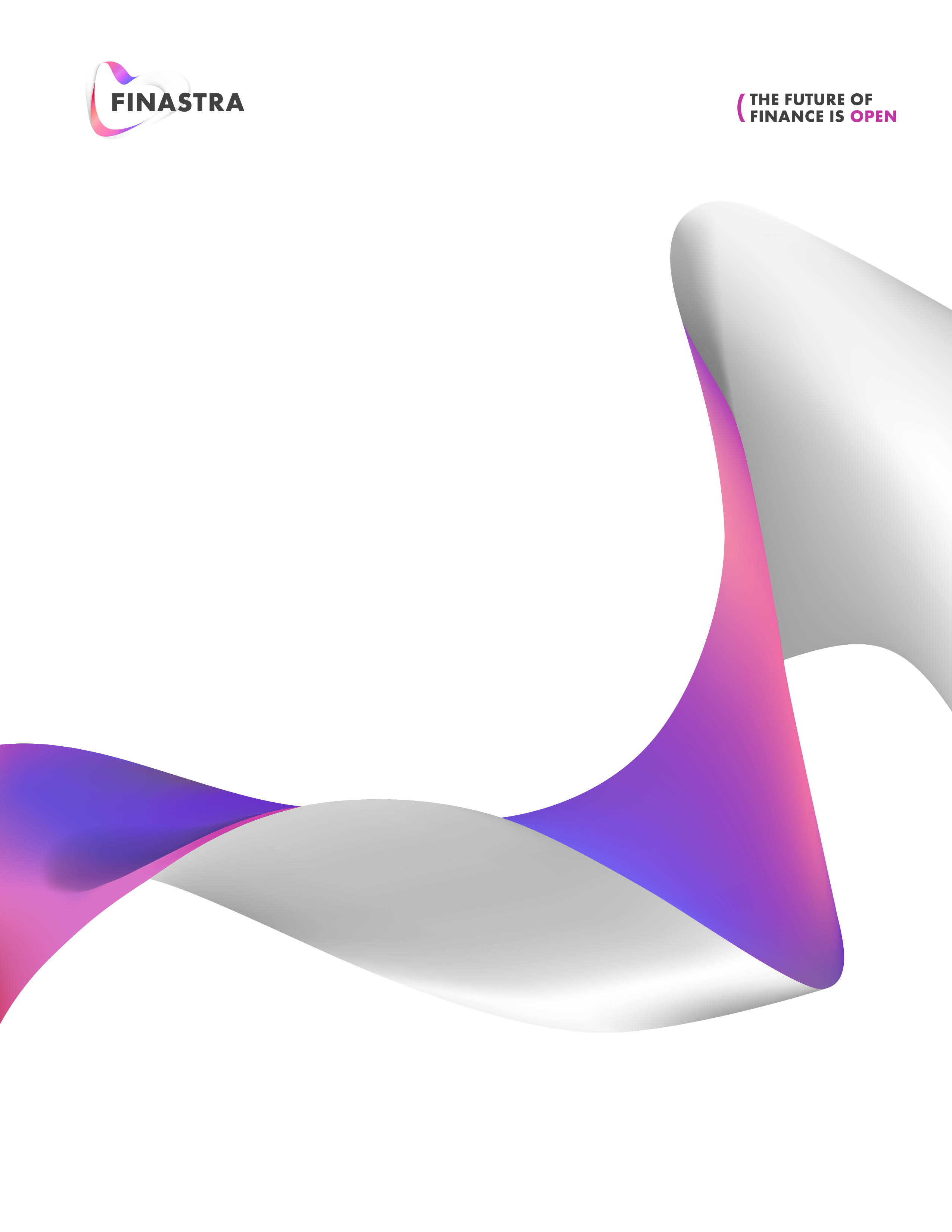


|  |
| --- |
| Shashi Jakka  July 20, 2021 |



|  |
| --- |
| Fusion Banking Loan IQ |
| Fusion Loan IQ – Account Balance Check Specification Document V1.0 |
|  |

Table of Contents

[1 Preface 4](#_Toc77590548)

[1.1 Document information 4](#_Toc77590549)

[1.2 Version History 4](#_Toc77590550)

[1.3 Key Contacts 5](#_Toc77590551)

[1.4 Document Sign-off 5](#_Toc77590552)

[1.5 Review and Changes 6](#_Toc77590553)

[1.6 Glossary 6](#_Toc77590554)

[2 Introduction 8](#_Toc77590555)

[2.1 Scope of Document 8](#_Toc77590556)

[2.2 Functional Requirement 8](#_Toc77590557)

[2.2.1 In Scope 8](#_Toc77590558)

[2.2.2 Out of Scope 8](#_Toc77590559)

[3 Interface Solution 8](#_Toc77590560)

[3.1 Solution in steps 9](#_Toc77590561)

[3.2 UBS Web Service Call Request and Response Sample 10](#_Toc77590562)

[3.2.1 Field Mapping 10](#_Toc77590563)

[3.2.2 Request 11](#_Toc77590564)

[3.2.3 Response 12](#_Toc77590565)

[3.3 Assumptions 13](#_Toc77590566)

[4 Dependency 14](#_Toc77590567)

[5 Referenced Documents 14](#_Toc77590568)

[6 Appendices 14](#_Toc77590569)

[7 Open Issues 15](#_Toc77590570)



**Table/Picture Index**

[Table 1: Document Information 4](#_Toc75182343)

[Table 2: Version History 5](#_Toc75182344)

[Table 3: Key Contacts 5](#_Toc75182345)

[Table 4: Document Signoff 6](#_Toc75182346)

[Table 5: Review Teams 6](#_Toc75182347)

[Table 6: Glossary 7](#_Toc75182348)

[Table 7: Field Mapping**Error! Bookmark not defined.**](#_Toc75182349)

[Table 8: Referenced Documents 14](#_Toc75182350)

[Table 9: Appendices 14](#_Toc75182351)

[Table 10: Open Issues 15](#_Toc75182352)

# Preface

## Document information

This document is one of the deliverables of the GIB – LIQ Implementation project of Finastra & PS Global Technologies FZCO (PS Global).

The purpose of this document is to specify the details related to Account Balance Check Validation.

This document will also list down all the assumptions that have been considered for the development of the solution.

The following table provides general information on the document:

|  |  |
| --- | --- |
| Document information | |
| Author | Shashi Jakka |
| Project | GIB - Loan IQ Project Implementation |
| Document name | 1.0 |
| Applicability | Finastra, GIB and Internal |
| Released by | PS Global |

Table 1: Document Information

## Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Date | Description of change |
| 1.0 | Shashi Jakka | 15.07.2021 | Initial Draft |
|  |  |  |  |
|  |  |  |  |

Table 2: Version History

## Key Contacts

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Business/Technical | Email | Phone |
| Sudhakar Vemuri | Finastra PSA | [Sudhakar.vemuri@psglobal.ae](mailto:Sudhakar.vemuri@psglobal.ae) ,[sudhakar.vemuri@finastra.com](mailto:sudhakar.vemuri@finastra.com) | +971 56 603 1201 |
| Moussa Abboud | Finastra Project Manager | [moussa.abboud@finastra.com](mailto:moussa.abboud@finastra.com) | +971 56 994 9419 |
| Satish Golla | Finastra Senior Business Consultant | [Satish.golla@psglobal.ae](mailto:Satish.golla@psglobal.ae) | +971 52 394 4109 |
| Roni Ruzario | Finastra Senior Technical Consultant | [roni.ruzario@psglobal.ae](mailto:roni.ruzario@psglobal.ae) | +971 50 652 4835 |
| Nawazish Khan | Finastra Senior Technical Consultant | [nawazish.khan@psglobal.ae](mailto:nawazish.khan@psglobal.ae) | +971 50 458 6487 |
| Shashi Jakka | PS Global Technical Consultant | [shash.jakka@psglobal.ae](mailto:shash.jakka@psglobal.ae) | +91 9533160014 |

Table 3: Key Contacts

## Document Sign-off

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Role | Date | Signature |
|  | <Client Business> |  |  |
|  | <Finastra Build Team> |  |  |
|  |  |  |  |
|  |  |  |  |

Table 4: Document Signoff

## Review and Changes

This document must be reviewed by Business Analysts, Developers, Product Managers, and IT staff to make sure that bank’s requirements are met and flow between the systems is understood. Below table shows the list of all the stakeholders responsible for the review of this document.

|  |  |  |
| --- | --- | --- |
| Review team member | Department | Role / Function |
|  |  |  |
|  |  |  |
|  |  |  |

Table 5: Review Teams

## Glossary

|  |  |
| --- | --- |
| Abbreviation | Definition |
| BO | Business Object |
|  |  |
| Loan IQ | Fusion Loan IQ |
| N/A | Not Applicable |
|  |  |

Table 6: Glossary

# Introduction

## Scope of Document

Scope of this document is to describe the solution and specification for the Account Balance Check required before the processing of DDA type of payments in Loan IQ

This document is relevant for those who:

* Are part of Finastra delivery team for Loan IQ
* Bank’s business & IT teams
* PS Global Business Analysts and Developers

## Functional Requirement

There are different type of payments in LoanIQ , that are need to validated before releasing,  
For DDA payments at the time of approval , should check borrowers account for sufficient funds to approve the payment. If the borrower’s account doesn’t have enough balance ,then the payment should be declined effectively.

### In Scope

For Principal Payment and Accrual Cycle Payment types in Loan IQ where the cashflow payment method is DDA this validation is performed.

### Out of Scope

Any other payment in Loan IQ that is not type of DDA are out scope

# Interface Solution

A trigger will be configured for the Payment Transaction that will be fired, whenever a particular payment related cashflows are sent to release , we’ll perform the validation , on given details by making webservice call. From the response we’ll check for the amount available in the account , then we’ll compare the requested amount and available amount , and return the result of it , based on result we’ll allow further processing, else throw an hard error with an error message.

## Solution in steps

🡪Below is the steps-by-step pseudo code for the Payment Validations

1. Setup the business event triggers with action as “Internal Validation” for the below transactions
   1. Loan Principal Payment (PPPMT)
   2. Loan Accrual Cycle Payment (OPMT)
2. When an event has been created in Loan IQ and after entering all the details, user will try to complete the workflow “Send to release”. At this point, triggers will be fired and do the validation. Validation pseudocode as below
   1. Identify the business object of the transaction
   2. Get the cashflows of the transaction
   3. Identify the direction of Cashflow
   4. Identify the payment method code from cashflow
   5. If Cashflow direction is “FROM” & Payment method code Is “DDA” then proceed further
   6. Else return saying “Validation is not required”
   7. Create a variable with name “requestedAmount” and store the value of amount being requested
   8. Using cashflows Remittance Instruction fetch the borrowers bank account details
   9. Make a Web Service call to bank for balance Inquiry with given WsURL
   10. If request is successfully sent and got the response back, parse the response and convert to into customised readable format for further processing
   11. Else throw an error “Failed to Connect with Bank server”
   12. From the response check the value of field “AvailableBalanceInAccountCurrency”
   13. If the value is proper Number, then procced further
   14. Else throw an error saying “Expected a Number, but received something else”
   15. Compare the value of “requestedAmount” and “AvailableBalanceInAccountCurrency”
   16. If Requested amount is greater than the available balance, throw an error

“The Account ‘Account Number’ has - Insufficient funds, balance available is - ‘Available balance’ ”

* 1. Else

Continue the transaction without any payment Error

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Object Type | Object Sub-type | Event Type | Event Name | Execution Type | Query Function |
| Outstanding Transaction (OTR) | Loan Principal Payment (PPPMT) | Transaction Event (TRE) | Sent to Approval (STA) | Internal Validation | liq.xq.em.evo.BalanceCheckValidation |
| Accrual Cycle Payment | Accrual Cycle Payment (OPMT) | Transaction Event (TRE) | Sent to Approval (STA) | Internal Validation | liq.xq.em.evo.BalanceCheckValidation |

## UBS Web Service Call Request and Response Sample

Request from Loan IQ should be sent in the format as below (This is based on the current version available).

### Field Mapping

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Loan IQ Field Name** | **Request /Response Tag** | **Object** | **Method** | **M/O** |
| strDirectionCode |  | Cashflow | cashflowobj.getRemittanceInstruction()  .getRemittanceMethod() | M |
| decRequestdAmount |  | Cashflow | object.amountAbs().amount | M |
| strAccountNumber | Gib:Account | Cashflow | Cashflowobj.getRemittanceInstruction()  .getAccountNumber() | M |
| strBranch | Gib:Branch | CashFlow | cashflowobj.getRemittanceInstruction()  .getBranch() | M |
| strIALGUID | gib1:IALGUID |  |  | M |
| strTime | gib1:MsgDateTime |  |  | M |
| decAvailableBalanceInAccountCurrency | gib:  AvailableBalanceIn  AccountCurrency |  |  | M |

### Request

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/" xmlns:gib="http://schemas.datacontract.org/2004/07/GIB.IAL.Account" xmlns:gib1="http://schemas.datacontract.org/2004/07/GIB.IAL.Framework.Message.Header">

<soapenv:Header/>

<soapenv:Body>

<tem:BalanceInquiry>

<!--Optional:-->

<tem:request>

<!--Optional:-->

<gib:Account>?</gib:Account>

<gib:Branch>?</gib:Branch>

<!--Optional:-->

<gib:CardNo>?</gib:CardNo>

<gib:MessageHeader>

<gib1:IALGUID>?</gib1:IALGUID>

<!--Optional:-->

<gib1:LanguageCode>?</gib1:LanguageCode>

<gib1:MsgADUserName>?</gib1:MsgADUserName>

<!--Optional:-->

<gib1:MsgADUserPassword>?</gib1:MsgADUserPassword>

<gib1:MsgDateTime>?</gib1:MsgDateTime>

<!--Optional:-->

<gib1:MsgIdentifier>?</gib1:MsgIdentifier>

<!--Optional:-->

<gib1:MsgKey>?</gib1:MsgKey>

<!--Optional:-->

<gib1:MsgProcCode>?</gib1:MsgProcCode>

<gib1:MsgSourceChannelID>?</gib1:MsgSourceChannelID>

<gib1:MsgSourceEnv>?</gib1:MsgSourceEnv>

<!--Optional:-->

<gib1:ParentLogId>?</gib1:ParentLogId>

</gib:MessageHeader>

</tem:request>

</tem:BalanceInquiry>

</soapenv:Body>

</soapenv:Envelope>

### Response

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tem="http://tempuri.org/" xmlns:gib="http://schemas.datacontract.org/2004/07/GIB.IAL.Account" xmlns:gib1="http://schemas.datacontract.org/2004/07/GIB.IAL.Framework.Message.Header">

<soapenv:Header/>

<soapenv:Body>

<tem:BalanceInquiryResponse>

<!--Optional:-->

<tem:BalanceInquiryResult>

<!--Optional:-->

<gib:Balances>

<!--Zero or more repetitions:-->

<gib:AccountBalanceDetails>

<!--Optional:-->

<gib:Account>?</gib:Account>

<!--Optional:-->

<gib:AvailableBalanceInAccountCurrency>?</gib:AvailableBalanceInAccountCurrency>

<!--Optional:-->

<gib:AvailableBalanceInLocalCurrency>?</gib:AvailableBalanceInLocalCurrency>

<!--Optional:-->

<gib:Currency>?</gib:Currency>

</gib:AccountBalanceDetails>

</gib:Balances>

<gib:MessageHeader>

<!--Optional:-->

<gib1:ErrorCode>?</gib1:ErrorCode>

<!--Optional:-->

<gib1:ErrorCodeNative>?</gib1:ErrorCodeNative>

<!--Optional:-->

<gib1:ErrorDescription>?</gib1:ErrorDescription>

<!--Optional:-->

<gib1:ErrorDescriptionNative>?</gib1:ErrorDescriptionNative>

<gib1:IALGUID>?</gib1:IALGUID>

<gib1:IsSucess>?</gib1:IsSucess>

<gib1:MsgIdentifier>?</gib1:MsgIdentifier>

<!--Optional:-->

<gib1:MsgProcessOnServer>?</gib1:MsgProcessOnServer>

<!--Optional:-->

<gib1:MsgStage>?</gib1:MsgStage>

<!--Optional:-->

<gib1:System>?</gib1:System>

</gib:MessageHeader>

</tem:BalanceInquiryResult>

</tem:BalanceInquiryResponse>

</soapenv:Body>

</soapenv:Envelope>

## Assumptions

* There will not be any change in Soap Request and Response format during the project.

# Dependency

# Referenced Documents

The following table contains links to referenced documents.

|  |  |  |
| --- | --- | --- |
| No. | Referenced Document | Link |
| 1 |  |  |
| 2 |  |  |

Table 7: Referenced Documents

# Appendices

|  |  |  |
| --- | --- | --- |
| No. | Appendix | Link |
| 1 | *Appendix* |  |
| 2 |  |  |

Table 8: Appendices

# Open Issues

*Open issues are relevant if draft versions are published. Open issues will be tracked outside of this document and should be solved by final release of the document.*

|  |  |
| --- | --- |
| No. | Description |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |

Table 9: Open Issues



|  |  |
| --- | --- |
| **About Finastra** Finastra unlocks the potential of people and businesses in finance, creating a platform for open innovation. Formed in 2017 by the combination of Finastra and D+H, we provide the broadest portfolio of financial services software in the world today –spanning retail banking, transaction banking, lending, and treasury and capital markets. Our solutions enable customers to deploy mission critical technology on premises or in the cloud. Our scale and geographical reach means that we can serve customers effectively, regardless of their size or geographic location – from global financial institutions, to community banks and credit unions. Through our open, secure and reliable solutions, customers are empowered to accelerate growth, optimize cost, mitigate risk and continually evolve to meet the changing needs of their customers. 48 of the world’s top 50 banks use Finastra technology. Please visit [**finastra.com**](http://www.finastra.com)  Finastra and the Finastra ‘ribbon’ mark are trademarks of the Finastra group companies.  © 2017 Finastra. All rights reserved. | **North American Headquarters** 120 Bremner Boulevard 30th Floor Toronto, Ontario M5J 0A8 Canada  T: +1 888 850 6656 |