**WHOLESALE BANKING ORDER MANAGEMENT REPORT GENERATION AND MANAGEMENT**

SEWP ZG629T DISSERTATION

By

**Sanjeev Kumar**

**2012HW68838**

Dissertation Work carried out at

Wipro Technologies, Bangalore

**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE**

**Pilani (Rajasthan), India**

November, 2016

SEWP ZG629T DISSERTATION

**WHOLESALE BANKING ORDER MANAGEMENT REPORT GENERATION AND MANAGEMENT**

Submitted in partial fulfillment of the requirements of

M.S. Software Engineering Degree Program

By

Sanjeev Kumar

(2012HW68838)

Under the supervision of

Hemadri Naidu Mullangi, Technical Lead,

Wipro Technologies, Bangalore.

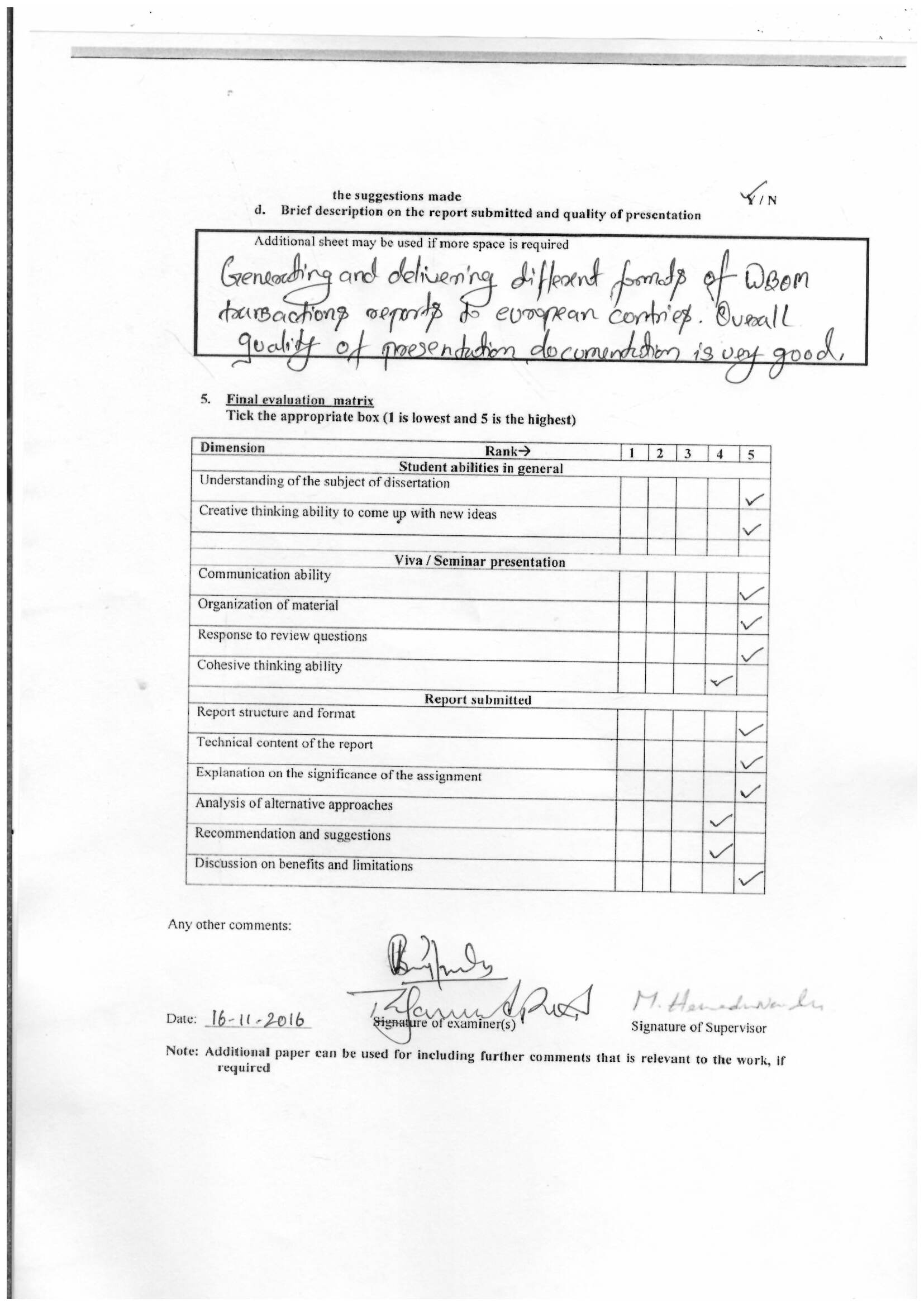
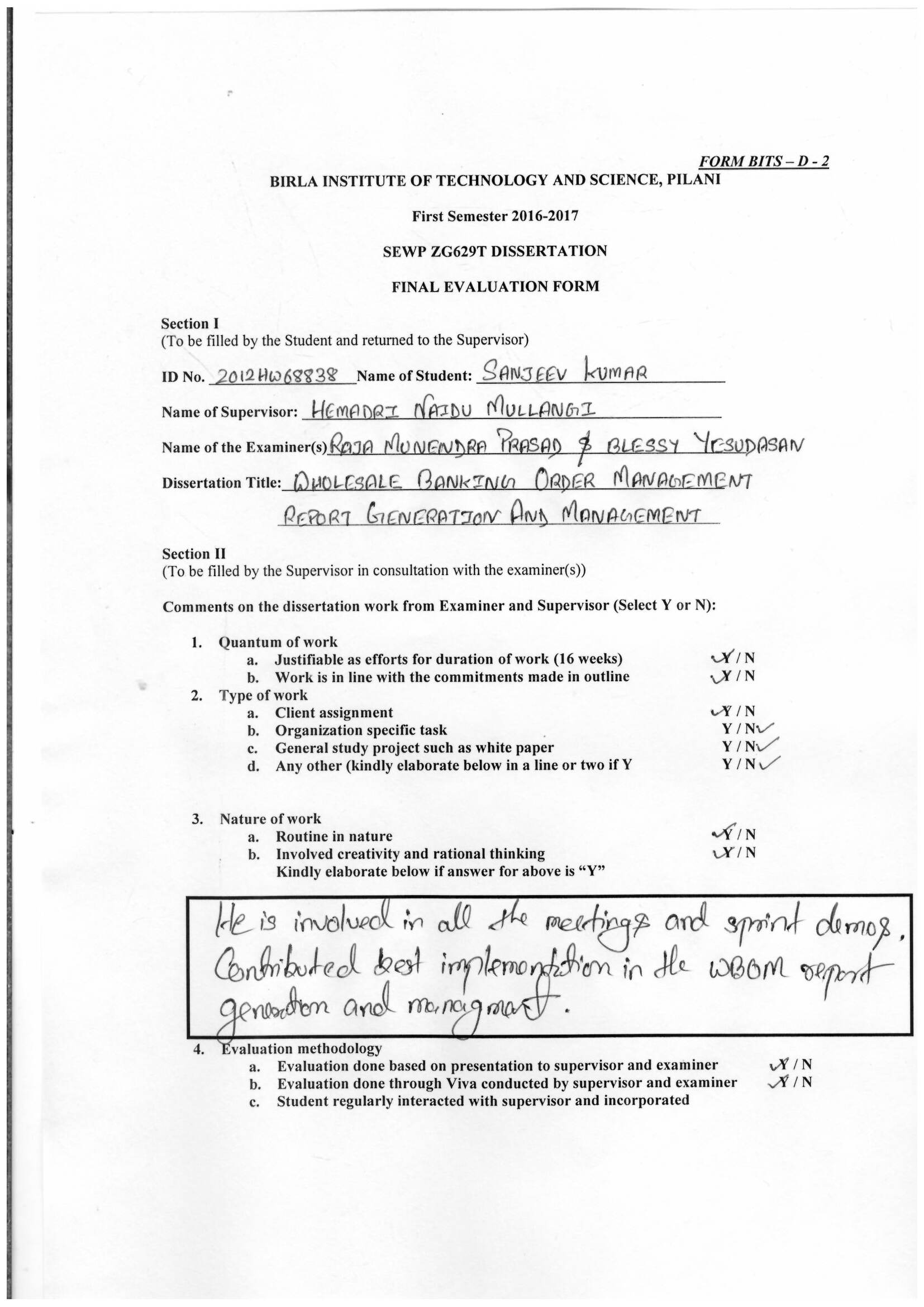
Dissertation work carried out at

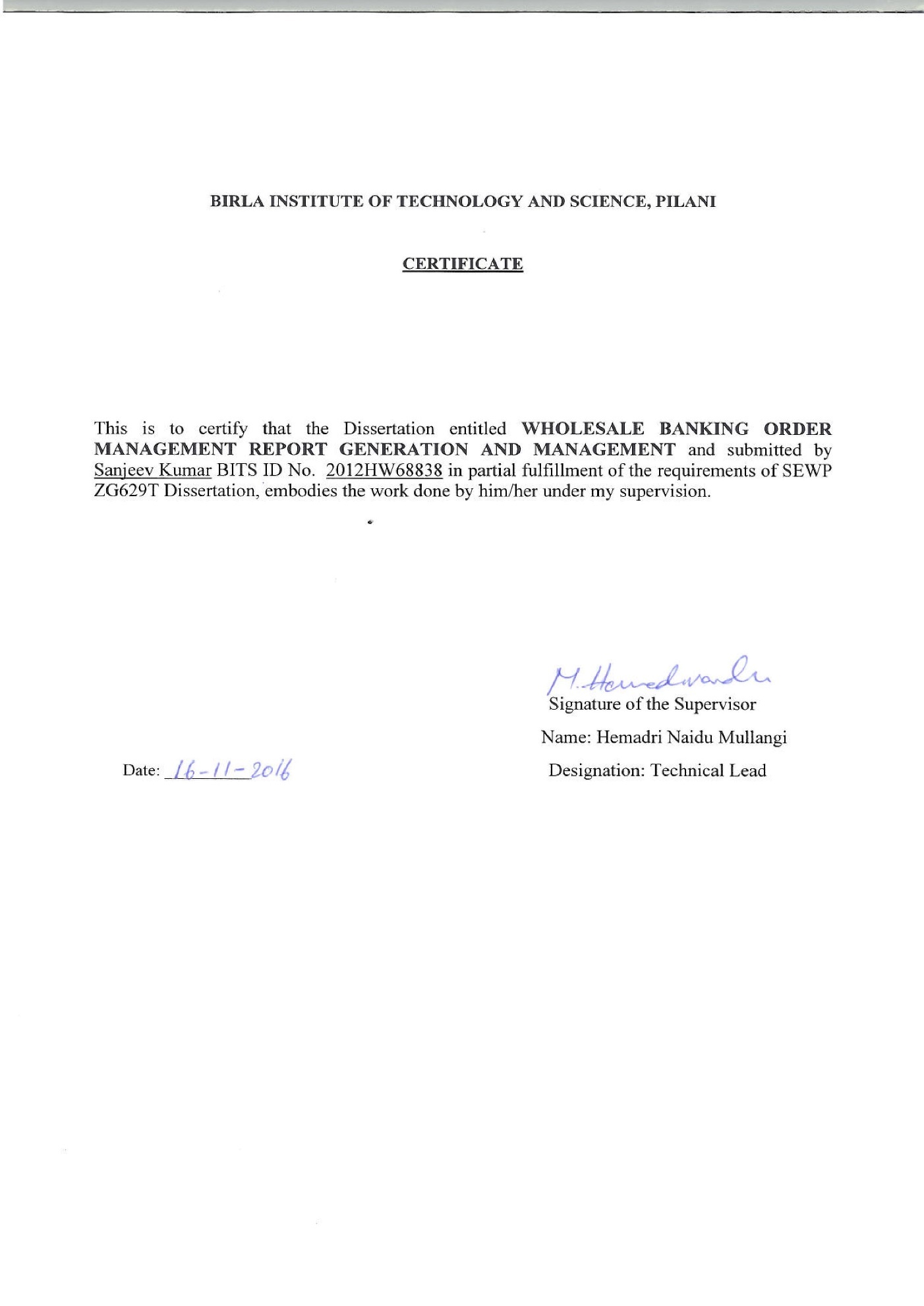
Wipro Technologies, Bangalore

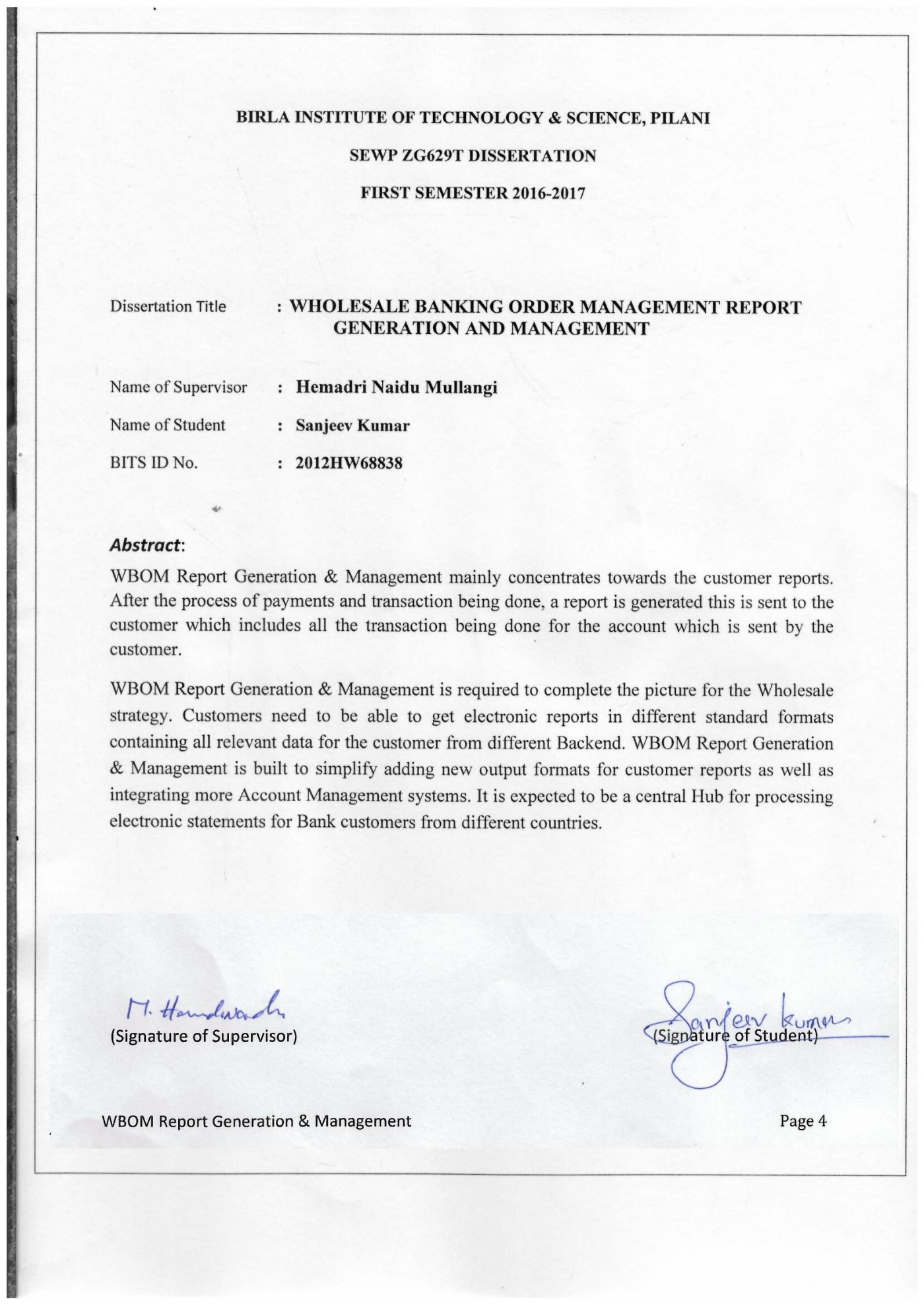
BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

PILANI (RAJASTHAN)

November, 2016







**ACKNOWLEDGEMENTS**

I would like to thank **Raja Munendra Prasad** and **Blessy Yesudasan** for their guidance, efforts and patience. They have been extremely supportive and co-operative in this whole effort. Their valuable comments have been immensely helpful in enhancing the quality of work.

I would like to express my sincere gratitude towards my team lead **Hemadri Naidu Mullangi** for examining my progress in spite of his busy schedule.

I would like to thank my **Colleagues** for their support and co-operation during the course of my project.

I am also thankful to **WIPRO** Technologies for giving me this opportunity and also making available all the resources required for this dissertation. Also I would like to thank our client for providing necessary tools required in completion of this work.

I am also thankful to **BITS**, Pilani and the Training department for all their efforts in organizing this course.

My sincere regards and thanks to **Prof. Ramesh Kumar**, **Sharana Gaddi** and WASE Co-coordinator for the information and support.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table of Contents** | | | | |
| **Contents:** | | | | |
| **1.** | [**Introduction: …………………………………………………………………………………………………………….……………………………………………**](#OLE_LINK_INTRO) | | | **10** |
|  | 1.1 | [About Wipro Technologies: ……………………………………………………………………………………………………………………………](#ABT_WIP_TECH) | | 10 |
|  | 1.2 | [Purpose of Project: …………………………………………………………………………………………………………………………………..……](#PRPOS_OF_PRJ) | | 10 |
|  | 1.3 | [Objective: ………………………………………………………………………………………………………………….…………………………………..](#OBJ) | | 11 |
|  | 1.4 | [Scope: ……………………………………………………………………………………………………………………………………………………………](#SCOPE) | | 11 |
|  | 1.5 | [Abbreviation and Acronyms: ……………………………………………………………………………………….…………………………………](#ABB_AND_ACRO) | | 12 |
| **2.** | [**Background: …………………………………………………………………………………………………………………..………………………………………**](#OLE_LINK_BACKGROUND) | | | **13** |
|  | 2.1 | [Process Description: ………………………………………………………………………………………………….…...................................](#PROCS_DESC) | | 13 |
|  | 2.2 | [Process Flow Diagram-Overview: …………………………………………………………………………………………………………………..](#PROCS_FLW) | | 14 |
|  | 2.2 | [Process flow Diagram of Receiving Account Justification (RAJ file): …………………………..…………………………………..](#PRC_FLW_RAJ) | | 15 |
| **3.** | [**Technology Used: ……………………………………………………………………………………………………….…………………………………………**](#OLE_LINK_TECH_USED) | | | **16** |
| **4.** | [**Functional Requirements: ……………………………………………………………………………………………………………………………………..**](#OLE_LINK_FUNC_REQ) | | | **17** |
|  | 4.1 | [Service Overview: ………………………………………………………………………………………………………………………………………….](#SERV_OVERVW) | | 18 |
|  | 4.2 | [Operation Overview: …………………………………………………………………………………….………………………………………………](#OPR_OVERVW) | | 18 |
| **5.** | [**The Design Phase: ………………………………………………………………………………………………………..……………………………………….**](#OLE_LINK_DESIGN_PHS) | | | **20** |
|  | 5.1 | [Operation Overview...........................................................................................................................................](#D_PH_OPR_OVRVEW) | | 20 |
|  |  | 5.1.1 | [Receive Account Justification: ………………………………………………………………………………………………………..](#RAJ_DEFINE) | 20 |
|  |  | 5.1.2 | [Operation Compose WBOM: ………………………….……………………………………………………………………………….](#OPR_CMP_WBOM_RPT) | 24 |
|  |  | 5.1.3 | [Deliver WBOM Report: ……………………………………………………………………………………………………………………](#DELVR_WBOM_REPOT) | 29 |
| **6.** | [**Low Level Design for the System: ……………………………………………………………..……………………………………………………………**](#OLE_LINK_LL_DESIGN) | | | **36** |
|  | 6.1 | [Project Structure: …………………………………….…………………………………………………………………………………………………..](#PRJ_STRUCTURE) | | 36 |
|  | 6.2 | [Project Interface Design: ……………………………………….……………………………………………………………………………………..](#PRJ_IF_DESIGN) | | 37 |
|  | 6.3 | [Data Design: …………………………………………………………….…………………………………………………………………………………..](#DTA_DESIGN) | | 38 |
| **7.** | [**Non-Functional Specification: ……………………………………………………………………………………………………………………………….**](#OLE_LINK_NFUNC_SPECS) | | | **41** |
|  | 7.1 | [SLA Specifications: ……………………………………………..………………………………………………………………………………………..](#SLA_SPCIFICATN) | | 41 |
|  |  | 7.1.1 | [Operation UploadDDContract, ReplicateProductAgreement, HandleTransformedDDContract : …….](#OPR_UDDC_RPA_HTDDC) | 41 |
|  | 7.2 | [QOS Specification: ……………………………………………………………………………………………………………………………………….](#QOS_SPECIFICATN) | | 42 |
|  |  | 7.2.1 | [Policy Requirements: ………………………………………………………………………………………………………………………](#POLICY_REQ) | 42 |
|  |  | 7.2.2 | [Queue Requirements: …….……………………………..……………………………………………………………………..………..](#QUEUE_REQ) | 42 |
|  |  | 7.2.3 | [Authentication: ……………………………………………………………………………………………………………………………...](#AUTH) | 42 |
|  |  | 7.2.4 | [Authorization: …………………………………..……………………………………………………………………………………………](#AUTHORIZTN) | 42 |
|  |  | 7.2.5 | [Integrity: ……………………………………………..…..…………………………………………………………………………………….](#INTEGRITY) | 43 |
|  |  | 7.2.6 | [Confidentiality: ………………………………………………………………………………………………………………………………](#CONFIDENTIALITY) | 43 |
|  |  | 7.2.7 | [Auditing and Logging: ……………………………………………………………………………………………………………………](#AUDIT_ND_LOGING) | 43 |
| **8.** | [**Build Standard and guidelines Review Checklist: …………………………………………………………………………………..……………….**](#OLE_LINK_STAND_CHKLST) | | | **45** |
| **9.** | [**Test Cases: ……………………………………………………………………………………………………………………………………………………………..**](#OLE_LINK_TEST_CASE) | | | **48** |
| **10.** | **Pl**[**an of Work: ………………………………………………………………………………………………………………………………………………………….**](#OLE_LINK_WORK_PLAN) | | | **53** |

Table of Figures

|  |  |  |
| --- | --- | --- |
| **Figure1.** | [**Process flow diagram Overview …………………………………………………………………………………….**](#IMG_PROCS_FLW_1) | **14** |
| **Figure2.** | [**Process flow of receiving account justification files using function Receive Account Justification.**](#IMG_PROCS_FLW) | **15** |
| **Figure3.** | [**Activity diagram of compose WBOM REPORT …………………………………………………………………...**](#IMG_ACT_DG_CMPSE_REPT) | **19** |
| **Figure4.** | [**Interaction Diagram – Collect WBOM (WBOM Update) ………………………………………………………**](#IMG_IT_DG_CLCT_WBOM_RPT) | **20** |
| **Figure5.** | [**Interaction Diagram – Compose WBO Report ……………………………………………………………….....**](#IMG_IT_DG_CMP_WBOM_RPT) | **25** |
| **Figure6.** | [**Interaction Diagram – WBOM Report Delivery ………………………………………………………………….**](#IMG_IT_DG_WBOM_REP_DELIVRY) | **30** |
| **Figure7.** | [**Package Structure of Project ………………………………………………………………………………………..**](#IMG_PROJECT_STRUCT) | **36** |
| **Figure8.** | [**Interface of Balance and Transaction ……………………………………………………………………………..**](#IMG_BALANCE_ND_TRANS) | **37** |
| **Figure9.** | [**WTX Request Message ………………………………………………………………………………………………...**](#IMG_WTX_REQ_MSG) | **38** |
| **Figure10.** | [**MT940 Tag & (Sub) Field Specification ……………………………………………………………………………**](#IMG_MT940_TAG) | **39** |
| **Figure11.** | [**MT940 Report Structure Sample File ………………………………………………………………………………**](#IMG_MT940_REPORT) | **40** |

1. Introduction

1.1 About Wipro Technologies

Wipro Technologies is the global technology services division of Wipro Ltd. (NYSE: WIT). Wipro Technologies offer a full portfolio of services across industries, delivering measurable business benefits to customers with six-sigma consistency. Wipro is the world's first software services company to attain SEI Level 5 certification for software quality. Wipro has business offices in North America, West Europe and Japan and development centers in India, USA and UK.

Wipro is a Service Organization engaged in a wide range of Product as well as consultancy services. Most organizations have recognized the need for offering a "Personalized Service" as an intelligent, cost effective way to retain their business. Customers and People management are the focus areas of Wipro.

Wipro is a **first CMMI Level 5 certified software services company** and the first outside USA to receive the **IEEE Software Process Award**.

In addition to the IT business, Wipro also has leadership position in niche market segments of consumer products and lighting solutions. The company has been listed since 1945 and started its technology business in 1980. Today, Wipro generates USD 7 billion (India GAAP figure 2010-11) of annual revenues. Its equity shares are listed in India on the Mumbai Stock Exchange and the National Stock Exchange; as well as on the New York Stock Exchange in the US.

1.2 Purpose of project

The objective of the service is WBOM Report Generation & Management mainly concentrates towards the customer reports. After the process of payments and transaction being done, a report is generated which is sent to the customer includes all the transaction being done for the account which is sent by the customer.

Balance and Transaction Reporting is required to complete the picture for the Wholesale strategy. Customers need to be able to get electronic reports in different standard formats containing all relevant data for the customer from different Backend. WBOM Report Generation & Management is built to simplify adding new output formats for customer reports as well as integrating more Account Management systems. It is expected to be a central Hub for processing electronic statements for Bank customers from different countries.

1.3 Objective

The objective of this dissertation is as mentioned below:

The proposed work is the study and implementation of the web service which *Composes Balance and Transaction Report.*

*Compose Report* is the creation of account statements and account notifications with a specific content, in a specific format and grouping of account statements and notifications in files.

Specific formats name:

**SWIFT MT940** is a common bank statement file across Europe. It was established by European   
bankers who needed a more efficient and secure system for interbank communications and   
transfer of funds and securities

**SWIFT MT942** gives a report of the debits/credits any given time of the day so that one can exercise better control. Other message under the same category would be the Balance Report (MT 941) which reports your intra-day balance is a standard structured SWIFT Customer Statement message. In short, it is an electronic bank account statement. It is an end of day statement file which details all entries booked to an account.

1.4 Scope

This function creates statements, reports and files based on input provided by two preceding functions: Manage statement work list and manually initiate account report. This means that the functional activity of selecting the agreements and accounts have to be used to create balance and transaction reports is not part of function Compose Report.

1.5 Abbreviations and Acronyms

|  |  |
| --- | --- |
| **Abbreviation Description** | **Description** |
| CDM | Canonical Data Model |
| XML | Extensible Markup Language |
| JMS | Java Messaging Service |
| XSD | XML Schema Document |
| WTX | WebSphere Transformation Extender (WebSphere Transformation Extender (WTX) is a powerful, transaction oriented, data integration solution that automates the transformation of high-volume, complex transactions without the need for hand-coding) |
| SOAP | Simple Object Access Protocol |
| SWIFT | stands for Society for Worldwide Interbank Financial Telecommunication |
| MT940 | Customer Statement End of the Day report (Message Type 940) |
| MT942 | Customer Statement End of the Day report Intraday Reports (Hourly basis) |

1. Background

2.1 Process Description

After each book run, WBOM Report Generation & Management receives an intraday account justification file for those accounts that have been intraday subscribed.

After each book day EOM receives a file from the account management system SAM (Server Account Management), which contains the current end-of-day account justifications of all Bank-accounts that have been end-of-day subscribed. To receive data from a Bank-account, WBOM Report Generation & Management has to subscribe to SAM to let SAM know it wants to receive data about this account.

The daily account justification file is processed by this function and the contents are stored in the report management database. The account justifications are part of the statement reports that will be generated for the Bank customers that have an agreement with Bank to receive statement reports.

2.2 Process flow Diagram-Overview



**Figure 1.** Process flow diagram Overview

2.3 Process flow diagram of Receiving Account Justification (RAJ File)



WBOM Domain

**Figure 2.**  Process flow of receiving account justification files using function Receive Account Justification

1. Technology Used:

|  |  |
| --- | --- |
| Phase | Technology to be used |
| * Requirements collection & Analysis | * Verbal, Questionnaires |
| * Coding and Unit testing | * Tools:  1. Tibco BW 2. Oracle Database 3. WTX  * Operating System:  1. UNIX 2. Windows |
| * Integration testing | * None |

1. Functional Requirements

WBOM Report Generation & Management mainly concentrates towards the customer reports. After the process of payments and transaction being done, a report is generated which is sent to the customer which includes all the transaction being done for the account which is sent by the customer.

WBOM Report Generation & Management is required to complete the picture for the Wholesale strategy. Customers need to be able to get electronic reports in different standard formats containing all relevant data for the customer from different Backend. WBOM Report Generation & Management is built to simplify adding new output formats for customer reports as well as integrating more Account Management systems. It is expected to be a central Hub for processing electronic statements for Bank customers from different countries.

**This WBOM Report Generation & Management Compose web service contains three operations to implement business functionality**

**Receive Account Justification:**

This operation receives the metadata information in xml file which contains Balances and transaction details. This service invokes WTX map (Using soap request message) to load the balance and transaction details into a staging table.

After Validation of Balance and transaction data, the service invokes MoveStagingToMain Procedure which will move the staging table records to main table. Customers will receive balance and transaction reports from Bank, which will help them to reconcile their payments.

**WBOM Report Generation & Management Compose Reports:**

This Operation will be used to compose in different report formats [MT940, MT942, etc.] and sends Metadata details in the SOAP request to WTX map. WTX Fetches balances and transaction from Oracle Database and processes it into files. It also generates the specified reports and places them into the XFB, FTP servers.

SWIFT MT940 is a common bank statement file across Europe. S.W.I.F.T. (or SWIFT) stands for   
Society for Worldwide Interbank Financial Telecommunication. It was established by European   
bankers who needed a more efficient and secure system for interbank communications and   
transfer of funds and securities.MT940 is End of the Day Customer report.

SWIFT MT942   gives a report of the debits/credits any given time of the day so that one can exercise better control. Other message under the same category would be the Balance Report (MT 942) which reports your intra-day balance is a standard structured SWIFT Customer Statement message. In short, it is an electronic bank account statement. It is a end of day statement file which details all entries booked to an account.

**WBOM Report Generation & Management Send Reports:**

After Compose This Operation will archive the generated report and places the file in the XFB server.

4.1 Service Overview

|  |  |  |
| --- | --- | --- |
| Service Name | Description | Service Context |
| WBOM Report Generation & Management | The purpose of function *Compose Report* is the creation of account statements and account notifications with a specific content, in a specific format and grouping of account statements and notifications in files. Statements (reports) such as MT940, MT942, etc. | NA |

**4.2 Operation Overview**

|  |  |  |  |
| --- | --- | --- | --- |
| **Operation Name** | **Operation Version** | **Description** | **Invocation Paradigm** |
| BalanceAndTransaction\_Initiate\_01 | 01 | This operation receives the file metadata information of an xml file containing account justification which inturn invokes the WTX map to load the data into a staging table. | Oneway |
| WBOMCustomerReport\_Release\_01 | 01 | This Operation will be used to Compose in different report formats [MT940, MT942, etc.] and sends the reports to customers. | Oneway |
| WBOMArchive\_Manage\_01 | 01 | The use case describes the implementation flow for Archiving a file and sending to Archiving system | Oneway |



***Figure 3.*** Activity diagram of compose WBOM Report Generation & Management

1. The Design phase
   1. Operations Overview
      1. Receive Account Justification

This operation receives the metadata information in xml file which contains Balances and transaction details. This service invokes WTX map (Using soap request message) to



**WBOM DB**

***Figure 4***. Interaction Diagram –Collect *WBOM Report Generation & Management (WBOM Update)*

|  |  |
| --- | --- |
| ID/Use Case : | Collect WBOM Report Generation & Management – WBOM Data Update |
| Description | Balance and Transaction data files received from SAM is used to update WBOM Database with the information after appropriate data enrichment and transformations to CDM format |
| Actor(s) | EOM: File share, TIBCO Composite Service, WTX, Logger DB, WBOM DB |
| Pre-conditions/ Assumptions |  |
| Post-conditions/Action | SAM Information is inserted in WBOM Tables |
| Input | SAM Update file received |
| Output | Updates to WBOM tables and WBOMFileIn MetaData table |
| Basic Flow | 1. SAM Update file received 2. TIBCO Composite service invoke UpdateLogDetails service (status: “Collect WBOM Report – WBOM Update: Initiated”) 3. Composite service invokes the right WTX service based on the file name to transform from SAM format to WBOM PDM. 4. SAM WTX Services (Maps) process the file based on the content type (RAJ, VB, BB) WTX fetches lookup files from EOM file share. 5. WTX fetches DB lookup & Meta data from WBOM DB. 6. WTX transforms the SAM format data to WBOM CDM format, and inserts it in staging tables. Only working days Balances are to be inserted. Any Balances related to ‘Non-working days’ should be ignored. Account information may need to be suppressed 7. WTX service checks if the file is duplicate or not based on comparing the File Identifier 8. WTX nvokes HandleTransformedBalanceAndTransactionInformation service.. 9. Composite service checks if Transformation is successful 10. Composite service invokes SP LOAD\_DATA to load data into WBOM DB. 11. SP LOAD\_DATA loads data from staging tables alongside with the NAL data from NALEnrichment table to main tables based on NAL Enrichment flag and Local Account number values. For all loaded accounts, all related WLI flags are updated to indicate that data is successfully loaded. 12. Composite service inserts a record in WBOMInputNotification table for the file processing status. The status is needed for checking and triggering Compose WBOM Service. 13. If the Inserted record in the WBOMInputNotification is transactions file “Account Justification”, then a message is sent to the ExtractTransactionEnrichment Queue to load Transactions Enrichments form OM DB “If needed; *i.e.* Tx type can be Enriched and Enrichment setting is True in the WLO data”. |
| Alternate Flow | 3a. If wrong file, raise exception and terminate flow.  8a. If duplicate, invoke Raise Incident and terminate processing  10a. If transformation is not successful Invoke Raise incident process and terminate processing  12a. If no NAL linkage or NAL enrichment data found, continue with main flow  13a. If load is unsuccessful, invoke Raise Incident process and terminate processing |
|  | For WTX Identified Exceptions:  Composite Service calls ExceptionHandler\_1 service to log Exceptions (and trigger any other automatic actions in the future) and decide whether to break or continue.  The decision to break or continue is based on the accumulation of the decisions of all WTX exceptions. |
| Exception handling | Technical Exceptions:  Technical Exception Handling Design for exception handling is taken care in ExceptionHandler\_1 service |

* + 1. Operation Compose WBOM Report

Compose WBOM Report

The process is triggered by a call to TIBCO InitiateBalanceAndTransaction service based on message received from WorkList Manager on a Queue. The WorkList Manager will trigger the Compose process once for each WorkList Object. Each WorkList Object would result in creation of one physical File, so the scope of the compose process is one file per agreement. Within a file, there will be as many statements as there are WorkList Items within the WorkList Object. The WorkList Manager would ensure all WorkList Items for which Zero statements are not to be reported are removed if there has been no updates to corresponding accounts for the book day. The WorkList Object contains the necessary keys to the Agreement, Agreement account, Statement Layout and report format specifications. The compose process extracts the WBOM data for accounts in scope of the WorkList Object from WBOM DB and stores in a file share. The Compose process extracts Agreement specific details based on the WorkList Object keys. The Compose process would trigger WTX by passing an XML message whose structure would have the File at a root level and under that would be blocks for each statement to be composed as per the WorkList Item details. WTX receives the trigger from the compose process and reads the WBOM data for the WorkList Item accounts and BTC data from corresponding files. WTX applies data and business rules as specified in ISO standards and arrives at an intermediate CDM from which it converts to the required output format for each statement. At the end of the compose process, the status of the WorkList Object is updated in the WorkList Object table in agreement DB as either success or failure. If the Compose process encounters any exception, then the exception path also needs to update the WorkList Object Status as (Failed) if the flow is to be aborted, unless the exception is handled in a way that the original flow can complete successfully.

The design intends to support different output format requirements. WTX maps should be built specific to different formats and the relevant ones executed based on the WorkList Item statement output format specifications.



**WBOM DB**

**Figure 5.** Interaction Diagram –Compose WBOM Report

|  |  |
| --- | --- |
| ID/Use Case : | Compose WBOM Report |
|  |  |
| Description | The use case describes the composition of WBOM reports as per report agreements and specifications by extracting WBOM report data from WBOM DB and applying transformations |
| Actor(s) | EOM: File share, TIBCO Composite Service, WTX, WBOM DB, Agreement DB, DB Utility |
| Pre-conditions/ Assumptions | The WorkList manager creates a WorkList Object for 1 File in an agreement that is ready for composition. |
| Post-conditions/Action | Report Created and corresponding Billing events are triggered |
| Input | WorkList events triggered when all data for a file is available |
| Output | CAMT 053 format file or MT940 format file. |
| Basic Flow | 1. TIBCO service InitiateBalanceAndTransactionReportRelease service is triggered 2. TIBCO service invoke UpdateLogDetails service (status: “Order Status Report: Initiated”) 3. The TIBCO service invokes TriggerUnload SP to extract Balance and Transactions data for each account in scope of the WorkList Object being processed as CLOB in Temp table Temp\_WLO\_CLOB. WLIs with Ignore field equal to ‘Y’ should be excluded from CLOB extraction. 4. TIBCO invokes WTX service TransformBalanceAndTransactionReport with the WLO Id & WLI Ids. 5. WTX invokes GetWBOMAgreementData to fetch agreement details for the request including agreement statement layout data, agreement account alias data, start statement numbers, File specification rules. 6. WTX selects the CLOB related to WLO in scope. 7. WTX reads BTC codes from BTC file. 8. WTX checks the report format to be generated (Camt 05x or MT94x or CAMT53 PSE) based on Statement Layout message format and version which is sent as an input in report agreement xml. As well as reported group in case of Camt 0.54. 9. WTX transforms the WBOM CLOB and applies data and format specification rules to create the required output format. 10. WTX has to extract extra data if the WLO is for Monthly statement, like starting Opening Balance and closing balances for the whole period covered in the monthly statement 11. WTX stores the output file in File share. 12. WTX calculates the report Meta Data (FileOut, ReportOut & StatementOut) and Billing Events. 13. WTX invokes CreateReportFileMetaData SP to update Meta data and Billing Events. 14. SP populates Metadata tables (WBOMFileOut, WBOMReportOut & WBOMStatementOut). Some fields are added to the ReportFileMetadata tables to be used for Monitoring and as identifiers for EAS Archive service .The same SP inserts new entries with Inserted status in the WBOMEvents table (only in case of CAMT) for the values calculated by WTX during report compose. 15. WTX writes the report file to the File share, with filename. 16. WTX invokes HandleTransformedBalanceAndTransactionReport service. 17. Composite service invokes SP UpdateWLObject to update the processing status of the triggering WLObject to ‘Composed’ if successful. 18. TIBCO service InsertStatusHistory is invoked to create a record in StatusHistory of the changed status of the WorkList Object. 19. TIBCO service invoke UpdateLogDetails service (status: “Balance and Transaction Report: Generated on <format>”). |
| Alternate Flow | Not specified. All logical transactional error that terminates the flow must have a routine in the exception path to update the status of the WorkList Object as Failed. |
| Exception handling | Technical Exceptions:  Technical Exception Handling Design for exception handling is taken care in ExceptionHandler\_1 service |

* + 1. Deliver WBOM Report

The flow is same as the POI OSR flow for report delivery, The service is split into two services to enable separation of WBOM/POI Building Blocks.

The triggering message for WBOM Report Delivery contains the Flow type (‘Send’ or ‘Resend’), WBOMFileOutId and report file metadata.

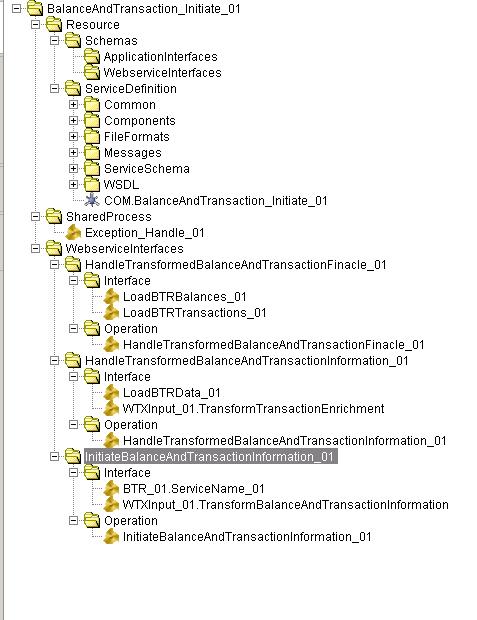
The service in return updates the WLObject status to ‘Delivered’ or ‘Failed’ as well as the WBOMFileSent table.



**WBOM** DB

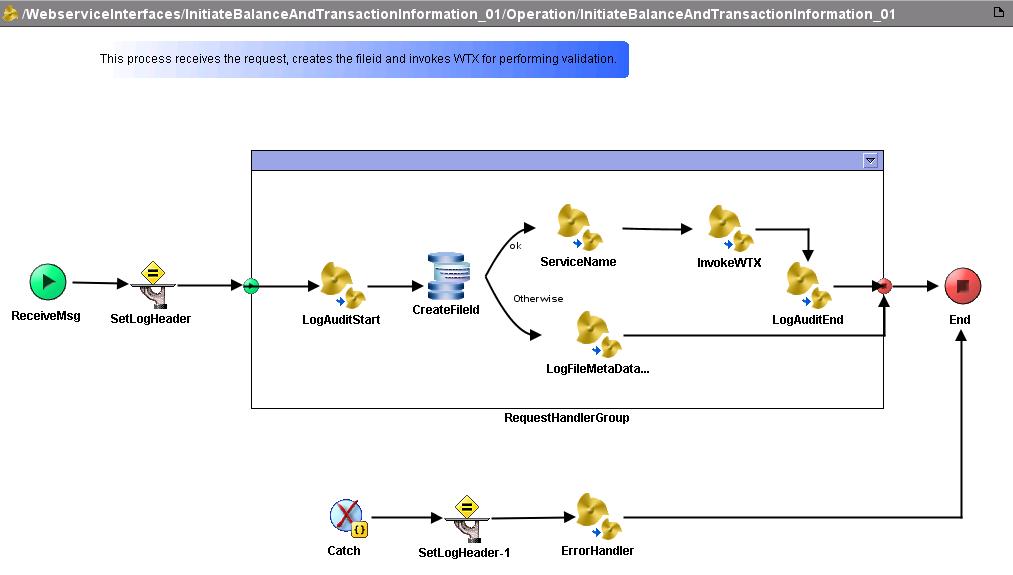
**Figure 6.** Interaction Diagram– WBOM Report Delivery

|  |  |
| --- | --- |
| ID/Use Case : | Deliver WBOM Report |
| Description | The use case describes packing the report file and sending to respective consumers as per Report Agreement and Channel Agreement. |
| Actor(s) | WBOM Report: File share, TIBCO Composite Service, WTX, WBOM DB, Agreement DB, Logger DB  Channel: FTP, SWIFTNet |
| Pre-conditions/ Assumptions | WBOMFileSent table filled with Delivery information and in case of Send Request The WLObject status updated in WBOM DB |
| Post-conditions/Action | (After successful send and archive) remove Report files from file share |
| Input | WBOMFileOutId, request type: ”Send, Resend”, ChannelAgreementId |
| Output | Encrypted and Compressed Report files sent to channel |
| Basic Flow | Deliver Report   1. TIBCO composite service operation WBOMCustomerReport\_Release\_01.ReceiveCustomerReportForDelivery\_01 operation is triggered 2. TIBCO composite service invoke *LoggerService\_1.UpdateLogDetails(Report)* operation on TIBCO proxy service (status: “Report: Initiate Delivery”) 3. The composite service invokes TIBCO operation to fetch Channel Agreement. – *ChannelAgreement\_01.GetChannelAgreementList\_01* operation on from EOM Agreement DB. 4. The composite service verifies the validity of channel agreement (as per rules specified) 5. The composite service invokes the Encryption and Compression module located in another TIBCO operation – *CustomerFile\_01.PackCustomerFile\_01* operation.(passing the channel agreement details) 6. The *CustomerFile\_01.PackCustomerFile\_01* module packs the customer report file based on information on channel agreement (compression, encryption details) 7. The module saves the packed file in file share 8. The module invokes the service and sends the metadata information about the packed customer file. – *CustomerReport\_Release\_01.HandlePackedCustomerFile\_01*operation. (status: “packed”) 9. The TIBCO service validates the response 10. The service creates a Log entry by invoking – *LoggerService\_1.UpdateLogDetails* *(Report)* operation on proxy service. (status: “Pack Successful”) 11. The composite service identify the channel to send the report 12. The composite service lookup the Channel XFB details 13. (If FTP) Service invokes proxy TIBCO service to trigger the sending of file via XFB to FTP channel. (*XFB\_Sender\_01\_SendFile\_01.SendFile\_01*) 14. (If SWIFTNet) Service invokes proxy TIBCO service to trigger the sending of file via XFB to SWIFTNet channel. (*XFB\_Sender\_01\_SendFile\_01.SendFile\_01*) 15. (If SWIFTNet) Service invokes SWIFTNet Channel adapter (*ChannelAdapter\_01.SendMessageToIRM\_01*) to send the MQ message to IRM to notify the file sent 16. Incase of Delivery to FTP or SWIFTNET Channel TIBCO will also send the rcvparm to XFB along with the folder name where the file need to be placed.  |  |  | | --- | --- | | Report | Folder Name | | MT940 | MT940 | | camt.053 | CAMT053 | | camt.054 | CAMT054 | |
| Exception/ Alternate Flow | 1. Request type is not “Send or Resend” or the 2. The Composite service calls *HandleException\_1.FormatExceptionMsg(ExceptionId, NVPairs*) to decide whether the process should break or not, where ExceptionId is “F00003”, Parameter is ‘RequestType’ , Domain “Send, Resend” & Value ‘Input value’   If response is Break   1. The Composite service calls *HandleException\_1.HandleException(ExceptionHandle*r) to decide whether the process should break or not, 2. Composite service updates the WBOMFileSent table with the failure reason and description as well as WBOMWorkListObject and WBOMStatusHistory |
| 1. WBOMFileOutId, ChannelAgreementId or WLOId is missing or equal to zero 2. The Composite service calls *HandleException\_1.FormatExceptionMsg(ExceptionId, NVPairs*) to decide whether the process should break or not, where ExceptionId is “F00001”, Parameter is ‘WBOMFileOutId’ 3. The Composite service calls *HandleException\_1.HandleException(ExceptionHandle*r) to decide whether the process should break or not, where ExceptionDefinetionId is ‘F00001’,   If response is Break   1. Composite service updates the WBOMFileSent table with the failure reason and description as well as WBOMWorkListObject and WBOMStatusHistory |
|  | 1. Channel Agreement, WBOMFileOut, WLO entry is not found 2. The Composite service calls *HandleException\_1.FormatExceptionMsg(ExceptionId, NVPairs*) to decide whether the process should break or not, where ExceptionId is “F00002”, TableName is “AGRAgreementChannel” Parameter is “AgreementChannelId”, Value is ‘AgreementChannelId’. 3. The Composite service calls *HandleException\_1.HandleException(ExceptionHandle*r) to decide whether the process should break or not, where ExceptionDefinetionId is ‘F00002’, 4. If response is Break 5. Composite service updates the WBOMFileSent table with the failure reason and description as well as WBOMWorkListObject and WBOMStatusHistory |
|  | 1. WorkListObject is not found 2. The Composite service calls *HandleException\_1.FormatExceptionMsg(ExceptionId, NVPairs*) to decide whether the process should break or not, where ExceptionId is “F00002”, TableName is “WBOMWorkListObject” Parameter is “WLOId”, Value is ‘WLOId’. 3. The Composite service calls *HandleException\_1.HandleException(ExceptionHandle*r) to decide whether the process should break or not, where ExceptionDefinetionId is ‘F00002’,   If response is Break   1. Composite service updates the WBOMFileSent table with the failure reason and description |
|  | 1. ReportFile is not found 2. The Composite service calls *HandleException\_1.FormatExceptionMsg(ExceptionId, NVPairs*) to decide whether the process should break or not, where ExceptionId is “F00004”, Parameter is “FileName”, Value is ‘LogicalfileName’, Parameter is “FilePath”, Value is ‘ReportFilePath’, 3. The Composite service calls *HandleException\_1.HandleException(ExceptionHandle*r) to decide whether the process should break or not, where ExceptionDefinetionId is ‘F00004’,   If response is Break   1. Composite service updates the WBOMFileSent table with the failure reason and description as well as WBOMWorkListObject and WBOMStatusHistory |
| Technical Exception | Technical Exceptions:  Technical Exception Handling Design for exception handling is taken care in ExceptionHandler\_1 service interaction pattern. Taken care in |

1. Low Level Design for the System
   1. Project Structure

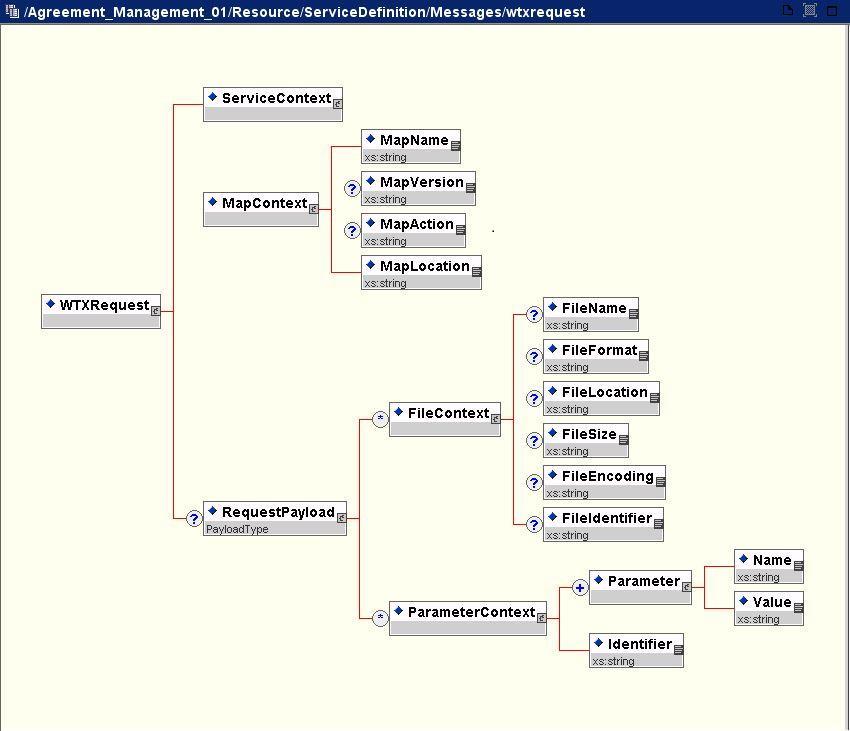
***Figure 7.*** *Package structure of the Project*

* 1. Project Interface Design

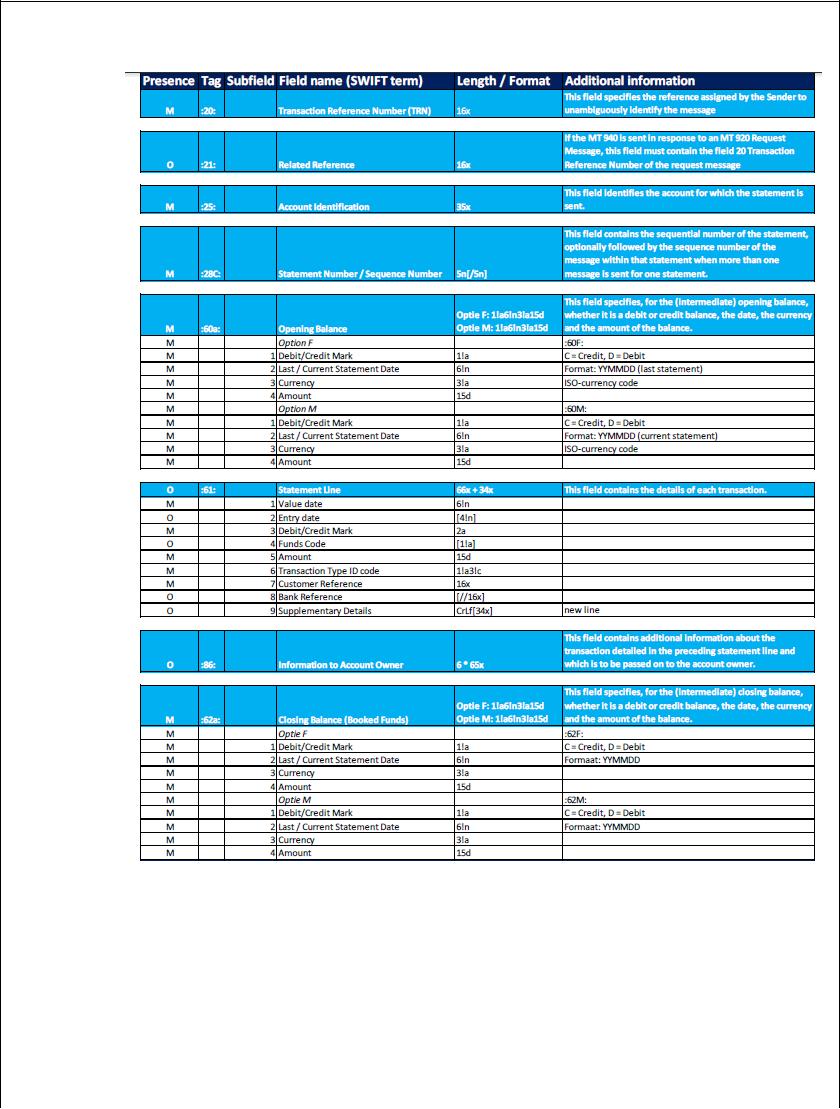


***Figure 8.*** *Interface of Balance and Transaction*

* 1. **Data Design**



***Figure 9.*** *WTX Request message*



***Figure 10.*** *MT940 Tag & (Sub) Field Specifications*



***Figure 11.*** *MT940 Report Structure sample file*

1. Non-Functional Specifications
2. **SLA Specifications** 
   * 1. Operation UploadDDContract, ReplicateProductAgreement, HandleTransformedDDContract :

|  |  |  |
| --- | --- | --- |
| Average Response Time | Average Response Time under normal conditions (In sec) | Not available at the time of design |
|  | Max response time when alarm should be raised (in sec) | Not available at the time of design |
| Availability | Availability Time in hours/week or hours/month or hours/year | 99% - 24/7 |
| Downtime | Expected maximum Value (hours/week or month or year) under normal conditions | 3-6 hrs. per week |
| Time Duration (Ex: Sunday morning 3:00 AM to 4:00 AM) | Not available at the time of design |
| Message Size | Average Message Size | Not available at the time of design |
| Max Message Size (Kb/Mb) | Not available at the time of design |
| Average Number of Requests | Average Number of Requests | Not available at the time of design |
| Peak number of request/sec | Not available at the time of design |
| Alarm Value (Requests/sec) when an alert notification should be sent | Not available at the time of design |
| Average Number of Faults | Expected Value (Faults/sec) under normal conditions | 0 |
| Alarm Value (Faults/sec) when an alert notification should be sent | 3 |
| Delivery Type | Guaranteed Delivery Required (Y/N) | Y |
| Once and Only Once delivery Required (Y/N) | Not available at the time of design |
| Number of Retries | Retry Count | 3 |
| Retry Interval | 3 sec |

1. **QOS Specifications**

**7.2.1** **Policy Requirements**

|  |  |
| --- | --- |
| **Parameter Name** | **Value** |
| Security Required (Yes/No) | Yes |
| Service Status Monitoring Required (Yes/No) | Yes |
| Alert Notification Required in case of error/fatal scenarios (Yes/No) | Yes |
| Alerting mechanism (email/SMS etc.) | Not Applicable |

**7.2.2** **Queue Requirements**

|  |  |
| --- | --- |
| **Parameter Name** | **Value** |
| |  | | --- | | Queue Monitoring Required (Yes/No) | | No |
| Maximum queue size when alerts should be sent (Yes/No) | No |
| Maximum Age of messages in queue in hrs./days | 10 min |

**7.2.3** **Authentication**

|  |  |
| --- | --- |
| **Parameter Name** | **Value** |
| Authentication Mechanism | As per BIS Security Guidelines |

**7.2.4** **Authorization**

As per BIS Security Guidelines.

**7.2.5** **Integrity**

|  |  |
| --- | --- |
| **Parameter Name** | **Value** |
| Message Integrity Required (Yes/No) | No |
| Message parameters for which integrity is required | Not Applicable |

**7.2.6** **Confidentiality**

|  |  |
| --- | --- |
| **Parameter Name** | **Value** |
| Transport level confidentiality (SSL) required (Yes/No) | Yes |
| Message level confidentiality Required (Yes/No) | No |
| Message parameters for which integrity is required | Not Applicable |

**7.2.7** **Auditing and Logging**

* **Auditing**

|  |  |
| --- | --- |
| **Parameter Name** | **Value** |
| Message level Auditing required (Yes/No) | No |
| Audit Request (Yes/No) | No |
| Audit Response (Yes/No) | No |
| Audit Fault (Yes/No) | No |
| Audit specific Message part  (provide the specific message parts to be audited) | Not Applicable |

* **Logging**

|  |  |
| --- | --- |
| **Parameter Name** | **Value** |
| Message logging required (Yes/No) | No |
| Log archival required (Yes/No) | Not Applicable |
| Log archival frequency (Daily/Weekly/Monthly) | Not Applicable |
| Log purging frequency (Monthly/Quarterly/Yearly) | Not Applicable |

1. Build Standard and guidelines Review Checklist

|  |  |  |  |
| --- | --- | --- | --- |
| **Technology** | **Checklist Item / Findings** | **Is the standard** |  |
|  |  | **Applicable (Y/N)?** |  |
|  |  |  |  |
| Tibco | Have your referenced BIS | Yes |  |
|  | Standards and Guidelines, Rule |  |  |
|  | Books, Templates: |  |  |
|  | • ISS\_STS\_BISging\_ |  |  |
|  | Guidelines |  |  |
|  | • ISS\_STS\_BIS\_ |  |  |
|  | Implementation Guidelines |  |  |
|  | • ISS\_STS\_BIS\_ |  |  |
|  | PolicyManager Implementation |  |  |
|  | Guidelines |  |  |
|  | • ISS\_STS\_BIS\_ |  |  |
|  | Registry Implementation |  |  |
|  | Guidelines |  |  |
|  | • ISS\_STS\_BIS\_ |  |  |
|  | Guidelines |  |  |
|  | • ISS\_STS\_BIS\_ |  |  |
|  | Implementation Guidelines |  |  |
|  | • ISS\_STS\_BIS\_ |  |  |
|  | Decisions Implementation |  |  |
|  | Guidelines |  |  |
|  | • ISS\_STS\_BIS\_ |  |  |
|  | Implementation Guidelines |  |  |
|  | • ISS\_STS\_BIS\_ |  |  |
|  | Adapter Implementation Guidelines |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Tibco | Verify that the base(Main) folder name of the BW project should be the "ServiceName\_Context\_Version | Yes | |
| Tibco | WebserviceInterfaces-Verify that all operations folders are under  ServiceName\_Context\_Version/ WebserviceInterfaces folder | Yes | |
| Tibco | Operation -Folder should contain main operation - Verify that ServiceName\_Context\_Version/ WebserviceInterfaces/Operation\_Version/ Operation  Process Name**:** OperationName\_Version | Yes | |
| Tibco | Interface -Folder should have all sub processes for external service calls. e.g. Proxy service or system interfaces.ServiceName\_Context\_Version/ WebserviceInterfaces/Operation\_Version/Interface  Process Naming(composite) | Yes | |
| Tibco | Orchestration - Folder should have the orchestration flow. The flow would be to call the Interface processes as per the requirements. This is applicable for composite services. | Yes | |
| Tibco | Shared Processes - All sub processes that are common across operations to be placed under ServiceName\_Context\_Version/SharedProcesses folder. | Yes | |
| Tibco | Look and Feel of all process flows should be proper.  a) Basically all the tasks should be aligned in a process flow.  b) No crossing of lines in a flow. | Yes | |
| Tibco | For folder and process naming there should be no underscore other than for versions. | Yes | |
| Tibco | All external applications related schema should be available under “Schemas/ApplicationInterfaces” and final CDM schema should be available under “Resource/ServiceDefinition” folder. | Yes | |
| Tibco | Verify that global variables are grouped into [Domain] , "Constants" and "SharedUtilities" | Yes | |
| Tibco | Verify if all global variables for password are of password type i.e. encrypted. (\*\*\*\*\*\*\*) | Yes |
| Tibco | Transport global variables should follow below:   1. Placed at SharedUtilities/Resource/Transport 2. No "Resources" global variable group should be there. 3. SharedUtilities/Resource/Transport/EMS/FF and respective RR GVs to be defined. | Yes |
| Tibco | Soap Action - <servicename>\_<servicecontext>\_<version>.<operationname>\_<version> | Yes |
| Tibco | NameSpaceRegistry - For each process Configuration -> Namespace Registry   1. Make sure under Namespace, Schema Import, WSDL Imports there are no unused namespace. If there remove them from here. 2. . If you use template generator to create template and not copy any processes or activates this should not happen. But if you have done make sure you delete unwanted namespaces. 3. Even of you have created Test folders those namespaces can also be retained even after deleting the test folder so make sure no such namespaces, schema and wsdl imports exist. | Yes |

1. Test cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Test Steps** | **Expected Behavior** | **Status (Pass/Fail)** |
| Pre-Test Validation | | | |
| 1. | Validating the Balance\_And\_Transaction\_Management XSD’s | XSD's should be validated with 0 errors and 0 warnings. | Pass |
| 2. | Verify the target namespaces as per the standards in WSDL. | The namespaces should be according to naming standards. | Pass |
| 3. | Checking the Concrete WSDL created properly. | The concrete WSDL should be created successfully. | Pass |
| Transport Layer Validation | | | |
| 4. | Checking whether the required queue are created in EMS server | Required queues and connection factories are created. | Pass |
| 5. | Verifying JMS connection palette parameters and also to verify the parameters are not hard coded. | 1. All parameters should be mapped from defined Global Variables.  2. Topic connection factory field should be blank.  3. Queue connection factory should be mapped from defined Global Variable. | Pass |
| 6. | Testing the JMS connectivity with the EMS server. | A message should be displayed as JMS connection factory <connection factory name> Connection test successful. | Pass |

|  |  |  |  |
| --- | --- | --- | --- |
| Process Flow Validation | | | |
| BalanceAndTransaction\_Initiate\_01 | | | |
| 7. | Checking whether all the mandatory elements as inputs are populated. | All the mandatory fields should be mapped in the BalanceAndTransaction\_Initiate\_01 service. | Pass |
| 8. | Checking whether BalanceAndTransaction service getting triggered whenever service consumer sends the request | BalanceAndTransaction\_Initiate\_01 service should be up and Running | Pass |
| 9. | Checking whether BalanceAndTransaction\_Initiate\_01 operation sends proper request to WTX(with Load Id) to invoke TransformFile operation of WTX service | WTX service was invoked successfully. | Pass |
| 10. | Checking whether BalanceAndTransaction\_Initiate\_01 operation invokes TransformFile operation of WTX service successfully. | WTX service was invoked successfully. | Pass |
| 11. | Check whether Account justification loaded in the WBOM data base. | Account Justifications are collected in the WBOM data base. | Pass |
| Process Flow Validation | | | |
| WBOMCustomerReport\_Release\_01 | | | |
| 12. | Checking whether the WBOMCustomerReport\_Release\_01 operation of Balance\_And\_Transaction Service gets triggered when the Service Consumer() sends a request | The WBOMCustomerReport\_Release\_01 operation of Balance\_And\_Transaction service should be invoked successfully. | Pass |
| 13. | Check whether the WBOMCustomerReport\_Release\_01 operation invokes the DB procedure 'CHECK\_DOUBLE\_DELIVERY’ to fetch the contract details. | 'CHECK\_DOUBLE\_DELIVERY' JDBC stored procedure was invoked successfully | Pass |
| 14. | Check whether 'CHECK\_DOUBLE\_DELIVERY' is validating the each received file and reports and one error message in Logger DB in case file received more than one time. | Error is logged into the Logger DB | Pass |
| 15. | If the file is duplicate then check whether only one copy of it will be loaded in the Data base | Only one copy of the Account Justification is loaded | Pass |
| 16. | If the Agreement doesn’t exist – Party doesn’t exist then check whether party and agreements gets added to it. | Party and the agreement is added to the database. | Pass |
| 17. | If the file contains 5 agreements each with semantic validation error then check whether all the agreements report an error status with one reason in Logger DB. | Agreements report an error status with reason in Logger DB. | Pass |
| 18. | Check whether FetchErrorReport fetches all the contract details of a file with status 'error' | Contract details (IBAN,CREDITORSCHEMEID, IDENTIFICATIONCODEOFTHESCHEME) with the status error will be returned | Pass |
| 19. | Check whether Cash Transaction Details and Account details are loaded | Transaction details and account details will be loaded in the data base. | Pass |
| 20. | Check whether reports are generated (MT 940, camt formats etc.) after the completion of compose process. | Reports will be in the respective FTP folders. | Pass |
| Process Flow Validation | | | |
| WBOMArchive\_Manage\_01 | | | |
| 21. | Check whether WBOMArchive\_Manage\_01 operation of the Balance And Transaction gets triggered at the given time. | WBOMCustomerReport\_Release\_01 operation of the Balance And Transaction should get invoked in the given time as it’s the Timer process. | Pass |
| 22. | Check whether WBOMArchive\_Manage\_01 operation of the Balance And Transaction gets triggered when reports are generated. | WBOMArchive\_Manage\_01 operation of the Balance And Transact should get invoked. | Pass |
| 23. | Check whether WBOMArchive\_Manage\_01 operation of the Balance And Transaction is archiving the generated report. | WBOMArchive\_Manage\_01 operation will place archived reports in the XFB folder after file is delivered. | Pass |
| Error Handling | | | |
| BalanceAndTransaction\_Initiate\_01 | | | |
| 24. | Check whether BalanceAndTransaction\_Initiate\_01 operation is able to handle wrong request sent by the client -- (Invalid ESB Service Request) | 1. The Balance And Transaction operation should throw SOAP Fault Message.  2. The ErrorHandler process should processed.  3. Error details are logged. | Pass |
| 25. | Check whether the BalanceAndTransaction\_Initiate\_01 is able to send appropriate message when the EMS server is down after the request is received successfully. | 1. The Balance And Transaction operation after receiving the message should throw an error and take error transition (Catch the error and take ErrorHandler process).  2. The ErrorHandler process should processed.  3. Error details are logged. | Pass |
| Error Handling | | | |
| WBOMCustomerReport\_Release\_01 | | | |
| 26. | Check whether WBOMCustomerReport\_Release\_01 operation is able to handle wrong request sent by the client -- (Invalid ESB Service Request) | 1. The Balance And Transaction operation should throw SOAP Fault Message.  2. The ErrorHandler process should processed.  3. Error details are logged. | Pass |
| 27. | Check whether the WBOMCustomerReport\_Release\_01 is able to send appropriate message when the EMS server is down after the request is received successfully. | 1. The Balance And Transaction operation after receiving the message should throw an error and take error transition(Catch the error and take  ErrorHandler process).  2. The ErrorHandler process should processed.  3. Error details are logged. | Pass |
| 28. | Check whether the WBOMCustomerReport\_Release\_01 service handles the error when there is Technical Error. | 1. Agreement\_Management operation should throw an error.  2. The ErrorHandler process should processed.  3. Error details are logged. | Pass |

1. Plan of Work

**10.1 Schedule**

The following table summarizes the actual progress of work against the planned dates presented in the initial outline documentation of the dissertation.

|  |  |  |  |
| --- | --- | --- | --- |
| **Serial Number of Task** | **Tasks or Subtasks to be done** | **Planned duration in weeks** | **Specific Deliverable in terms of the project** |
| 1. | Requirements Gathering and review and estimation | 2 week | Requirements Document, FP estimation documents, Integration Testing document |
| 2. | Preparation of Design and Design Review | 3 weeks | Design Document, system test case document |
| 3. | Coding and unit testing | 4 weeks | Unit tested code and results |
| 4. | Final Release | 2 weeks | Completely deployed system |
| 5. | Integration and System testing | 2 weeks | Integration tested Code and results |
| 6. | Completion of Dissertation Report | 1 week | Dissertation Report |

1. **REFERENCES**

Project and Quality Related Documents and references

* Software Engineering: A Practitioner's Approach (Roger S. Pressman)
* Java The Complete Reference (Herbert Scheldt)
* Client Requirements Document.
* Client Secure Practices Document.
* <http://knet.wipro.com/docknet>
* <http://www.techonthenet.com/>
* <http://velociq.wipro.com/>

1. **DEVELOPMENT STANDARDS**

The programming standards to be followed by the application developers

* Technical standards (Tibco processes)
* Standard guidelines / style guides / formats for screens
* Use of development techniques / tools

guidance. Sjat ENT 1 1