E0067- (LM) POD (Proof Of Delivery) DELMAN to SAP S4

SAP Functional Specification Document

Interface

Object ID	E0067
Object Type	Interface
System	☑ SAP ☐ Satellite App: {Application Name}
Object Title	Description
Stream Area	☐ Sales & Promotion ☑ Distribution ☐ Warehouse ☐ Procurement ☐ Production Planning ☐ Quality Management ☐ Plant Maintenance ☐ Finance Accounting ☐ Finance Controlling ☐ Project System ☐ Success Factor ☐ Master Data Governance
Complexity	☐ Low ☑ Medium ☐ High ☐ Very High

Guidelines for Writing Form Functional Specification

1. The content in BLUE is the synopsis of what the section of FSD should contain. Please remove the content in blue before updating the respective sections.
2. Update “Table of contents” section before base lining/delivering/updating of the Functional Specification Design.
3. Section 1 – “Document History” is mandatory to log initial and further changes to the Functional Specification Design. Please highlight in different color if, critical content pertaining to a Change Request is being updated.
4. Enable Track changes if to modify Baseline Functional Specification Design.
5. Process flow diagram should be mentioned in detail.
6. Sections 6 should capture all possible business test cases. This will become the base for technical unit testing (TUT) and functional unit testing (FUT).
7. Please remove all sample texts and sample attachments after writing the FS. They are only indicative in nature.

1 Document Control Information

1.1 Document Edit History

Version	Date	Additions/Modifications	Prepared/Revised by
Version 1	24-Mar-2025	Initial Design	Robby
Version 2	15-May-2025	Add ERNAM in Return T_LIST and logic getting from CDPOS and CDHDR	Robby

1.2 Document Review and Sign Off

The individuals listed here will be required to review and approve this document.

Reviewed By: The “Reviewed By” signature indicates the individual(s) who reviewed this document for content and clarity, and to the best of their knowledge, this document satisfactorily achieves the purpose and scope defined herein:		
	Name	**Title/Role** <Designation, Department>
Reviewed By	Joseph Jonathan	
Signature	@Joseph Jonathan	Date: Apr 14, 2025
Reviewed By		
Signature		Date:

Approved By: The “Approved By” signature indicates the individual(s) who approved this document for content and clarity, and to the best of their knowledge, this document complies with corporate policies and procedures:		
	Name	**Title/Role** <Designation, Department>

Approved By	Amendha Steviany Anggara	
Signature	@Amendha Steviany Anggara	Date: Apr 14, 2025
Approved By		
Signature		Date:

2 Functional Specification Details

2.1 Impacted Sub-Process

E-020-040 LM - Route Settlement Var.0 (PDA)

2.2 Assumption

- DELMAN system will only send the record once
- For FULL SKR due to “TOKO TUTUP” will only be sent to SAP S4 during the Security IN.

2.3 Risks

- Delay on waiting time for Truck to come from Transport to Warehouse. Manual POD is taking longer time due to now is batch manage.

3 Operational Considerations

The following sections outline the requirements for the interface object. The requirements, business rules and design specifications are combined in this document to provide a comprehensive view of the functional design.

3.1 Data Source

DELMAN has few data sources, however middleware (webservices) will group it into 2 groups of interface structure.

ZSDT_PODALL

Field Name	Data Type	Sample
TOR_ID	CHAR (20)	100147109
VBELN	CHAR (10)	2245907510
KUNNR	CHAR (10)	18765
GRUND	CHAR (4)	H0
file_path_toko	TEXT	49189_93372469_250301.jpg|140897_93373029_250308.jpg

ZSDT_PODPARTIAL

Field Name	Data Type	Sample
TOR_ID	CHAR (20)	100147109
VBELN	CHAR (10)	2245907510
POSNR	NUMC (6)	10
MATNR	CHAR (40)	10025
LFIMG	QUAN (17)	36
VRKME	UNIT	BOX

GRUND	CHAR (4)	H0
PODAT	DATS (10)	20250210
POTIM	TIMS (8)	230130
KUNNR	CHAR (10)	18765
file_path_toko	TEXT	49189_93372469_250301.jpg|140897_93373029_250308.jpg

3.2 Trigger

Upon call from DELMAN (Source system).

3.3 Processing Options

Inbound/Outbound

- [x] Inbound
- [ ] Outbound

Processing Mode

- [ ] Batch
- [x] Real Time
- [ ] Near Real Time

Processing Type

- [x] Synchronous
- [ ] Asynchronous

Require Middleware

- [x] Yes
- [ ] No

Interface Type

- [ ] RFC
- [ ] IDOC
- [ ] Direct Database
- [x] API
- [ ] File
- [ ] CDC (Change Data Capture)
- [ ] Other (if other, describe here) _____

Frequency

- [ ] Annually
- [ ] Quarterly
- [ ] Monthly
- [ ] Weekly
- [ ] Daily
- [ ] On Demand
- [x] Other _Every 5 Minutes___

3.4 Dependencies

- POD Relevant configuration

3.4.1 Environment / Configuration

- POD Relevant configuration

3.4.2 Development Dependencies

- [TBA]-Send Freight Data to DELMAN

3.4.3 Run / Execution Dependencies

Not Applicable since the Interface will always run after the Delivery has been Goods Issued by Warehouse.

3.5 Expected System Load

How often will the development object be executed?

Average Load-

Peak Load-

4 Functional Design Considerations

Interface will run on real time and store the data with process status into table ZSDT_PODINT (POD Quantity from DELMAN).

Since real time processing might fail, this RICEFW object will have User Interface to view or reprocess the data. SAP will also need to provide the API so webservice can call the API directly upon background job.

4.1 Interface Details

Program will get the Interface data from DELMAN and it is grouped into 2 Remote function call (RFC).

1. **ZSDT_PODALL**

This Interface process based on header delivery level, which means all accepted or rejected by customer.

Program will check on POD Date for accepted item and POD table (TVPOD) for processing or skipping the data.

2. **ZSDT_PODPARTIAL**

This Interface process based on item delivery level, which means each delivery has some rejected quantity by customer.

Program will check on POD table (TVPOD) for processing or skipping the data.

ZSDT_PODALL → need to rename the FM

```
LOOP at T_LIST
    //Call direct Update PODAT + POTIM if still initial
    CLEAR MANUAL. //set MANUAL as blank and change to X when Delivery Processed before this interface
    SELECT LFART,WADAT_IST,PODAT,POTIM FROM LIKP WHERE VBELN =T_LIST-VBELN
    //Error for Delivery Not Exist
    IF SY-SUBRC <> 0
        T_LIST-SUBRC = 100
        T_LIST-MESSAGE = "Delivery Order not found"
    ELSE
        STATS=1.
        T_LIST-PODAT = LIKP-WADAT_IST.
        T_LIST-LFART = LIKP-LFART.
        IF GRUND = INITIAL
            SELECT LFART,WADAT_IST,PODAT,POTIM FROM LIKP WHERE VBELN =T_LIST-VBELN
            IF LIKP-PODAT = INITIAL
                CALL BAPI WS_DELIVERY_UPDATE
                    VBKOK_WA-VBELN_VL = T_LIST-VBELN;
                    VBKOK_WA-PODAT= LIKP-WADAT_IST;
                    VBKOK_WA-POTIM= INITIAL;
                    VBKOK_WA-KZPOD= 'B';
                    COMMIT = X.
                GET Return Message
                IF SY-SUBRC <> 0
                    T_LIST-SUBRC = 100.
                    T_LIST-MESSAGE = BAPI_RET_MESSAGE
                ELSE
                    T_LIST-MESSAGE = "POD updated successfully"
                    STATS=2.
                ENDIF
                    T_LIST-ERNAM = SY-UNAME.
                    UPDATE MANUAL,LFART,PODAT, POTIM, SUBRC, MESSAGE INTO T_LIST
            ELSE
                T_LIST-SUBRC = 100
                T_LIST-MESSAGE = "POD has been completed manually"
                SELECT CHANGENR FROM CDPOS WHERE OBJECTID = T_LIST-VBELN AND FNAME='PODAT' AND VALUE_NEW <> '000000'.
                SELECT USERNAME,UDATE,UTIME INTO LT_CDHDR FROM CDHDR WHERE OBJECTID = T_LIST-VBELN AND CHANGENR IN CDPOS-CHANGENR.
                SORT LT_CDHDR DESCENDING BY UDATE,UTIME
                SELECT SINGLE USERNAME FROM LT_CDHDR.
                ERNAM= LT_CDHDR-USERNAME.
                MANUAL = 'X'.
```

```
                UPDATE MANUAL,LFART,PODAT, POTIM,SUBRC,MESSAGE,ERNAM INTO T_LIST
                STATS=2.
            ENDIF
        ELSE
            //Call direct update rejection and PODAT + POTIM if rejection found
            SELECT * FROM TVPOD WHERE VBELN = T_LIST-VBELN
            IF SY-SUBRC = 0
                T_LIST-SUBRC = 100
                T_LIST-MESSAGE = "POD has been completed manually".
                SELECT CHANGENR FROM CDPOS WHERE OBJECTID = T_LIST-VBELN AND FNAME='PODAT'
                SELECT USERNAME,UDATE,UTIME INTO LT_CDHDR FROM CDHDR WHERE OBJECTID = T_LIST-VBELN AND CHANGENR IN CDPOS-CHANGENR.
                SELECT PODAT,POTIM FROM LIKP WHERE VBELN = T_LIST-VBELN.
                SORT LT_CDHDR DESCENDING BY UDATE,UTIME
                SELECT SINGLE USERNAME FROM LT_CDHDR.
                ERNAM= LT_CDHDR-USERNAME.
                MANUAL = 'X'.
                UPDATE MANUAL,LFART,PODAT, POTIM,SUBRC,MESSAGE,ERNAM INTO T_LIST
                UPDATE MANUAL,LFART,PODAT, POTIM,SUBRC,MESSAGE INTO T_LIST
                STATS=2.
            ELSE
                SELECT VBELN, POSNR, LFIMG INTO LT_LIPS FROM LIPS WHERE VBELN = T_LIST-VBELN AND LFIMG > 0
                CALL BAPI WS_DELIVERY_UPDATE
                    VBKOK_WA-VBELN_VL = T_LIST-VBELN;
                    VBKOK_WA-PODAT= LIKP-WADAT_IST;
                    VBKOK_WA-POTIM= INITIAL;
                    VBKOK_WA-KZPOD= 'B';
                    COMMIT = X.
                    LOOP AT LT_LIPS
                        TVPOD_TAB-VBELN = LT_LIST-VBELN.
                        TVPOD_TAB-POSNR = LT_LIPS-POSNR.
                        TVPOD_TAB-LFIMG_DIFF = LT_LIPS-LFIMG.
                        TVPOD_TAB-GRUND = LT_LIST-VBELN.
                    ENDLOOP
                    IF SY-SUBRC <> 0
                        T_LIST-SUBRC = 200.
                        T_LIST-MESSAGE = BAPI_RET_MESSAGE
                    ELSE
                        T_LIST-MESSAGE = "POD updated successfully"
                        STATS=2.
                    ENDIF
                    UPDATE PODAT, POTIM, SUBRC, MESSAGE INTO T_LIST
            ENDIF
        MOVE T_LIST into ZSD_PODINT
        ZSD_PODINT-TOR_ID= T_LIST-TOR_ID
        ZSD_PODINT-VBELN= T_LIST-VBELN
        ZSD_PODINT-GRUND= T_LIST-GRUND
        ZSD_PODINT-IMGLINK= T_LIST-IMGLINK
        ZSD_PODINT-SUMMARY= 'X'.
        ZSD_PODINT-STATS= STATS.
        ENDIF
    ENDIF
ENDLOOP
//Return message back to caller as this is SYNCHRONOUS interface
RETURN T_LIST
```

ZSDT_PODPARTIAL → need to rename the FM

```
MOVE T_LIST into T_POD
REMOVE DUPLICATE T_POD BY VBELN
DATE= SY-DATUM
TIME=SY-UZEIT
//Inbound at T_POD (loop per Delivery Order)
LOOP at T_POD
    //Call direct Update PODAT + POTIM if still initial
    CLEAR MANUAL. //set MANUAL as blank and change to X when Delivery Processed before this interface
    SELECT LFART,WADAT_IST,PODAT,POTIM FROM LIKP WHERE VBELN =T_POD-VBELN.
    //Error for Delivery Not Exist
    IF SY-SUBRC <> 0
        T_POD-SUBRC = 200
        T_POD-MESSAGE = "Delivery Order not found"
    ELSE
        T_POD-STATS=1.
        T_POD-PODAT = LIKP-WADAT_IST.
        T_POD-LFART = LIKP-LFART.
        SELECT * FROM TVPOD WHERE VBELN = T_POD-VBELN
        IF SY-SUBRC =0
            T_POD-SUBRC = 100
            T_POD-MESSAGE = "POD has been completed manually"
            T_LIST-PODAT = LIKP-PODAT.
```

```
            T_LIST-POTIM = LIKP-POTIM.
            MANUAL = 'X'.
            T_POD-STATS=2.
        ELSE
            //Looping per DO number, and loop for each item for the batch split reject quantity
            SELECT * INTO LT_LIST FROM T_LIST WHERE VBELN = T_POD-VBELN
            LOOP at LT_LIST
                LT_LIPSDET-GRUND = LT_LIST-GRUND.
                IF LT_LIST-LFIMG >0
                    MENGE = LT_LIST-LFIMG.
                    CLEAR LT_LIPS.
                    CLEAR BOM_IND.
                    //Get all delivery quantity based on item and batch split, then put the item no, and rejection quantity , Sequence using the first item to be rejected
                    SELECT * INTO LT_LIPS FROM LIPS WHERE VBELN= LT_LIST-VBELN AND (POSNR = LT_LIST-POSNR OR UECHA = LT_LIST-POSNR)
                    LOOP at LT_LIPS
                        IF LT_LIPS-PSTYV = 'ZTAQ'
                            BOM_IND = 'X'.
                        ENDIF
                        LT_LIPSDET-POSNR = LT_LIPS-POSNR.
                        IF  LT_LIPS-LFIMG >0
                            IF  MENGE> LT_LIPS-LFIMG.
                                LT_LIPSDET-LFIMG_DIFF = LT_LIPS-LFIMG.
                                MENGE = MENGE-LT_LIPS-LFIMG.
                            ELSE
                                LT_LIPSDET-LFIMG_DIFF = LT_LIPS-MENGE.
                                MENGE = 0;
                            ENDIF
                        ENDIF
                        //Exit Loop when Rejection remaining amount = 0
                        IF MENGE = 0
                            EXIT;
                        ENDIF
                    ENDLOOP

                    //start to populate ZTAE item category based on ZTAQ in delivery order item.
                    IF BOM_IND = 'X'.
                        //getting material no and plant to check master data BOM sales and get the master data for component
                        SELECT VBELN, VGBEL, VGPOS, MATNR, WERKS INTO K_LIPS FROM LIPS WHERE VBELN = LT_LIST-VBELN AND POSNR = LT_LIST-POSNR.
                        SELECT STLNR FROM MAST WHERE MATNR = K_LIPS-MATNR AND STLAN='5' AND WERKS = K_LIPS-WERKS
                        SELECT VBELN, POSNR, MATNR INTO ZVBAP FROM VBAP WHERE VBELN = K_LIPS-VGBEL AND UEPOS = K_LIPS-VGPOS AND PSTYV = 'ZTAE'
                        LOOP at ZVBAP
                            //Use the Main POD quantity * master data BOM.
                            SELECT MENGE FROM STPO WHERE STLNR = MAST-STLNR AND MATNR = ZVBAP-MATNR.
                            SELECT VBELN, POSNR, LFIMG FROM LIPS WHERE VBELN = LT_LIST-VBELN AND VGBEL = ZVBAP-VBELN AND VGPOS = ZVBAP-POSNR.
                            INSERT new LT_LIPSDET using LIPS
                            LT_LIPSDET-LFIMG_DIFF = STPO-MENGE * LT_LIST-LFIMG.
                        ENDIF
                    ENDIF


                ENDIF
                //Move to Internal table, later put in ZTABLE
                MOVE LT_LIST to TZSD_PODINT
                    TZSD_PODINT-TOR_ID = LT_LIST-TOR_ID
                    TZSD_PODINT-VBELN = LT_LIST-VBELN
                    TZSD_PODINT-POSNR = LT_LIST-POSNR
                    TZSD_PODINT-MATNR = LT_LIST-MATNR
                    TZSD_PODINT-ARKTX = LT_LIST-ARKTX
                    TZSD_PODINT-GRUND = LT_LIST-GRUND
                    TZSD_PODINT-LFIMG_DIFF = LT_LIST-LFIMG_DIFF
                    TZSD_PODINT-VRKME = LT_LIST-VRKME
                    TZSD_PODINT-PODAT = LIKP-WADAT_IST
                    TZSD_PODINT-POTIM = SY-UZEIT.
                    TZSD_PODINT-IMGLINK = LT_LIST-IMGLINK
                    TZSD_PODINT-STATS = T_POD-STATS.
                    TZSD_PODINT-ERDAT =DATE.
                    TZSD_PODINT-ERZET = TIME.
            //Ending loop per item from DELMAN
            ENDLOOP
            CALL BAPI WS_DELIVERY_UPDATE
                VBKOK_WA-VBELN_VL = LT_POD-VBELN;
                VBKOK_WA-PODAT= LIKP-WADAT_IST;
                VBKOK_WA-POTIM= SY-UZEIT;
                VBKOK_WA-KZPOD= 'B';
                COMMIT = X;
                LOOP at LT_LIPSDET
                    TVPOD_TAB-VBELN = LT_POD-VBELN
```

```
                    TVPOD_TAB-POSNR = LT_LIPSDET-POSNR
                    TVPOD_TAB-LFIMG_DIFF = LT_LIPSDET-LFIMG_DIFF
                    TVPOD_TAB-GRUND = LT_LIPSDET-GRUND
                ENDLOOP
            GET Return Message
            IF SY-SUBRC <> 0
                T_LIST-SUBRC = 200.
                T_LIST-MESSAGE = BAPI_RET_MESSAGE.
            ELSE
                T_LIST-MESSAGE = "POD updated successfully"
                T_POD-STATS=2.
            ENDIF
        ENDIF
    ENDIF
ENDLOOP
// Update Return message based on Delivery No
LOOP at T_POD
UPDATE T_LIST WHERE VBELN = T_POD-VBELN
    UPDATE LFART,PODAT, POTIM, SUBRC, MESSAGE INTO T_LIST
    PODAT=T_POD-PODAT
    LFART= T_POD-LFART
    SUBRC=T_POD-SUBRC
    MESSAGE=T_POD-SUBRC
UPDATE T_ZSD_PODINT-STATS WHERE VBELN = T_POD-VBELN
ENDLOOP
MOVE TZSD_PODINT to ZSD_POD_INT
//Return message back to caller as this is SYNCHRONOUS interface
RETURN T_LIST

```

4.1.1 Data Structure

ZSDT_PODINT [POD Quantity from DELMAN]

FIELD NAME	Data Type	Primary Key	Descriptions	Value Field
TOR_ID	CHAR (20)	Yes	Freight Order No	
VBELN	CHAR (10)	Yes	Delivery No	
POSNR	NUMC (6)	Yes	Delivery Item	
SUMMARY	CHAR (1)		Summary Indicator	
KUNNR	CHAR (10)		Customer ID	
MATNR	CHAR (4)		Material No	
GRUND	CHAR (4)		Rejection Reason	
LFIMG_DIFF	QUAN (17)		Reject Quantity	
VRKME	CHAR		UoM	
PODAT	DATS (10)		POD Date	
POTIM	TIMS (8)		POD Time	
IMGL	STRING		Image Link	
STATS	CHAR (1)		Status	1-Error 2-Success
ERDAT	DATS (10)		Created On	
ERZET	TIMS (8)		Created Time	

ZSDT_PODALL → FM need rename

Input:

Field Name	Data Type
TOR_ID	CHAR (20)
VBELN	CHAR (10)
KUNNR	CHAR (10)
GRUND	CHAR (4)
FILE_PATH_TOKO	STRING

Return:

Field Name	Data Type
TOR_ID	CHAR (20)
LFART	CHAR (4)
VBELN	CHAR (10)
KUNNR	CHAR (10)
GRUND	CHAR (4)
MANUAL	CHAR (1)
ERNAM	CHAR (12)
PODAT	DATUM (8)
POTIM	TIME (8)
SUBRC	CHAR (3)
MESSAGE	STRING

ZSDT_PODPARTIAL → FM need rename

Input:

Field Name	Data Type
TOR_ID	CHAR (20)
VBELN	CHAR (10)
KUNNR	CHAR (10)
PODAT	DATUM (8)
POTIM	TIME (8)
POSNR	NUMC (6)
MATNR	CHAR (40)
LFIMG	QUAN (17)
VRKME	CHAR (3)
GRUND	CHAR (4)
FILE_PATH_TOKO	STRING

Return:

Field Name	Data Type
TOR_ID	CHAR (20)
LFART	CHAR (4)
VBELN	CHAR (10)
KUNNR	CHAR (10)
PODAT	DATUM (8)
POTIM	TIME (8)
POSNR	NUMC (6)
MATNR	CHAR (40)
LFIMG	QUAN (17)
VRKME	CHAR (3)
GRUND	CHAR (4)
MANUAL	CHAR (1)
ERNAM	CHAR (12)
PODAT	DATUM (8)
POTIM	TIME (8)
SUBRC	CHAR (3)
MESSAGE	STRING

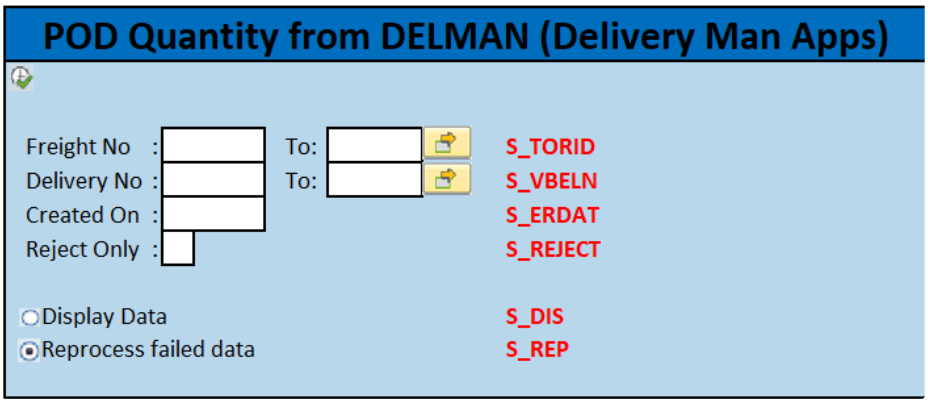
4.1.2 Mapping and Transformation


Interface Field... ing.xls
24 Mar 2025, 01:55 AM

4.2 User Interface

User interface intended for User to view the image which interfaced as link. then SAP will try to access the JPG via Link and insert the necessary Username & Password to access the database.

4.2.2 User Interface-Display Data



Display Data (S_DIS)

POD Quantity from DELMAN (Delivery Man Apps)

No	Freight Order No	Delivery No	Rejection	Image	Status
1	40000123	20000012	Partial Rejection		Unprocessed
2	40000123	20000013	Partial Rejection		Unprocessed
3	40000123	20000014	Full Rejection		Processed
4	40000123	20000015			Processed
5	40000123	20000016			Processed
6	40000123	20000017			Processed

Clicking Image→ Login to AWS and try to display the image by using link and each image separated by ' | '

1. User will see Selection Screen and Input selection criteria
2. Program will select the data from ZSDT_PODINT and check for Authorization object based on freight no
3. Program will display the data after summarize based on Delivery No.
4. Upon Clicking the Image, program will pop-up new screen to view the image. (this requires SAP to login to AWS system to view the image).

Logic for S_DIS

A. Check for Data in ZSDT_PODINT

- If not Found message error: “Tidak ada data dari hasil pencarian.

B. Remove duplicate the data and Check for authority based on EXEC_ORG_ID

C. When authorities not found remove the record

- If no more lines found message error: “Tidak ada otorisasi, silahkan coba hubungi bagian IT.”

D. Show report.

```
SELECT * INTO LT_DETAIL FROM ZSDT_PODINT WHERE TOR_ID IN S_TORID AND VBELN IN S_VBELN AND  ERDAT = S_ERDAT
```

```
IF SY-SUBRC <> 0
    Message Error: "Tidak ada data dari hasil pencarian."
    Exit;
ELSE
    REMOVE DUPLICATE LT_DETAIL BY USING SUMMARY_INDICATOR, TOR_ID, VBELN, GRUND, IMGL, STATS
    LOOP AT LT_DETAIL
        SELECT EXEC_ORG_ID FOR ALL ENTRIES FROM /SCMTMS/D_TORROT WHERE TOR_ID IN S_TORID
        CHECK AUTH OBJECT USING T_TOR_EXE-TM_XEORG against EXEC_ORG_ID AND ACTIVITY = 03
        IF NOT AUTHORIZED
            REMOVE ENTRIES From LT_DETAIL WHEN AUTH OBJECT NOT FOUND
        ELSE
            //Fill Rejection Column
            IF LT_DETAIL-GRUND <> INITIAL
                IF LT_DETAIL-SUMMARY = X
                        LT_DETAIL-GTEXT = "FULL REJECTION"
                   ELSE
                        LT_DETAIL-GTEXT = "PARTIAL REJECTION"
            Move LT_DETAIL to LT_SHOW
        ENDIF
    ENDLOOP
        //if nothing to show, means user doesn't have authority
        IF LT_SHOW <1
            Message Error: "Tidak ada otorisasi, silahkan coba hubungi bagian IT"
        Exit;
        ELSE
            //Remove duplicate and only show header level
            REMOVE DUPLICATE ENTRIES FROM LT_DETAIL BY VBELN
            Show the Report
        ENDIF
ENDIF
```

4.2.3 User Interface-Reprocess Failed Data

1. Get all data with status = 1 (error) and Source = 1 (SKR)
2. Check authorization based on Freight Order Purchasing Organization
3. Perform POD and pop-up message warning, POD has been processed.

Logic for S_REP

```
//Get all data with error status and source from SKR
SELECT * INTO LT_CHECK FROM ZSDT_PODINT WHERE STATUS = '1' AND TOR_ID IN S_TORID AND VBELN IN S_VBELN AND ERDAT = S_ERDAT
IF SY-SUBRC <> 0
    Message Warning: "Tidak ada data untuk di proses ulang."
    Exit.
ELSE
REMOVE Duplicate LT_CHECK by TOR_ID
//Removing entries when user doesn't have authorization
LOOP AT LT_CHECK
    SELECT EXEC_ORG_ID FOR ALL ENTRIES FROM /SCMTMS/D_TORROT WHERE TOR_ID= LT_CHECK-TOR_ID
    CHECK AUTH OBJECT USING T_TOR_EXE-TM_XEORG against EXEC_ORG_ID AND ACTIVITY = 03
    IF NOT AUTHORIZED
        REMOVE ENTRIES From LT_CHECK WHEN AUTH OBJECT NOT FOUND
    ENDIF
ENDLOOP

SELECT * INTO LT_REP FROM ZSDT_PODINT WHERE TOR_ID IN LT_CHECK-TOR_ID
REMOVE DUPLICATE BY VBELN from LT_REP
IF LT_REP > 0
    LOOP at LT_REP
        IF LT_REP-SUMMARY <> INITIAL
                SELECT * FROM TVPOD WHERE VBELN = LT_REP-VBELN
                //Update status to success if TVPOD found or POD status  completed and remove from processing
                IF SY-SUBRC = 0
                    UPDATE ZSDT_PODINT-STATS= '2' WHERE VBELN=LT_REP-VBELN
                    REMOVE Record from LT_REP
                ELSE
                    //POD has been completed without reject by user
                    SELECT PDSTK FROM LIKP WHERE VBELN = LT_REP-VBELN
                    IF PDSTK = C
                        UPDATE ZSDT_PODINT-STATS= '2' WHERE VBELN=LT_REP-VBELN
                        Remove Record FROM LT_REP
                    ELSE //IF POD data not found and not completed
                        SELECT * INTO LT_POD FROM ZSDT_PODINT WHERE VBELN=LT_REP-VBELN
                    LOOP at LT_POD
                        //IF no rejection qty skip this step
                        IF LT_POD-LFIMG_DIFF >0
                            //Get all delivery quantity based on item and batch split, then put the item no, and rejection quantity
, Sequence using the first item to be rejected
                            SELECT * FROM LT_LIPS WHERE VBELN= LT_POD-VBELN AND (POSNR = LT_POD-POSNR OR UECHA = LT_POD-POSNR)
```

```
                            MENGE = LT_POD-LFIMG_DIFF;
                            LOOP at LT_LIPS
                                LT_LIPSDET-POSNR = LT_LIPS-POSNR;
                                IF  MENGE> LT_LIPS-LFIMG
                                    LT_LIPSDET-LFIMG_DIFF = LT_LIPS-LFIMG;
                                    MENGE = MENGE-LT_LIPS-LFIMG;
                                ELSE
                                    LT_LIPSDET-LFIMG_DIFF = LT_LIPS-MENGE;
                                    MENGE = 0;
                                ENDIF
                                //Exit Loop when Rejection remaining amount = 0
                                IF MENGE = 0
                                    EXIT;
                                ENDIF
                            // END of Get all delivery quantity based on item and batch split, then put the item no, and rejection quantity
                            ENDLOOP
                        ENDIF
                    ENDLOOP
                        CALL BAPI WS_DELIVERY_UPDATE
                            VBKOK_WA-VBELN_VL = LT_POD-VBELN;
                            VBKOK_WA-PODAT=LT_POD-PODAT;
                            VBKOK_WA-POTIM= LT_POD-POTIM;
                            VBKOK_WA-KZPOD= 'B';
                            COMMIT = X;
                            LOOP at LT_LIPSDET
                                TVPOD_TAB-VBELN = LT_POD-VBELN
                                TVPOD_TAB-POSNR = LT_LIPSDET-POSNR
                                TVPOD_TAB-LFIMG_DIFF = LT_LIPSDET-LFIMG_DIFF
                                TVPOD_TAB-GRUND = LT_POD-GRUND
                            ENDLOOP
                        //Update Table Status
                        IF ERROR
                        STATS = 1
                        ELSE
                        STATS = 2
                        ENDIF
                        UPDATE ZSDT_PODINT-STATS= STATS WHERE VBELN=LT_REP-VBELN
                ENDIF
            ELSE
                //For rejection handling
                IF LT_REP-GRUND = INITIAL
                    SELECT LFART,WADAT_IST,PODAT,POTIM FROM LIKP WHERE VBELN =T_REP-VBELN
                    IF LIKP-PODAT = INITIAL
                        CALL BAPI WS_DELIVERY_UPDATE
                            VBKOK_WA-VBELN_VL = T_REP-VBELN;
                            VBKOK_WA-PODAT= LIKP-WADAT_IST;
                            VBKOK_WA-POTIM= LT_REP-POTIM;
                            VBKOK_WA-KZPOD= 'B';
                            COMMIT = X.
                        GET Return Message
                        IF SY-SUBRC = 0
                            STATS=2.
                        ENDIF
                            UPDATE ZSDT_PODINT-STATS= STATS WHERE VBELN=LT_REP-VBELN
                    ELSE
                        STATS=2.
                        UPDATE ZSDT_PODINT-STATS= STATS WHERE VBELN=LT_REP-VBELN
                    ENDIF
                ELSE
                    //Call direct update rejection and PODAT + POTIM if rejection found
                    SELECT * FROM TVPOD WHERE VBELN = T_REP-VBELN
                    IF SY-SUBRC = 0
                        STATS=2.
                        UPDATE ZSDT_PODINT-STATS= STATS WHERE VBELN=LT_REP-VBELN
                    ELSE
                        SELECT VBELN, POSNR, LFIMG INTO LT_LIPS FROM LIPS WHERE VBELN = T_REP-VBELN
                        CALL BAPI WS_DELIVERY_UPDATE
                            VBKOK_WA-VBELN_VL = T_LIST-VBELN;
                            VBKOK_WA-PODAT= LIKP-WADAT_IST;
                            VBKOK_WA-POTIM= T_LIST-POTIM;
                            VBKOK_WA-KZPOD= 'B';
                            COMMIT = X.
                            LOOP AT LT_LIPS
                                TVPOD_TAB-VBELN = LT_LIST-VBELN.
                                TVPOD_TAB-POSNR = LT_LIPS-POSNR.
                                TVPOD_TAB-LFIMG_DIFF = LT_LIPS-LFIMG.
                                TVPOD_TAB-GRUND = LT_LIST-VBELN.
                            ENDLOOP
                            IF SY-SUBRC=0
                                STATS=2.
```

```
                          UPDATE ZSDT_PODINT-STATS= STATS WHERE VBELN=LT_REP-VBELN
                      ENDIF
                ENDIF
            ENDIF
        ENDIF
    //end of looping per Delivery No
    ENDLOOP
ENDIF
Message Warning: "Proses ulang selesai dilakukan, silahkan melihat status dari menu display data"
```

5 Security and Controls

5.1 Security Requirements

Upon viewing or processing the report check Freight Org Plan. Exec. Org. vs Authorization object T_TOR_EXE

/SCMTMS/D_TORROT-EXEC_ORG_ID = T_TOR_EXE-TMXEORG with ACTVT = 03

Authorization Object	Authorization Field
T_TOR_EXE	ACTVT = 03
T_TOR_EXE	TM_XEORG

6 Functional Unit Test Scenarios

6.1 Test Scenario / Data

Step #	Test Type	Scenario Title	Steps Performed	Expected Results	Actual Results
1	POD_ALL - FULL SKR	POD_ALL - Full SKR - Auto -Success	• Create Delivery Order • Perform the Interface	• POD data updated • Data updated into ZSDT_PODINT table with status = 2	
2	POD_ALL - FULL SKR	POD_ALL - Full SKR - Auto -Fail	• Create Delivery Order • Open Delivery Order via VLPOD or VL02N • Perform the Interface	• POD data not updated • Data updated into ZSDT_PODINT table with status = 1	
3	POD_ALL - FULL SKR	POD_ALL - Full SKR - Manual POD by user	• Create Delivery Order • Create POD manually via VLPOD • Perform the Interface	• POD data not updated • Data updated into ZSDT_PODINT table with status = 2 (skip process)	
4	POD_ALL - Accept all	POD_ALL - Accept all - Auto - Success	• Create Delivery Order • Perform the Interface	• POD data updated • Data updated into ZSDT_PODINT table with status = 2	
5	POD_ALL - Accept all	POD_ALL - Accept all - Auto - Fail	• Create Delivery Order	• POD data not updated	

			• Open Delivery Order via VLPOD or VL02N • Perform the Interface	• Data updated into ZSDT_PODINT table with status = 1	
6	POD_ALL - Accept all	POD_ALL - Accept all - Manual POD by user	• Create Delivery Order • Create POD manually via VLPOD • Perform the Interface	• POD data not updated • Data updated into ZSDT_PODINT table with status = 2 (skip process)	
7	POD_PARTIAL	POD_PARTIAL - Auto - Success	• Create Delivery Order • Perform the Interface	• POD data updated • Data updated into ZSDT_PODINT table with status = 2	
8	POD_PARTIAL	POD_PARTIAL - Auto - Fail	• Create Delivery Order • Open Delivery Order via VLPOD or VL02N • Perform the Interface	• POD data not updated • Data updated into ZSDT_PODINT table with status = 1	
9	POD_PARTIAL	POD_PARTIAL - Manual POD by user	• Create Delivery Order • Create POD manually via VLPOD • Perform the Interface	• POD data not updated • Data updated into ZSDT_PODINT table with status = 2 (skip process)	

6.2 Error Handling, Validation, Correction and Recovery

No.	Exception Desc.	Mess. Type	Language	Message Text (50 Characters only)
1.	No Authorization	E	E	"Tidak ada otorisasi, silahkan coba hubungi bagian IT"
2.	No Data Found	E	E	“Tidak ada data berdasarkan hasil pencarian”
3.	Reprocessing completed	W	E	"Proses ulang selesai dilakukan, silahkan melihat status dari menu display data"

4.	No data to reprocess	W	E	“Tidak ada data untuk di proses ulang.”

7 Attachments and Documentation

<Attach any additional information in the form of documentation / Appendix / attachments.>