Technical Specification

The Implementation of RFID Enabling Engine (RFP_RFID_EE_001)

RFID
Enabling Engine

1 Executive Summary

Radio-frequency identification (RFID) is the use of an object (typically referred to as an RFID tag) applied to or incorporated into a product, animal, or person for the purpose of identification and tracking using radio waves. In this project, an open-source RFID Enabling Engine (as sponsored by **The Chamber of Hong Kong Logistics Industry**) will be developed to support the upcoming implementation of RFID track & trace system through out the logistic industry.

This document describes the technical specification of developing such RFID Enabling Engine and how to implement it. The major functions of the RFID Enabling Engine will be summarized as follow:

- Print Tag Function
- Activate Tag Function
- Associate Tag Function
- Read Tag Function
- Disassociate Tag Function
- Verify Tag Function

The RFID Enabling Engine will be designed as a J2EE application which can be downloaded from GS1 Hong Kong and is obligated to modified and executed by outsider. The enabling engine will be able to support specification that include EPC (Electronic Product Code) & EPCIS (EPC Information Service) issued by GS1 EPCglobal.

The pilot user can make use of the RFID Enabling Engine to process the functions listed above. The XML data format will be introduced so that the RFID Enabling Engine can communicate with the internal system / application of the pilot user. If any RFID hardware (i.e. RFID Reader / Printer) will be involved, the RFID Enabling Engine also ensures it can connect to those devices with the aid of RFID Middleware.

2 Introduction

Radio-frequency identification (RFID) is the use of an object (typically referred to as an RFID tag) applied to or incorporated into a product, animal, or person for the purpose of identification and tracking using radio waves. In this project, an open-source RFID Enabling Engine (as sponsored by **The Chamber of Hong Kong Logistics Industry**) will be developed to support the upcoming implementation of RFID track & trace system through out the logistic industry.

This document describes the technical specification of developing such RFID Enabling Engine and how to implement it. Overall functional components of the RFID Enabling Engine as well as the framework are illustrated in the following sections, which are organized as follows:

Section 3 provides the references required for understanding the rationale of the technical specification and the standard specifications. The references resources are necessary as the supplements to the technical specification.

Section 4 describes the architecture representation of the document. The technical specification of the RFID Enabling Engine will be explained in details in a View Model. The architectural representation will be addressed from different perspective which corresponding to different concerns.

Section 5 illustrates the macroscopic view of the RFID Enabling Engine on how the application relates & interact with external environment. External entities include RFID Middleware (for RFID hardware communication) & pilot user's application.

Section 6 describes the functions provided by the RFID Enabling Engine which will be explained details in use case model.

Section 7 provides the logical view of the application, which visualizes the design mechanism and architectural structure of the application.

Section 8 illustrates the process view of the application. This section describes the details of various data exchange interface, including RFID Enabling Engine to middleware interface, RFID Enabling Engine to application interface and RFID Enabling Engine to hardware interface.

Section 9 provides the implementation view of the application. Class diagram, collaboration diagrams will be introduced to explain specification details.

Section 10 provides the deployment view. This includes how the environmental setup and the deployment procedures related to the enabling engine.

Section 11 will provide the database schema which gives the technical design details in data perspective. (Database specification will be mentioned in Appendix A)

Section 12 will describes additional properties of the RFID Enabling Engine. The conceptual model of UI design will also be mentioned in Appendix B.

3 References

Supplement specifications & documents are listed below for user reference purpose. By referring to the documentation listed, the user can acquire a general idea of why & how the RFID Enabling Engine be developed in the specification that mentioned in the View Model.

Document	Version	Date	Organization
The EPCglobal Architecture Framework	1.3	19 Mar 2009	EPCglobal Inc.
EPC Information Services (EPCIS) Version 1.0.1 Specification	1.0.1	21 Sept 2007	EPCglobal Inc.
GS1 GTIN Check Digit Calculator Algorithm (Web)	-	-	GS1
RFID Enabling Engine Installation Guide	1.0		GS1 HK
RFID Enabling Engine User Manual	1.0		GS1 HK

4 Architecture Representation

The architecture representation will make use of the View Model, to illustrate the technical specification in different perspectives, which corresponds to different concerns:

Requirement View:	describes the functional & non-functional requirements of the RFID Enabling Engine, as illustrated by use case diagram
Logical View:	describes the object oriented model of the application design, the decomposition of the engine into detailed package
Process View:	describes the interface interaction between different entities
Implementation View:	describes the RFID Enabling Engine functions and libraries in the form of class library
Deployment View:	describes the RFID Enabling Engine mapping onto hardware
Data View:	describes the database specification of the RFID Enabling Engine

5 Relation to External Environment

External Entities likes RFID Middleware, RFID Hardware API & Pilot User Application can interact with the RFID Enabling Engine to demonstrate an actual real case RFID environment:

- RFID Middleware – following the EPCglobal standards (ALE specification v1.0), the middleware act as a middle layer to communicate & filter RFID data in-between RFID Enabling Engine & RFID equipments (e.g. readers)
- RFID Hardware API – the RFID Enabling Engine is capable to control RFID equipments (e.g. printer) by requesting service from the hardware API
- Pilot User Application – the RFID Enabling Engine will provide XML file for other application to capture (like EPCIS event) or receive XML file from RFID Middleware (like receiving converted EPC data).

Specific supported track and trace event will be listed as follow:

- Verify Event (Sample)

  The Verify Event specifies OBSERVE action and Shipping bizstep as illustrated below:

```
- <xsd:ReadDocument xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:xsd="xsd.read.sedna.com">
- <xsd:ReadBody>
- <xsd:ReadList>
- <xsd:Read>
  <xsd:rfid>000000000000000002221006</xsd:rfid>
  <xsd:action>OBSERVE</xsd:action>
  <xsd:bizStep>urn:epcglobal:fmcg:bizstep:shipping</xsd:bizStep>
  <xsd:disposition>urn:epcglobal:fmcg:disp:active</xsd:disposition>
- <xsd:bizLocation>
  <xsd:id>urn:epcglobal:fmcg:loc:tt</xsd:id>
    </xsd:bizLocation>
  <xsd:readpoint>22</xsd:readpoint>
  <xsd:timestamp>2009-08-07T17:31:10.000+08:00</xsd:timestamp>
    </xsd:Read>
- <xsd:Read>
  <xsd:rfid>000000000000000002221002</xsd:rfid>
  <xsd:action>OBSERVE</xsd:action>
  <xsd:bizStep>urn:epcglobal:fmcg:bizstep:shipping</xsd:bizStep>
  <xsd:disposition>urn:epcglobal:fmcg:disp:active</xsd:disposition>
- <xsd:bizLocation>
  <xsd:id>urn:epcglobal:fmcg:loc:tt</xsd:id>
    </xsd:bizLocation>
  <xsd:readpoint>22</xsd:readpoint>
  <xsd:timestamp>2009-08-07T17:31:10.000+08:00</xsd:timestamp>
    </xsd:Read>
- <xsd:Read>
  <xsd:rfid>000000000000000002221015</xsd:rfid>
  <xsd:action>OBSERVE</xsd:action>
  <xsd:bizStep>urn:epcglobal:fmcg:bizstep:shipping</xsd:bizStep>
  <xsd:disposition>urn:epcglobal:fmcg:disp:active</xsd:disposition>
- <xsd:bizLocation>
  <xsd:id>urn:epcglobal:fmcg:loc:tt</xsd:id>
    </xsd:bizLocation>
  <xsd:readpoint>22</xsd:readpoint>
  <xsd:timestamp>2009-08-07T17:31:10.000+08:00</xsd:timestamp>
    </xsd:Read>
    </xsd:ReadList>
    </xsd:ReadBody>
    </xsd:ReadDocument>
```

- Association Event (Sample)
  The Association Event specifies ADD action and Packing bizStep as illustrated below:

```
- <xsd:ReadDocument xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:xsd="xsd.read.sedna.com">
- <xsd:ReadBody>
- <xsd:ReadList>
- <xsd:Read>
  <xsd:parent>000000000000000002221015</xsd:parent>
  <xsd:rfid>000000000000000002221016</xsd:rfid>
  <xsd:action>ADD</xsd:action>
  <xsd:bizStep>urn:epcglobal:fmcg:bizstep:packing</xsd:bizStep>
  <xsd:disposition>urn:epcglobal:fmcg:disp:active</xsd:disposition>
- <xsd:bizLocation>
```

```
  <xsd:id>urn:epcglobal:fmcg:loc:tt</xsd:id>
    </xsd:bizLocation>
  <xsd:readpoint>22</xsd:readpoint>
  <xsd:timestamp>2009-08-07T17:25:48.000+08:00</xsd:timestamp>
    </xsd:Read>
- <xsd:Read>
  <xsd:parent>000000000000000002221015</xsd:parent>
  <xsd:rfid>000000000000000002221014</xsd:rfid>
  <xsd:action>ADD</xsd:action>
  <xsd:bizStep>urn:epcglobal:fmcg:bizstep:packing</xsd:bizStep>
  <xsd:disposition>urn:epcglobal:fmcg:disp:active</xsd:disposition>
- <xsd:bizLocation>
  <xsd:id>urn:epcglobal:fmcg:loc:tt</xsd:id>
    </xsd:bizLocation>
  <xsd:readpoint>22</xsd:readpoint>
  <xsd:timestamp>2009-08-07T17:25:48.000+08:00</xsd:timestamp>
    </xsd:Read>
- <xsd:Read>
  <xsd:parent>000000000000000002221015</xsd:parent>
  <xsd:rfid>000000000000000002221007</xsd:rfid>
  <xsd:action>ADD</xsd:action>
  <xsd:bizStep>urn:epcglobal:fmcg:bizstep:packing</xsd:bizStep>
  <xsd:disposition>urn:epcglobal:fmcg:disp:active</xsd:disposition>
- <xsd:bizLocation>
  <xsd:id>urn:epcglobal:fmcg:loc:tt</xsd:id>
    </xsd:bizLocation>
  <xsd:readpoint>22</xsd:readpoint>
  <xsd:timestamp>2009-08-07T17:25:48.000+08:00</xsd:timestamp>
    </xsd:Read>
    </xsd:ReadList>
    </xsd:ReadBody>
    </xsd:ReadDocument>
```

- Disassociation Event (Sample)

  The Disassociation Event specifies DELETE action and Disaggregate bizStep as illustrated below:

```
- <xsd:ReadDocument xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:xsd="xsd.read.sedna.com">
- <xsd:ReadBody>
- <xsd:ReadList>
- <xsd:Read>
  <xsd:parent>000000000000000002221016</xsd:parent>
  <xsd:rfid>000000000000000002221006</xsd:rfid>
  <xsd:action>DELETE</xsd:action>
  <xsd:bizStep>urn:epcglobal:fmcg:bizstep:disaggregate</xsd:bizStep>
  <xsd:disposition>urn:epcglobal:fmcg:disp:active</xsd:disposition>
- <xsd:bizLocation>
  <xsd:id>urn:epcglobal:fmcg:loc:tt</xsd:id>
     </xsd:bizLocation>
  <xsd:readpoint>22</xsd:readpoint>
  <xsd:timestamp>2009-08-07T17:30:13.000+08:00</xsd:timestamp>
     </xsd:Read>
- <xsd:Read>
  <xsd:parent>000000000000000002221016</xsd:parent>
  <xsd:rfid>000000000000000002221002</xsd:rfid>
  <xsd:action>DELETE</xsd:action>
  <xsd:bizStep>urn:epcglobal:fmcg:bizstep:disaggregate</xsd:bizStep>
  <xsd:disposition>urn:epcglobal:fmcg:disp:active</xsd:disposition>
- <xsd:bizLocation>
  <xsd:id>urn:epcglobal:fmcg:loc:tt</xsd:id>
     </xsd:bizLocation>
  <xsd:readpoint>22</xsd:readpoint>
  <xsd:timestamp>2009-08-07T17:30:13.000+08:00</xsd:timestamp>
     </xsd:Read>
- <xsd:Read>
  <xsd:parent>000000000000000002221016</xsd:parent>
  <xsd:rfid>000000000000000002221015</xsd:rfid>
  <xsd:action>DELETE</xsd:action>
  <xsd:bizStep>urn:epcglobal:fmcg:bizstep:disaggregate</xsd:bizStep>
  <xsd:disposition>urn:epcglobal:fmcg:disp:active</xsd:disposition>
- <xsd:bizLocation>
  <xsd:id>urn:epcglobal:fmcg:loc:tt</xsd:id>
    </xsd:bizLocation>
  <xsd:readpoint>22</xsd:readpoint>
  <xsd:timestamp>2009-08-07T17:30:13.000+08:00</xsd:timestamp>
    </xsd:Read>
    </xsd:ReadList>
    </xsd:ReadBody>
    </xsd:ReadDocument>
```

- ➢ Read Event (Sample)
  For the event with the correct XML format but has not been specified as Verify, Association, Disaaociation Event, the XML received will be regarded as Read Event as illustrated below:

```
- <xsd:ReadDocument xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:xsd="xsd.read.sedna.com">
- <xsd:ReadBody>
- <xsd:ReadList>
- <xsd:Read>
  <xsd:rfid>000000000000000002221006</xsd:rfid>
  <xsd:action>OBSERVE</xsd:action>
  <xsd:bizStep>urn:epcglobal:fmcg:bizstep:picking</xsd:bizStep>
  <xsd:disposition>urn:epcglobal:fmcg:disp:in_progress</xsd:disposition>
- <xsd:bizLocation>
  <xsd:id>urn:epcglobal:fmcg:loc:tt</xsd:id>
    </xsd:bizLocation>
  <xsd:readpoint>22</xsd:readpoint>
  <xsd:timestamp>2009-08-07T17:31:10.000+08:00</xsd:timestamp>
    </xsd:Read>
     </xsd:ReadList>
    </xsd:ReadBody>
    </xsd:ReadDocument>
```

- Receiving Converted EPC Data Event(Sample)

  The Receiving Converted EPC Data Event specifies EPC data received from RFID Middleware as illustrated below:

```
- <Order xmlns="com.sedna.web.epcconvert">
- <EPCListResult>
- <Item>
  <barcode>2526451354223219884032251</barcode>
  <epc>urn:epc:id:giai:252645135.4223219884032251</epc>
  <tag>urn:epc:tag:giai-96:0.252645135.4223219884032251</tag>
  <rfid>340CF0F0F0FF00FF00FF00FB</rfid>
    </Item>
- <Item>
  <barcode>2526451354223219884032257</barcode>
  <epc>urn:epc:id:giai:252645135.4223219884032257</epc>
  <tag>urn:epc:tag:giai-96:0.252645135.4223219884032257</tag>
  <rfid>340CF0F0F0FF00FF00FF0101</rfid>
    </Item>
    </EPCListResult>
    </Order>
```

6 Requirement View

The RFID Enabling Engine implements the EPCglobal standard and will be developed under the guidance of EPCglobal Architecture Framework. The engine will support EPC schema as illustrated before in order to complete data exchange with EPC network (e.g. ezTRACK) & RFID Middleware.

6.1 System Functions

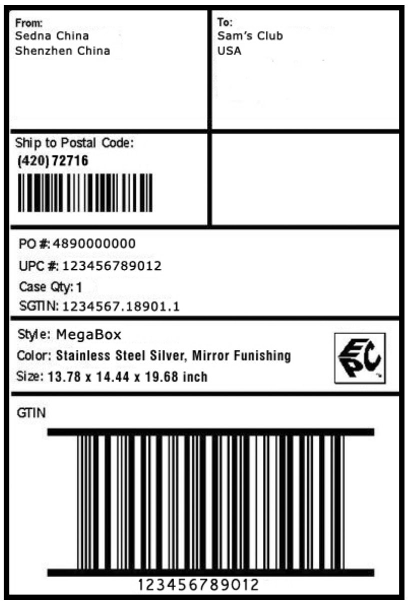
The RFID Enabling Engine will consist of the following functions:

6.1.1 Print Tag Function

The print tag function of the RFID Enabling Engine support printing of:

- ➢ Item-level RFID EPC Tag (SGTIN)
  (by user selection of EPC Filter Value)
- ➢ Carton-level RFID EPC Tag (SGTIN)
  (by user selection of EPC Filter Value)
- ➢ Pallet-level or Container-level RFID EPC Tag (GIAI / GRAI)

According to the announced standard of Wal*Mart, a carton-level label template will be included in the print tag function.



From:
Sedna China
Shenzhen China

To:
Sam's Club
USA

Ship to Postal Code:
(420) 72716

PO #: 4890000000
UPC #: 123456789012
Case Qty: 1
SGTIN: 1234567.18901.1

Style: MegaBox
Color: Stainless Steel Silver, Mirror Funishing
Size: 13.78 x 14.44 x 19.68 inch

GTIN
123456789012

6.1.2 Activate Tag Function

The RFID Enabling Engine support auto-submission / manual-submission of EPCIS event for tag activation.

- ➢ Auto-submission

If the auto-submission has been turned on, every single tag will be activated automatically during printing. EPCIS event will be submitted to EPCIS Service automatically.

- Manual-submission
  If the auto-submission has been turned off, tags will require manual activation. EPCIS event will be submitted to EPCIS Service after user activates manually (Activate All).

6.1.3 Associate Tag Function

Three levels of data association will be supported by RFID Enabling Engine:

- Item-level tag to carton-level tag
- Carton-level tag to pallet-level tag
- Pallet-level tag to container-level tag

EPCIS event according to each association will be submitted to EPCIS Service after confirmation by user.

6.1.4 Read Tag Function

The RFID Enabling Engine supports tag reading (i.e. captured RFID EPC data from RFID Middleware / RFID equipment) and capable to register the EPCIS event respective to its location information to EPCIS Service.

6.1.5 Disassociate Tag Function

Three levels of data disassociation will be supported by RFID Enabling Engine:

- Item-level tag & carton-level tag
- Carton-level tag & pallet-level tag
- Pallet-level tag & container-level tag

EPCIS event according to each disassociation will be submitted to EPCIS Service after confirmation by user.

6.1.6 Verify Tag Function

The RFID Enabling Engine support tag verification of tag data against pre-defined value.

6.1.7 Handheld Reader Support Function

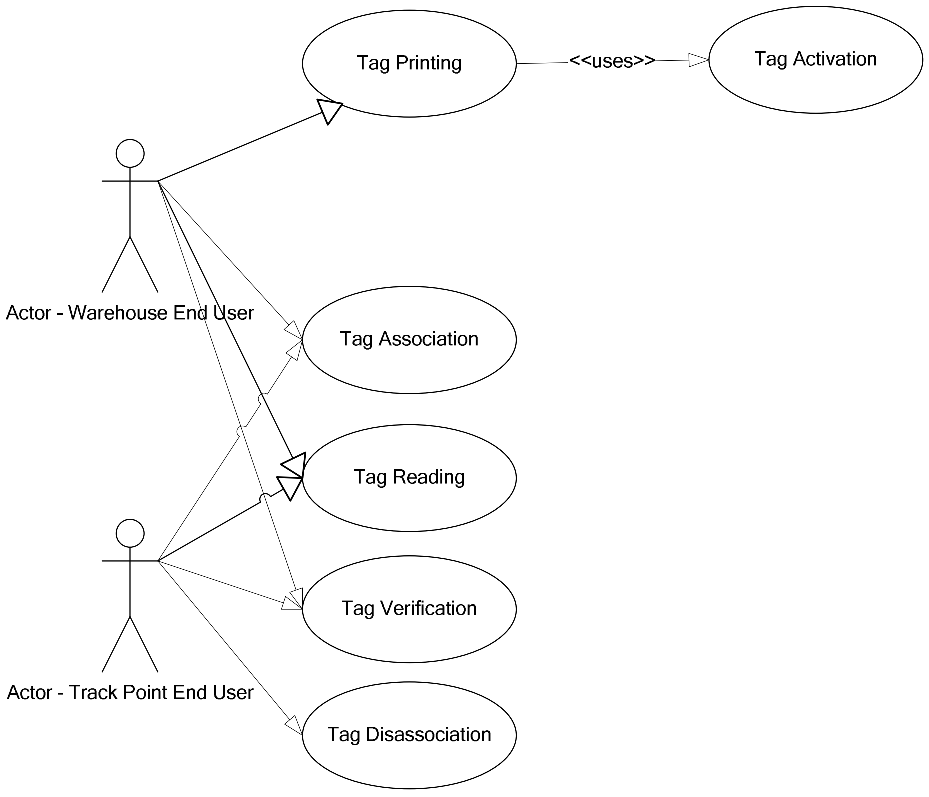
Data exchange interface in XML data format will be supported by RFID Enabling Engine to receive RFID data captured by RFID handheld devices. (For XML sample, please refer to section 5)

6.1.8 RFID Middleware Support Function

Data exchange interface in XML data format will be supported by RFID Enabling Engine to deliver data to RFID Middleware for EPC data conversion. Converted EPC data will then be received by RFID Enabling Engine for upcoming process. (e.g. Printing)

6.2 Use Cases

The following Use Case diagram illustrates the major functions supported by the RFID Enabling Engine.



Actor	Definition
Warehouse End User	Initial end user that interact with the RFID Enable Engine
Track Point End User	End user of each supply chain track point that interact

	with the RFID Enable Engine

6.2.1 Tag Printing

6.2.1.1 Description

The use case describes the tag printing function & procedure

6.2.1.2 Actors

- Warehouse End User of the first tagging track point

6.2.1.3 Preconditions

- None

6.2.1.4 Major Flow of Event

- Manual data input of product information (UPC / product relevant information)
- Connect to RFID printer and trigger tag printing

6.2.1.5 Alternate Flow

- None

6.2.1.6 Post-conditions

- A transaction of PO has been generated in the system

6.2.2 Tag Activation

6.2.2.1 Description

The use case describes the tag activation function & procedure

6.2.2.2 Actors

- Warehouse End User of the first tagging track point

6.2.2.3 Preconditions

- Tag printing complete

6.2.2.4 Major Flow of Event

- Submit EPCIS event data of tag activation
- Return receive signal from EPCIS Service

6.2.2.5 Alternate Flow

- Auto-trigger resubmission continuously if data failed to transmit
- Default resubmission timespan: 60 seconds

6.2.2.6 Post-conditions

- Tag activated
- Data submitted to EPCIS Service

6.2.3 Tag Association

6.2.3.1 Description

The use case describes the tag association function & procedure

6.2.3.2 Actors

- Warehouse End User of the first tagging track point
- Other Track Point End User

6.2.3.3 Preconditions

- None

6.2.3.4 Major Flow of Event

- Associate different levels tag data by RFID handheld reader (reading)
- Transfer associated data from RFID handheld reader to RFID Enabling Engine
- Submit EPCIS event data of tag association

6.2.3.5 Alternate Flow

- Exception is thrown when duplicate parent tag is read
- Exception is thrown when EPC not registered in database
- Auto-trigger resubmission continuously if data failed to transmit
- Default resubmission timespan: 60 seconds

6.2.3.6 Post-conditions

- Data submitted to EPCIS Service

6.2.4 Tag Verification

6.2.4.1 Description

The use case describes the tag verification function & procedure

6.2.4.2 Actors

- Warehouse End User of the first tagging track point
- Other Track Point End User

6.2.4.3 Preconditions

- None

6.2.4.4 Major Flow of Event

- Receive RFID tag data from handheld to enabling engine
- Verify RFID EPC data integrity
- Submit EPCIS event data of tag verification

6.2.4.5 Alternate Flow

- Exception is thrown when duplicate tag is read
- Exception is thrown when EPC not registered in database
- Auto-trigger resubmission continuously if data failed to transmit
- Default resubmission timespan: 60 seconds

6.2.4.6 Post-conditions

- Data submitted to EPCIS Service

6.2.5 Tag Reading

6.2.5.1 Description

The use case describes the tag reading function & procedure

6.2.5.2 Actors

- Warehouse End User of the first tagging track point
- Other Track Point End User

6.2.5.3 Preconditions

- None

6.2.5.4 Major Flow of Event

- Physical tag transferred from one location to another
- Read tag data (transferred from RFID Handheld Reader or RFID Middleware)
- Register the tag data with location information in EPCIS

6.2.5.5 Alternate Flow

- Exception is thrown when duplicate tag is read
- Exception is thrown when EPC not registered in database
- Auto-trigger resubmission continuously if data failed to transmit
- Default resubmission timespan: 60 seconds

6.2.5.6 Post-conditions

- Data submitted to EPCIS Service

6.2.6 Tag Disassociation

6.2.6.1 Description

The use case describes the tag disassociation function & procedure

6.2.6.2 Actors

- Warehouse End User of the first tagging track point

- Other Track Point End User

6.2.6.3 Preconditions

- Tag association complete

6.2.6.4 Major Flow of Event

- Disassociate different levels tag data
- Submit EPCIS event data of tag disassociation

6.2.6.5 Alternate Flow

- Exception is thrown when duplicate tag is read
- Exception is thrown when EPC not registered in database
- Auto-trigger resubmission continuously if data failed to transmit
- Default resubmission timespan: 60 seconds

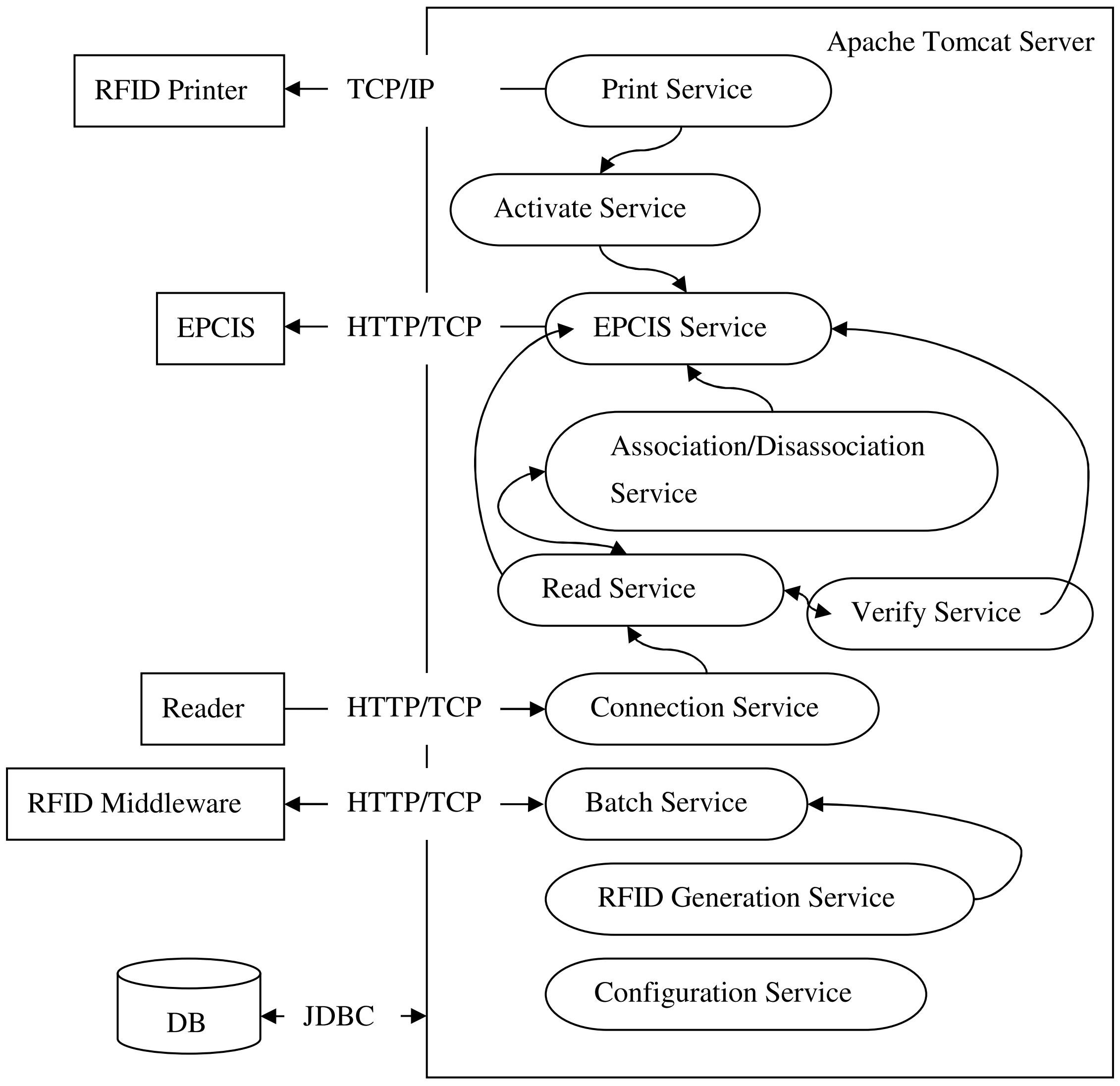
6.2.6.6 Post-conditions

- Data submitted to EPCIS Service

7 Logical View

7.1 Architectural Design

The RFID Enabling Engine is a J2EE application hosted in Apache Tomcat server. It connects databases via JDBC. The RFID Enabling Engine interfaces with outside components via HTTP/TCP and TCP/IP.



7.2 Design Mechanism

The core business logics of the RFID Enabling Engine are divided into components.

7.2.1 Print Service

This component handles printing. It consists of the following methods.

- Class Name
  PrintDispatchAction
- Package
  org.rfidee.web.action
- Method List

Return Type	Declaration
ActionForward	printEpcPrint(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
void	printing(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printOrder(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)

- Class Name
  PrintDispatchActionImpl
- Package
  org.rfidee.web.action
- Method List

Return Type	Declaration
byte[]	convertBytes(String printer, byte[] bytes)
String	convertChars(String printer, String string)
String	convertSGTINBarcode(String printer, String string)
String	convertSSCCBarcode(String printer, String string)
String	convertZipcode(String printer, String string)
String	generatePrintingData(String string, PrintOrder printOrder, RfidItem rfidItem, String printer)
List	printEpcPrintImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)

- Class Name
  File2String
- Package
  org.rfidee.common.util
- Method List

Return Type	Declaration
String	readFileAsString(String filePath)

7.2.2 Activate Service

This component handles activating the RFID tag. It consists of the following methods.

- Class Name
  PrintDispatchAction
- Package
  org.rfidee.web.action
- Method List

Return Type	Declaration
void	activateTag(List list)
TagHistory	saveActivateHistory(RfidItem rfidItem)
void	packActivateEvent(List<TagHistory> tagHistoryList)
void	packActivateEventImpl(String epc, int count, PrintOrder printOrder, RfidItem rfidItem, List<TagHistory> packTagHistoryList)
void	packAll(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
void	packAllActivateEvents()

7.2.3 EPCIS Service

This component handles registering the event in EPCIS. It consists of the following methods.

- Class Name
  schedulingUpload2GS1
- Package
  org.rfidee.common.quartz
- Method List

Return Type	Declaration
void	run()

- Class Name
  processState
- Package
  org.rfidee.common.quartz
- Method List

Return Type	Declaration
processState	getInstance()
void	doProcess()

- Class Name
  HTTPPostSender
- Package
  org.rfidee.common.util
- Method List

Return Type	Declaration
URLConnection	proceed(String XML)

- Class Name
  SSLUtilities
- Package
  org.rfidee.common.util
- Method List

Return Type	Declaration
void	trustAllHostnames()
void	trustAllHttpsCertificates()

7.2.4 Connection Service

This component handles the connection with the reader. It consists of the following methods.

- Class Name
  CallbackServerDispatchAction
- Package
  org.rfidee.web.action
- Method List

Return Type	Declaration
ActionForward	SendEPC(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)

- Class Name
  ALEHandler
- Package
  org.rfidee.module.alehandler
- Method List

Return Type	Declaration
int	process()
boolean	validate()

- Class Name
  InputHandler
- Package
  org.rfidee.module.alehandler
- Method List

Return Type	Declaration
int	process()
boolean	validate()

7.2.5 Read Service

This component handles and classifies the RFID tag from reader. It consists of the following methods.

- Class Name
  ReadHandler
- Package
  org.rfidee.module.alehandler
- Method List

Return Type	Declaration
void	clear()
void	clearAssoDissoChildItems(String rfid, String action, String bizStep)
void	clearAssoDissoItems()
void	clearAssoDissoParentItems(String rfid, String action, String bizStep)
RfidItem	getValidDBObject(RfidItem object)
Int	handle(ECReportsDocument doc)
int	handle(ReadDocumentDocument1 doc)
boolean	isDisable()
boolean	isExists(String parentrfid, String childrfid, String bizStep, String action)
String	removeCharAt(String s, int pos)
void	setDisable(boolean disable)

- Class Name
  ReadDispatchAction
- Package
  org.rfidee.web.action
- Method List

Return Type	Declaration
List<ReadItem>	getReadItemList(ReadItem item)
void	packAssoEvent(List<TagHistory> tagHistoryList)
void	packAssoEventImpl(String epc, int count, PrintOrder printOrder, RfidItem rfidItem, List<TagHistory> packTagHistoryList)
Void	packReadEvent(List<TagHistory> tagHistoryList)
void	packVerifyEvent(List<TagHistory> tagHistoryList)
void	packVerifyEventImpl(String epc, int count, PrintOrder printOrder, RfidItem rfidItem, List<TagHistory> packTagHistoryList)
TagHistory	saveAssoHistory(ReadItem readItem, String parent)

TagHistory	saveVerifyHistory(ReadItem readItem)
ActionForward	clearReadItem(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	clearReadItems(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	readConfirmAll(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	readLayout(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	readRefresh(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	showAllRead(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)

7.2.6 Verify Service

This component handles verifying the RFID tag from reader. It consists of the following methods.

- Class Name
  VerifyHandler
- Package
  org.rfidee.module.alehandler
- Method List

Return Type	Declaration
void	clear()
boolean	isDisable()
boolean	isExists(String childrfid)
boolean	isVerifyEvent(String action, String bizStep)
void	setDisable(boolean disable)

- Class Name
  VerificationDispatchAction
- Package
  org.rfidee.web.action
- Method List

Return Type	Declaration
void	packVerifyEvent(List<TagHistory> tagHistoryList)
void	packVerifyEventImpl(String parent, String epc, int count, PrintOrder printOrder, RfidItem rfidItem, List<TagHistory> packTagHistoryList)
TagHistory	saveVerifyHistory(ReadItem readItem)
ActionForward	clearItems(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	confirmAll(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	removeItem(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	showAll(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	verificationOutput(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	verificationRefreshPage(ActionMapping mapping, ActionForm form,

	HttpServletRequest request, HttpServletResponse response)

7.2.7 Association/Disassociation Service

This component handles associating or disassociating the RFID tag from reader. It consists of the following methods.

- Class Name
  AssoDisassoHandler
- Package
  org.rfidee.module.alehandler
- Method List

Return Type	Declaration
int	AssociationCount()
boolean	checkRelationshipByRfid(String parentrfid, String childrfid)
void	clear()
int	DisAssociationCount()
boolean	isADissoAllExist(String parentrfid)
boolean	isAlreadyAsso(String rfid, String parent)
boolean	isAssoEvent(String action, String bizStep)
boolean	isDisable()
boolean	isDissoEvent(String action, String bizStep)
boolean	isExists(String parentrfid, String childrfid, String bizStep, String action)
boolean	isInvalidExists(String parentrfid, String childrfid, String bizStep, String action)
void	setDisable(boolean disable)

- Class Name
  AssociationDispatchAction
- Package
  org.rfidee.web.action
- Method List

Return Type	Declaration
List<ReadItem>	getReadItemList(ReadItem item)
void	packAssoEvent(List<TagHistory> tagHistoryList)
void	packAssoEventImpl(String epc, int count, PrintOrder printOrder, RfidItem rfidItem, List<TagHistory> packTagHistoryList)
TagHistory	saveAssoHistory(ReadItem readItem, String parent)
ActionForward	assoDisassoRefresh(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)

ActionForward	clearAssoDissoItems(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	clearAssoDissoParentItems(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	comfirmAll(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	comfirmByRfid(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
int	parentSizecount(String rfid, String action, String bizStep)
ActionForward	showAllAssociation(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	showAllparent(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	showAssoDisassoLayout(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)

7.2.8 Batch Service

This component handles creating, editing and searching a batch. It consists of the following methods.

- Class Name
  PrintDispatchAction
- Package
  org.rfidee.web.action
- Method List

Return Type	Declaration
ActionForward	pcBack(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	pcDeletePattern(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	pcDetails(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	PCInput(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	pcNext(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	pcSample(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	pcSave(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printBack(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printDeletePattern(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printDetails(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printEpcList(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printEpcSearch(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printGiaiAdd(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printGraiAdd(ActionMapping mapping, ActionForm form,

	HttpServletRequest request, HttpServletResponse response)
ActionForward	printInput(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printNext(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printPoDelete(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printPoList(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printPoSearch(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printSample(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printSave(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	printSgtinAdd(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
ActionForward	TagDetails(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)

- Class Name
  PrintDispatchActionImpl
- Package
  org.rfidee.web.action
- Method List

Return Type	Declaration
PrintDispatchActionImpl	getInstance()
void	clearTemplateGiaiImpl(ActionForm form)
void	clearTemplateGraiImpl(ActionForm form)
void	clearTemplateSgtinImpl(ActionForm form)
List	pcPatternListImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response, boolean flag)
void	pcSampleImpl(ActionForm form)
PrintOrder	pcSaveImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)

PrintOrder	printDetailsImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
RfidItem	printEpcSearchImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
void	printGiaiCleanImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
Map	printGiaiVarInputImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
Pattern	printGiaiVarOutputImpl(Map<String, String> map)
void	printGraiCleanImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
Map	printGraiVarInputImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
Pattern	printGraiVarOutputImpl(Map<String, String> map)
void	printNextImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
List	printOrderImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
List	printPatternListImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response, boolean flag)
PrintOrder	printPoDeleteImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
PrintOrder	printPoSearchImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
void	printSampleImpl(ActionForm form)
PrintOrder	printSaveImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
void	printSgtinCleanImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
Map	printSgtinVarInputImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
Pattern	printSgtinVarOutputImpl(Map<String, String> map)
void	TagDetailsImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response, RfidItem rfidItem)

- Class Name
  CheckGiai
- Package
  org.rfidee.common.util
- Method List

Return Type	Declaration
List	check()
ActionErrors	getErrorMessages()

- Class Name
  CheckGrai
- Package
  org.rfidee.common.util
- Method List

Return Type	Declaration
List	check()
ActionErrors	getErrorMessages()

- Class Name
  CheckSgtin
- Package
  org.rfidee.common.util
- Method List

Return Type	Declaration
List	check()
ActionErrors	getErrorMessages()

- Class Name
  InsertZero
- Package
  org.rfidee.common.util
- Method List

Return Type	Declaration
String	insert(String s, int len)

7.2.9 RFID Generation Service

This component handles generating the RFID tag serial number. It consists of the following methods.

- Class Name
  PrintDispatchActionImpl
- Package
  org.rfidee.web.action
- Method List

Return Type	Declaration
String	getPrefix(String str)
Boolean	GenerateRfidImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response, PrintOrder printOrder)
Boolean	GenerateRfidWithOutDataplexImpl(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response, PrintOrder printOrder)
int	GraiGetCheckDigit(String PreGRAI)
String	HexDecConversion(String Hex, int Dec, boolean hextodec)
String	increment(String str, int length, int value)
String	incrementHex(String str, int length, int value)
void	saveRfidItem(String epc, PrintOrder printOrder, Pattern pattern)
String	sendtoDataplex(String url,String content)
int	SgtinGetCheckDigit(String GTIN13)

- Class Name
  HTTPPostSender
- Package
  org.rfidee.common.util
- Method List

Return Type	Declaration
URLConnection	proceed(String XML)

- Class Name
  SSLUtilities
- Package
  org.rfidee.common.util

➢ Method List

Return Type	Declaration
void	trustAllHostnames()
void	trustAllHttpsCertificates()

7.2.10 Configuration Service

This component handles the system configuration. It consists of the following methods.

- Class Name
  AdministrationDispatchAction
- Package
  org.rfidee.web.action
- Method List

Return Type	Declaration
ActionForward	adminConfig(ActionMapping mapping
ActionForward	adminConfigError(ActionMapping mapping
ActionForward	adminSave(ActionMapping mapping

8 Process View

8.1 RFID Enabling Engine to RFID Equipment Interface

The RFID Enabling Engine support RFID tag printing in the form of calling RFID Printer API. The RFID Enabling Engine connects the physical RFID Printer via TCP/IP connection.

For RFID tag reading, the RFID Enabling Engine support the standard XML file format so that it can communicate with RFID Reader via HTTP connection. The CallbackServerDispatchAction (*java interface class*) is responsible for collecting the RFID data and the java classes in the package, org.rfidee.module.alehandler, are responsible for processing.

8.2 RFID Enabling Engine to EPCIS Interface

The RFID Enabling Engine pre-defines several EPCIS events which are activation, verification, association and disassociation in the system. The engine will process the necessary event data into the EPCIS standard event data. A scheduler class, schedulingUpload2GS1, in the engine will then post the data to EPCIS via HTTP connection, keep track of the status for submission and process auto-resubmission in case of any failure submission. (Sample XML can refer to section 5)

8.3 RFID Enabling Engine to RFID Middleware Interface

The RFID Enabling Engine pre-defines XML data format for data exchange with RFID Middleware. After the creation of serial number in SGTIN, GIAI or GRAI, the data can be delivered to RFID Middleware for EPC data conversion. Converted EPC data will then be sent back to RFID Enabling Engine for upcoming process. (e.g. printing)

Sample XML for delivering data to RFID Middleware:

```
- <Order xmlns="com.sedna.web.epcconvert">
- <EPCList>
- <GiaiItem>
   <company_prefix>252645135</company_prefix>
   <serial_reference>4223219884032251</serial_reference>
   <serial_end_reference>4223219884032257</serial_end_reference>
   <epc_filter>0</epc_filter>
```

```
    </GiaiItem>
    </EPCList>
    </Order>
```

Sample XML for receiving data from RFID Middleware:

```
- <Order xmlns="com.sedna.web.epcconvert">
- <EPCListResult>
- <Item>
  <barcode>2526451354223219884032251</barcode>
  <epc>urn:epc:id:giai:252645135.4223219884032251</epc>
  <tag>urn:epc:tag:giai-96:0.252645135.4223219884032251</tag>
  <rfid>340CF0F0F0FF00FF00FF00FB</rfid>
    </Item>
- <Item>
  <barcode>2526451354223219884032257</barcode>
  <epc>urn:epc:id:giai:252645135.4223219884032257</epc>
  <tag>urn:epc:tag:giai-96:0.252645135.4223219884032257</tag>
  <rfid>340CF0F0F0FF00FF00FF0101</rfid>
    </Item>
    </EPCListResult>
    </Order>
```

9 Implementation View

9.1 Database & Application Server Consideration

9.1.1 MySQL

MySQL is the world's most popular open-source database. Its advantages include fast performance, high reliability, high usability and high quality technical support. MySQL has a good support for different OS, such as Windows, Linux, HP-UX, AIX, etc. Moreover, it also support common database APIs such as ODBC and JDBC which ease the development tasks. For example, MySQL works smoothly with data of total size over 7TB, 100 millions of row, without service degrading. The use of MySQL is a feasible solution.

9.1.2 Apache Tomcat

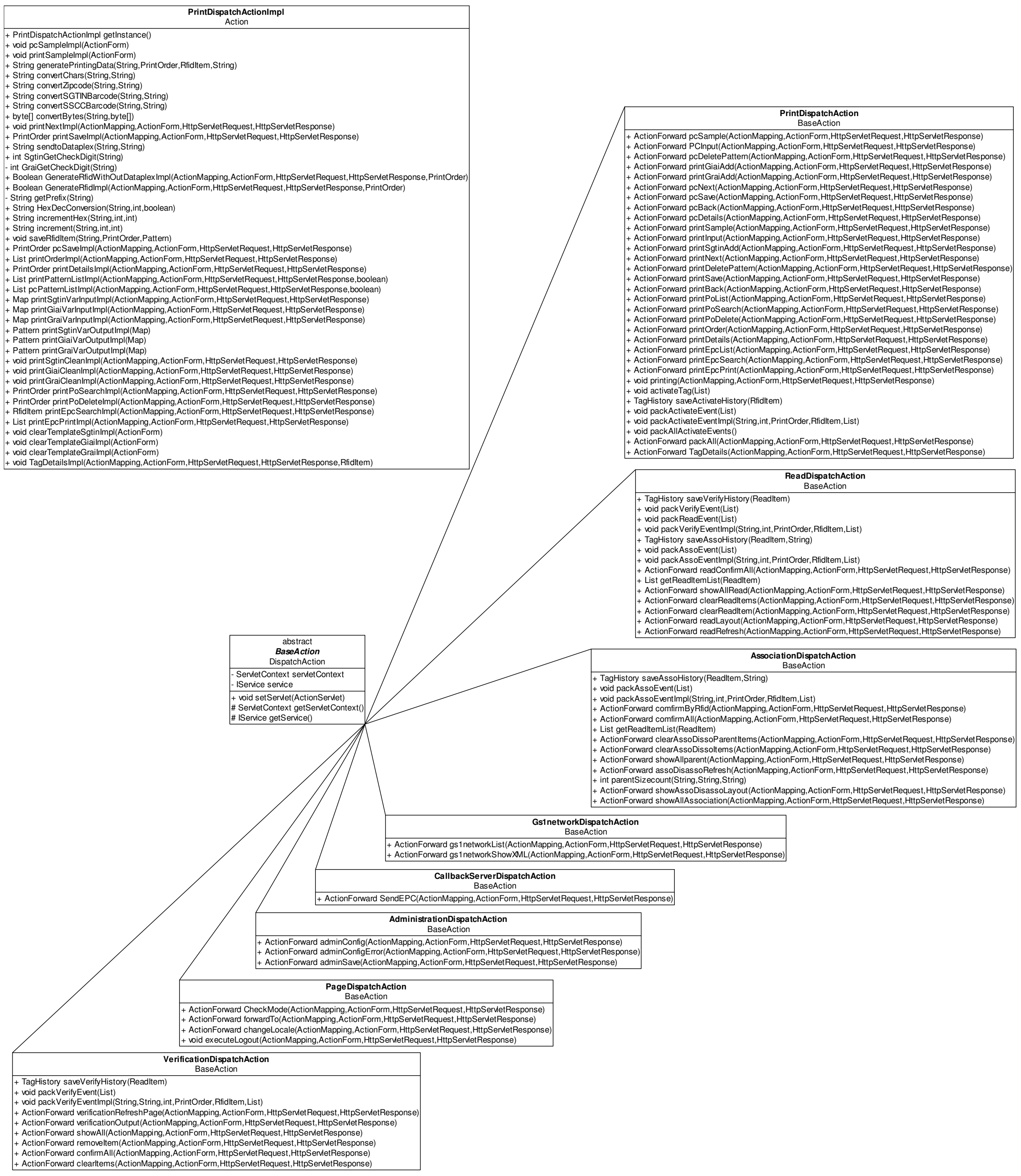
Apache Tomcat is a servlet container developed by the Apache Software Foundation (ASF). Tomcat implements the Java Servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, and provides a "pure Java" HTTP web server environment for Java code to run. Apache Tomcat includes tools for configuration and management, but can also be configured by editing XML configuration files.

9.2 Class Diagram

Here the methods and properties of the classes within the packages are shown in details.

9.2.1 org.rfidee.web

9.2.1.1 action



```
PrintDispatchActionImpl
Action
+ PrintDispatchActionImpl getInstance()
+ void pcSampleImpl(ActionForm)
+ void printSampleImpl(ActionForm)
+ String generatePrintingData(String,PrintOrder,RfidItem,String)
+ String convertChars(String,String)
+ String convertZipcode(String,String)
+ String convertSGTINBarcode(String,String)
+ String convertSSCCBarcode(String,String)
+ byte[] convertBytes(String,byte[])
+ void printNextImpl(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ PrintOrder printSaveImpl(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ String sendtoDataplex(String,String)
+ int SgtinGetCheckDigit(String)
- int GraiGetCheckDigit(String)
+ Boolean GenerateRfidWithOutDataplexImpl(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse,PrintOrder)
+ Boolean GenerateRfidImpl(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse,PrintOrder)
- String getPrefix(String)
+ String HexDecConversion(String,int,boolean)
+ String incrementHex(String,int,int)
+ String increment(String,int,int)
+ void saveRfidItem(String,PrintOrder,Pattern)
+ PrintOrder pcSaveImpl(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ List printOrderImpl(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ PrintOrder printDetailsImpl(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ List printPatternListImpl(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse,boolean)
+ List pcPatternListImpl(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse,boolean)
+ Map printSgtinVarInputImpl(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ Map printGiaiVarInputImpl(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ Map printGraiVarInputImpl(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ Pattern printSgtinVarOutputImpl(Map)
+ Pattern printGiaiVarOutputImpl(Map)
+ Pattern printGraiVarOutputImpl(Map)
+ void printSgtinCleanImpl(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ void printGiaiCleanImpl(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ void printGraiCleanImpl(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ PrintOrder printPoSearchImpl(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ PrintOrder printPoDeleteImpl(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ RfidItem printEpcSearchImpl(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ List printEpcPrintImpl(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ void clearTemplateSgtinImpl(ActionForm)
+ void clearTemplateGiaiImpl(ActionForm)
+ void clearTemplateGraiImpl(ActionForm)
+ void TagDetailsImpl(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse,RfidItem)
```

```
PrintDispatchAction
BaseAction
+ ActionForward pcSample(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward PCInput(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward pcDeletePattern(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward printGiaiAdd(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward printGraiAdd(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward pcNext(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward pcSave(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward pcBack(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward pcDetails(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward printSample(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward printInput(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward printSgtinAdd(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward printNext(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward printDeletePattern(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward printSave(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward printBack(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward printPoList(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward printPoSearch(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward printPoDelete(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward printOrder(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward printDetails(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward printEpcList(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward printEpcSearch(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward printEpcPrint(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ void printing(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ void activateTag(List)
+ TagHistory saveActivateHistory(RfidItem)
+ void packActivateEvent(List)
+ void packActivateEventImpl(String,int,PrintOrder,RfidItem,List)
+ void packAllActivateEvents()
+ ActionForward packAll(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward TagDetails(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
```

```
ReadDispatchAction
BaseAction
+ TagHistory saveVerifyHistory(ReadItem)
+ void packVerifyEvent(List)
+ void packReadEvent(List)
+ void packVerifyEventImpl(String,int,PrintOrder,RfidItem,List)
+ TagHistory saveAssoHistory(ReadItem,String)
+ void packAssoEvent(List)
+ void packAssoEventImpl(String,int,PrintOrder,RfidItem,List)
+ ActionForward readConfirmAll(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ List getReadItemList(ReadItem)
+ ActionForward showAllRead(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward clearReadItems(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward clearReadItem(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward readLayout(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward readRefresh(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
```

```
abstract
BaseAction
DispatchAction
- ServletContext servletContext
- IService service
+ void setServlet(ActionServlet)
# ServletContext getServletContext()
# IService getService()
```

```
AssociationDispatchAction
BaseAction
+ TagHistory saveAssoHistory(ReadItem,String)
+ void packAssoEvent(List)
+ void packAssoEventImpl(String,int,PrintOrder,RfidItem,List)
+ ActionForward comfirmByRfid(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward comfirmAll(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ List getReadItemList(ReadItem)
+ ActionForward clearAssoDissoParentItems(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward clearAssoDissoItems(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward showAllparent(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward assoDisassoRefresh(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ int parentSizecount(String,String,String)
+ ActionForward showAssoDisassoLayout(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward showAllAssociation(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
```

```
Gs1networkDispatchAction
BaseAction
+ ActionForward gs1networkList(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward gs1networkShowXML(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
```

```
CallbackServerDispatchAction
BaseAction
+ ActionForward SendEPC(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
```

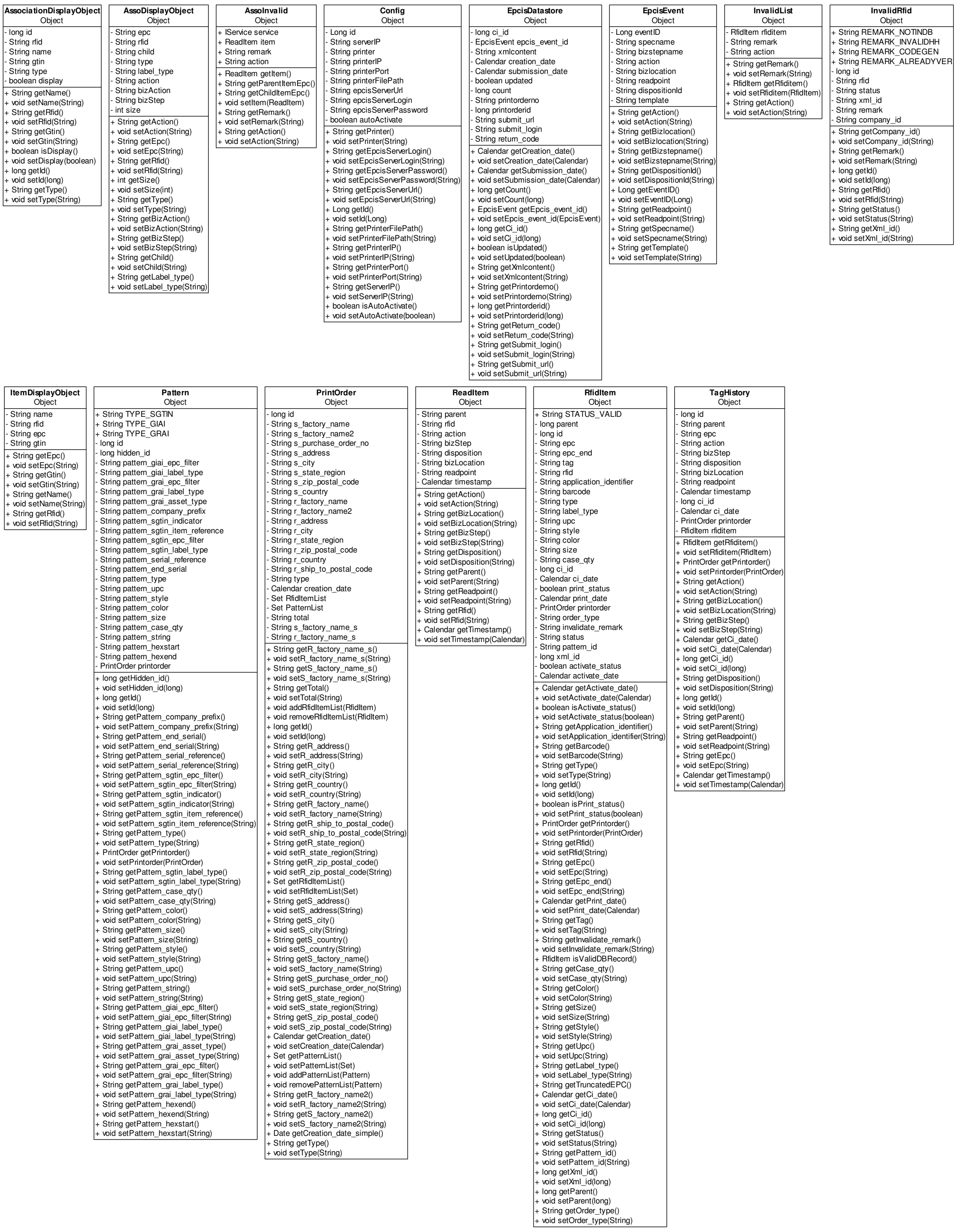
```
AdministrationDispatchAction
BaseAction
+ ActionForward adminConfig(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward adminConfigError(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward adminSave(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
```

```
PageDispatchAction
BaseAction
+ ActionForward CheckMode(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward forwardTo(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward changeLocale(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ void executeLogout(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
```

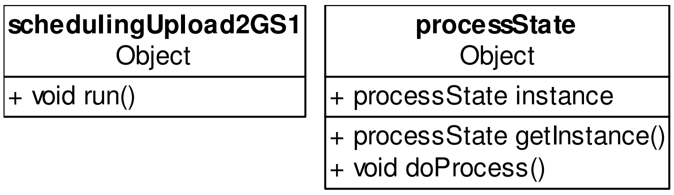
```
VerificationDispatchAction
BaseAction
+ TagHistory saveVerifyHistory(ReadItem)
+ void packVerifyEvent(List)
+ void packVerifyEventImpl(String,String,int,PrintOrder,RfidItem,List)
+ ActionForward verificationRefreshPage(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward verificationOutput(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward showAll(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward removeItem(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward confirmAll(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
+ ActionForward clearItems(ActionMapping,ActionForm,HttpServletRequest,HttpServletResponse)
```

9.2.2 org.rfidee.common

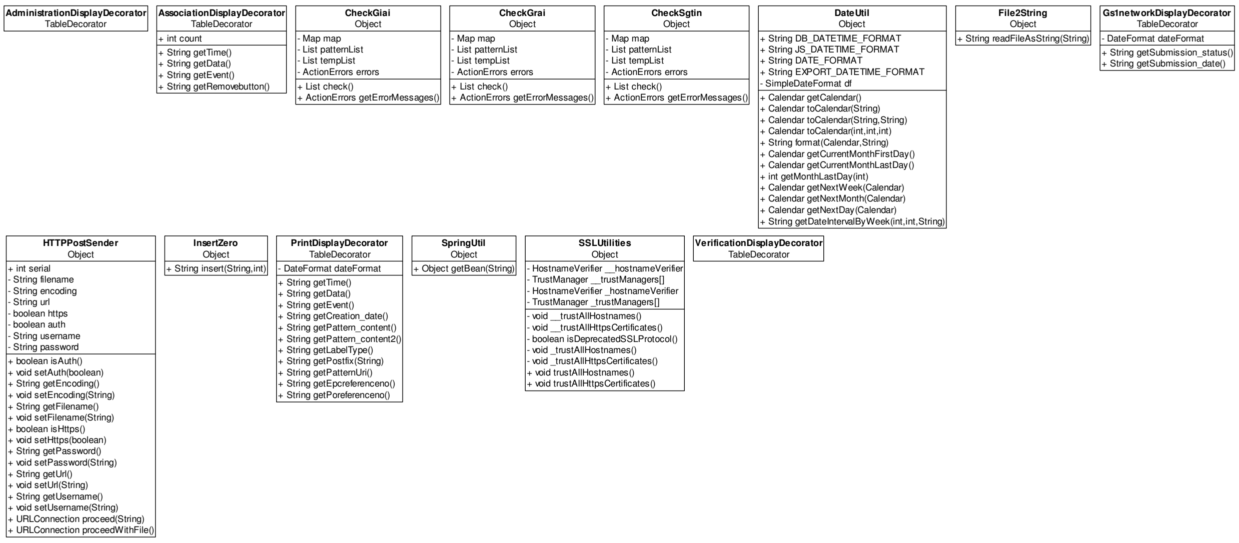
9.2.2.1 bo


AssociationDisplayObject
Object
- long id
- String rfid
- String name
- String gtin
- String type
- boolean display
+ String getName()
+ void setName(String)
+ String getRfid()
+ void setRfid(String)
+ String getGtin()
+ void setGtin(String)
+ boolean isDisplay()
+ void setDisplay(boolean)
+ long getId()
+ void setId(long)
+ String getType()
+ void setType(String)
AssoDisplayObject
Object
- String epc
- String rfid
- String child
- String type
- String label_type
- String action
- String bizAction
- String bizStep
- int size
+ String getAction()
+ void setAction(String)
+ String getEpc()
+ void setEpc(String)
+ String getRfid()
+ void setRfid(String)
+ int getSize()
+ void setSize(int)
+ String getType()
+ void setType(String)
+ String getBizAction()
+ void setBizAction(String)
+ String getBizStep()
+ void setBizStep(String)
+ String getChild()
+ void setChild(String)
+ String getLabel_type()
+ void setLabel_type(String)
AssoInvalid
Object
+ IService service
+ ReadItem item
+ String remark
+ String action
+ ReadItem getItem()
+ String getParentItemEpc()
+ String getChildItemEpc()
+ void setItem(ReadItem)
+ String getRemark()
+ void setRemark(String)
+ String getAction()
+ void setAction(String)
Config
Object
- Long id
- String serverIP
- String printer
- String printerIP
- String printerPort
- String printerFilePath
- String epcisServerUrl
- String epcisServerLogin
- String epcisServerPassword
- boolean autoActivate
+ String getPrinter()
+ void setPrinter(String)
+ String getEpcisServerLogin()
+ void setEpcisServerLogin(String)
+ String getEpcisServerPassword()
+ void setEpcisServerPassword(String)
+ String getEpcisServerUrl()
+ void setEpcisServerUrl(String)
+ Long getId()
+ void setId(Long)
+ String getPrinterFilePath()
+ void setPrinterFilePath(String)
+ String getPrinterIP()
+ void setPrinterIP(String)
+ String getPrinterPort()
+ void setPrinterPort(String)
+ String getServerIP()
+ void setServerIP(String)
+ boolean isAutoActivate()
+ void setAutoActivate(boolean)
EpcisDatastore
Object
- long ci_id
- EpcisEvent epcis_event_id
- String xmlcontent
- Calendar creation_date
- Calendar submission_date
- boolean updated
- long count
- String printorderno
- long printorderid
- String submit_url
- String submit_login
- String return_code
+ Calendar getCreation_date()
+ void setCreation_date(Calendar)
+ Calendar getSubmission_date()
+ void setSubmission_date(Calendar)
+ long getCount()
+ void setCount(long)
+ EpcisEvent getEpcis_event_id()
+ void setEpcis_event_id(EpcisEvent)
+ long getCi_id()
+ void setCi_id(long)
+ boolean isUpdated()
+ void setUpdated(boolean)
+ String getXmlcontent()
+ void setXmlcontent(String)
+ String getPrintorderno()
+ void setPrintorderno(String)
+ long getPrintorderid()
+ void setPrintorderid(long)
+ String getReturn_code()
+ void setReturn_code(String)
+ String getSubmit_login()
+ void setSubmit_login(String)
+ String getSubmit_url()
+ void setSubmit_url(String)
EpcisEvent
Object
- Long eventID
- String specname
- String bizstepname
- String action
- String bizlocation
- String readpoint
- String dispositionId
- String template
+ String getAction()
+ void setAction(String)
+ String getBizlocation()
+ void setBizlocation(String)
+ String getBizstepname()
+ void setBizstepname(String)
+ String getDispositionId()
+ void setDispositionId(String)
+ Long getEventID()
+ void setEventID(Long)
+ String getReadpoint()
+ void setReadpoint(String)
+ String getSpecname()
+ void setSpecname(String)
+ String getTemplate()
+ void setTemplate(String)
InvalidList
Object
- RfidItem rfiditem
- String remark
- String action
+ String getRemark()
+ void setRemark(String)
+ RfidItem getRfiditem()
+ void setRfiditem(RfidItem)
+ String getAction()
+ void setAction(String)
InvalidRfid
Object
+ String REMARK_NOTINDB
+ String REMARK_INVALIDHH
+ String REMARK_CODEGEN
+ String REMARK_ALREADYVER
- long id
- String rfid
- String status
- String xml_id
- String remark
- String company_id
+ String getCompany_id()
+ void setCompany_id(String)
+ String getRemark()
+ void setRemark(String)
+ long getId()
+ void setId(long)
+ String getRfid()
+ void setRfid(String)
+ String getStatus()
+ void setStatus(String)
+ String getXml_id()
+ void setXml_id(String)
ItemDisplayObject
Object
- String name
- String rfid
- String epc
- String gtin
+ String getEpc()
+ void setEpc(String)
+ String getGtin()
+ void setGtin(String)
+ String getName()
+ void setName(String)
+ String getRfid()
+ void setRfid(String)
Pattern
Object
+ String TYPE_SGTIN
+ String TYPE_GIAI
+ String TYPE_GRAI
- long id
- long hidden_id
- String pattern_giai_epc_filter
- String pattern_giai_label_type
- String pattern_grai_epc_filter
- String pattern_grai_label_type
- String pattern_grai_asset_type
- String pattern_company_prefix
- String pattern_sgtin_indicator
- String pattern_sgtin_item_reference
- String pattern_sgtin_epc_filter
- String pattern_sgtin_label_type
- String pattern_serial_reference
- String pattern_end_serial
- String pattern_type
- String pattern_upc
- String pattern_style
- String pattern_color
- String pattern_size
- String pattern_case_qty
- String pattern_string
- String pattern_hexstart
- String pattern_hexend
- PrintOrder printorder
+ long getHidden_id()
+ void setHidden_id(long)
+ long getId()
+ void setId(long)
+ String getPattern_company_prefix()
+ void setPattern_company_prefix(String)
+ String getPattern_end_serial()
+ void setPattern_end_serial(String)
+ String getPattern_serial_reference()
+ void setPattern_serial_reference(String)
+ String getPattern_sgtin_epc_filter()
+ void setPattern_sgtin_epc_filter(String)
+ String getPattern_sgtin_indicator()
+ void setPattern_sgtin_indicator(String)
+ String getPattern_sgtin_item_reference()
+ void setPattern_sgtin_item_reference(String)
+ String getPattern_type()
+ void setPattern_type(String)
+ PrintOrder getPrintorder()
+ void setPrintorder(PrintOrder)
+ String getPattern_sgtin_label_type()
+ void setPattern_sgtin_label_type(String)
+ String getPattern_case_qty()
+ void setPattern_case_qty(String)
+ String getPattern_color()
+ void setPattern_color(String)
+ String getPattern_size()
+ void setPattern_size(String)
+ String getPattern_style()
+ void setPattern_style(String)
+ String getPattern_upc()
+ void setPattern_upc(String)
+ String getPattern_string()
+ void setPattern_string(String)
+ String getPattern_giai_epc_filter()
+ void setPattern_giai_epc_filter(String)
+ String getPattern_giai_label_type()
+ void setPattern_giai_label_type(String)
+ String getPattern_grai_asset_type()
+ void setPattern_grai_asset_type(String)
+ String getPattern_grai_epc_filter()
+ void setPattern_grai_epc_filter(String)
+ String getPattern_grai_label_type()
+ void setPattern_grai_label_type(String)
+ String getPattern_hexend()
+ void setPattern_hexend(String)
+ String getPattern_hexstart()
+ void setPattern_hexstart(String)
PrintOrder
Object
- long id
- String s_factory_name
- String s_factory_name2
- String s_purchase_order_no
- String s_address
- String s_city
- String s_state_region
- String s_zip_postal_code
- String s_country
- String r_factory_name
- String r_factory_name2
- String r_address
- String r_city
- String r_state_region
- String r_zip_postal_code
- String r_country
- String r_ship_to_postal_code
- String type
- Calendar creation_date
- Set RfidItemList
- Set PatternList
- String total
- String s_factory_name_s
- String r_factory_name_s
+ String getR_factory_name_s()
+ void setR_factory_name_s(String)
+ String getS_factory_name_s()
+ void setS_factory_name_s(String)
+ String getTotal()
+ void setTotal(String)
+ void addRfidItemList(RfidItem)
+ void removeRfidItemList(RfidItem)
+ long getId()
+ void setId(long)
+ String getR_address()
+ void setR_address(String)
+ String getR_city()
+ void setR_city(String)
+ String getR_country()
+ void setR_country(String)
+ String getR_factory_name()
+ void setR_factory_name(String)
+ String getR_ship_to_postal_code()
+ void setR_ship_to_postal_code(String)
+ String getR_state_region()
+ void setR_state_region(String)
+ String getR_zip_postal_code()
+ void setR_zip_postal_code(String)
+ Set getRfidItemList()
+ void setRfidItemList(Set)
+ String getS_address()
+ void setS_address(String)
+ String getS_city()
+ void setS_city(String)
+ String getS_country()
+ void setS_country(String)
+ String getS_factory_name()
+ void setS_factory_name(String)
+ String getS_purchase_order_no()
+ void setS_purchase_order_no(String)
+ String getS_state_region()
+ void setS_state_region(String)
+ String getS_zip_postal_code()
+ void setS_zip_postal_code(String)
+ Calendar getCreation_date()
+ void setCreation_date(Calendar)
+ Set getPatternList()
+ void setPatternList(Set)
+ void addPatternList(Pattern)
+ void removePatternList(Pattern)
+ String getR_factory_name2()
+ void setR_factory_name2(String)
+ String getS_factory_name2()
+ void setS_factory_name2(String)
+ Date getCreation_date_simple()
+ String getType()
+ void setType(String)
ReadItem
Object
- String parent
- String rfid
- String action
- String bizStep
- String disposition
- String bizLocation
- String readpoint
- Calendar timestamp
+ String getAction()
+ void setAction(String)
+ String getBizLocation()
+ void setBizLocation(String)
+ String getBizStep()
+ void setBizStep(String)
+ String getDisposition()
+ void setDisposition(String)
+ String getParent()
+ void setParent(String)
+ String getReadpoint()
+ void setReadpoint(String)
+ String getRfid()
+ void setRfid(String)
+ Calendar getTimestamp()
+ void setTimestamp(Calendar)
RfidItem
Object
+ String STATUS_VALID
- long parent
- long id
- String epc
- String epc_end
- String tag
- String rfid
- String application_identifier
- String barcode
- String type
- String label_type
- String upc
- String style
- String color
- String size
- String case_qty
- long ci_id
- Calendar ci_date
- boolean print_status
- Calendar print_date
- PrintOrder printorder
- String order_type
- String invalidate_remark
- String status
- String pattern_id
- long xml_id
- boolean activate_status
- Calendar activate_date
+ Calendar getActivate_date()
+ void setActivate_date(Calendar)
+ boolean isActivate_status()
+ void setActivate_status(boolean)
+ String getApplication_identifier()
+ void setApplication_identifier(String)
+ String getBarcode()
+ void setBarcode(String)
+ String getType()
+ void setType(String)
+ long getId()
+ void setId(long)
+ boolean isPrint_status()
+ void setPrint_status(boolean)
+ PrintOrder getPrintorder()
+ void setPrintorder(PrintOrder)
+ String getRfid()
+ void setRfid(String)
+ String getEpc()
+ void setEpc(String)
+ String getEpc_end()
+ void setEpc_end(String)
+ Calendar getPrint_date()
+ void setPrint_date(Calendar)
+ String getTag()
+ void setTag(String)
+ String getInvalidate_remark()
+ void setInvalidate_remark(String)
+ RfidItem isValidDBRecord()
+ String getCase_qty()
+ void setCase_qty(String)
+ String getColor()
+ void setColor(String)
+ String getSize()
+ void setSize(String)
+ String getStyle()
+ void setStyle(String)
+ String getUpc()
+ void setUpc(String)
+ String getLabel_type()
+ void setLabel_type(String)
+ String getTruncatedEPC()
+ Calendar getCi_date()
+ void setCi_date(Calendar)
+ long getCi_id()
+ void setCi_id(long)
+ String getStatus()
+ void setStatus(String)
+ String getPattern_id()
+ void setPattern_id(String)
+ long getXml_id()
+ void setXml_id(long)
+ long getParent()
+ void setParent(long)
+ String getOrder_type()
+ void setOrder_type(String)
TagHistory
Object
- long id
- String parent
- String epc
- String action
- String bizStep
- String disposition
- String bizLocation
- String readpoint
- Calendar timestamp
- long ci_id
- Calendar ci_date
- PrintOrder printorder
- RfidItem rfiditem
+ RfidItem getRfiditem()
+ void setRfiditem(RfidItem)
+ PrintOrder getPrintorder()
+ void setPrintorder(PrintOrder)
+ String getAction()
+ void setAction(String)
+ String getBizLocation()
+ void setBizLocation(String)
+ String getBizStep()
+ void setBizStep(String)
+ Calendar getCi_date()
+ void setCi_date(Calendar)
+ long getCi_id()
+ void setCi_id(long)
+ String getDisposition()
+ void setDisposition(String)
+ long getId()
+ void setId(long)
+ String getParent()
+ void setParent(String)
+ String getReadpoint()
+ void setReadpoint(String)
+ String getEpc()
+ void setEpc(String)
+ Calendar getTimestamp()
+ void setTimestamp(Calendar)

9.2.2.2 quartz

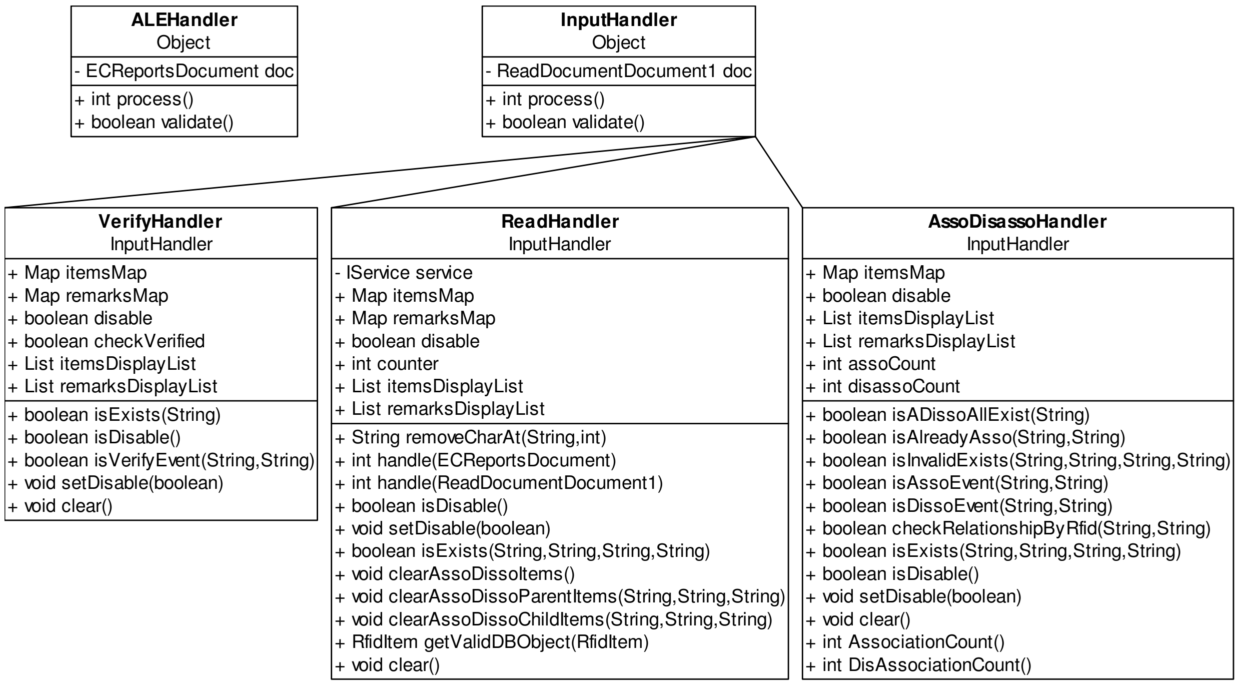


9.2.2.3 util


AdministrationDisplayDecorator
TableDecorator
AssociationDisplayDecorator
TableDecorator
+ int count
+ String getTime()
+ String getData()
+ String getEvent()
+ String getRemovebutton()
CheckGiai
Object
- Map map
- List patternList
- List tempList
- ActionErrors errors
+ List check()
+ ActionErrors getErrorMessages()
CheckGrai
Object
- Map map
- List patternList
- List tempList
- ActionErrors errors
+ List check()
+ ActionErrors getErrorMessages()
CheckSgtin
Object
- Map map
- List patternList
- List tempList
- ActionErrors errors
+ List check()
+ ActionErrors getErrorMessages()
DateUtil
Object
+ String DB_DATETIME_FORMAT
+ String JS_DATETIME_FORMAT
+ String DATE_FORMAT
+ String EXPORT_DATETIME_FORMAT
- SimpleDateFormat df
+ Calendar getCalendar()
+ Calendar toCalendar(String)
+ Calendar toCalendar(String,String)
+ Calendar toCalendar(int,int,int)
+ String format(Calendar,String)
+ Calendar getCurrentMonthFirstDay()
+ Calendar getCurrentMonthLastDay()
+ int getMonthLastDay(int)
+ Calendar getNextWeek(Calendar)
+ Calendar getNextMonth(Calendar)
+ Calendar getNextDay(Calendar)
+ String getDateIntervalByWeek(int,int,String)
File2String
Object
+ String readFileAsString(String)
Gs1networkDisplayDecorator
TableDecorator
- DateFormat dateFormat
+ String getSubmission_status()
+ String getSubmission_date()
HTTPPostSender
Object
+ int serial
- String filename
- String encoding
- String url
- boolean https
- boolean auth
- String username
- String password
+ boolean isAuth()
+ void setAuth(boolean)
+ String getEncoding()
+ void setEncoding(String)
+ String getFilename()
+ void setFilename(String)
+ boolean isHttps()
+ void setHttps(boolean)
+ String getPassword()
+ void setPassword(String)
+ String getUrl()
+ void setUrl(String)
+ String getUsername()
+ void setUsername(String)
+ URLConnection proceed(String)
+ URLConnection proceedWithFile()
InsertZero
Object
+ String insert(String,int)
PrintDisplayDecorator
TableDecorator
- DateFormat dateFormat
+ String getTime()
+ String getData()
+ String getEvent()
+ String getCreation_date()
+ String getPattern_content()
+ String getPattern_content2()
+ String getLabelType()
+ String getPostfix(String)
+ String getPatternUri()
+ String getEpcreferenceno()
+ String getPoreferenceno()
SpringUtil
Object
+ Object getBean(String)
SSLUtilities
Object
- HostnameVerifier __hostnameVerifier
- TrustManager __trustManagers[]
- HostnameVerifier _hostnameVerifier
- TrustManager _trustManagers[]
- void __trustAllHostnames()
- void __trustAllHttpsCertificates()
- boolean isDeprecatedSSLProtocol()
- void _trustAllHostnames()
- void _trustAllHttpsCertificates()
+ void trustAllHostnames()
+ void trustAllHttpsCertificates()
VerificationDisplayDecorator
TableDecorator

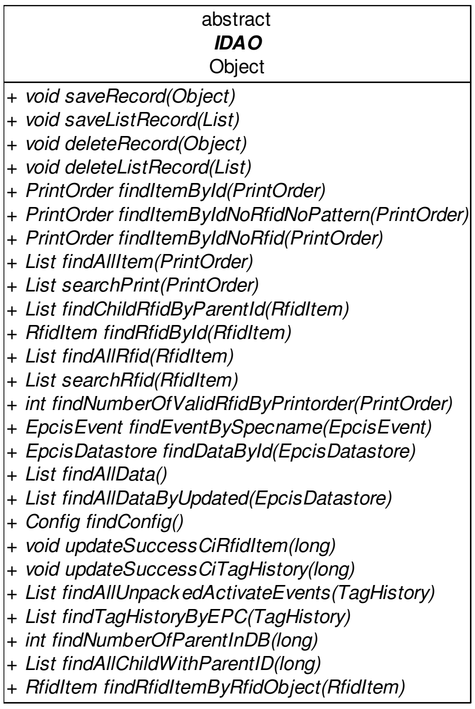
9.2.3 org.rfidee.module

9.2.3.1 alehandler


ALEHandler
Object
- ECReportsDocument doc
+ int process()
+ boolean validate()
InputHandler
Object
- ReadDocumentDocument1 doc
+ int process()
+ boolean validate()
VerifyHandler
InputHandler
+ Map itemsMap
+ Map remarksMap
+ boolean disable
+ boolean checkVerified
+ List itemsDisplayList
+ List remarksDisplayList
+ boolean isExists(String)
+ boolean isDisable()
+ boolean isVerifyEvent(String,String)
+ void setDisable(boolean)
+ void clear()
ReadHandler
InputHandler
- IService service
+ Map itemsMap
+ Map remarksMap
+ boolean disable
+ int counter
+ List itemsDisplayList
+ List remarksDisplayList
+ String removeCharAt(String,int)
+ int handle(ECReportsDocument)
+ int handle(ReadDocumentDocument1)
+ boolean isDisable()
+ void setDisable(boolean)
+ boolean isExists(String,String,String,String)
+ void clearAssoDissoItems()
+ void clearAssoDissoParentItems(String,String,String)
+ void clearAssoDissoChildItems(String,String,String)
+ RfidItem getValidDBObject(RfidItem)
+ void clear()
AssoDisassoHandler
InputHandler
+ Map itemsMap
+ boolean disable
+ List itemsDisplayList
+ List remarksDisplayList
+ int assoCount
+ int disassoCount
+ boolean isADissoAllExist(String)
+ boolean isAlreadyAsso(String,String)
+ boolean isInvalidExists(String,String,String,String)
+ boolean isAssoEvent(String,String)
+ boolean isDissoEvent(String,String)
+ boolean checkRelationshipByRfid(String,String)
+ boolean isExists(String,String,String,String)
+ boolean isDisable()
+ void setDisable(boolean)
+ void clear()
+ int AssociationCount()
+ int DisAssociationCount()

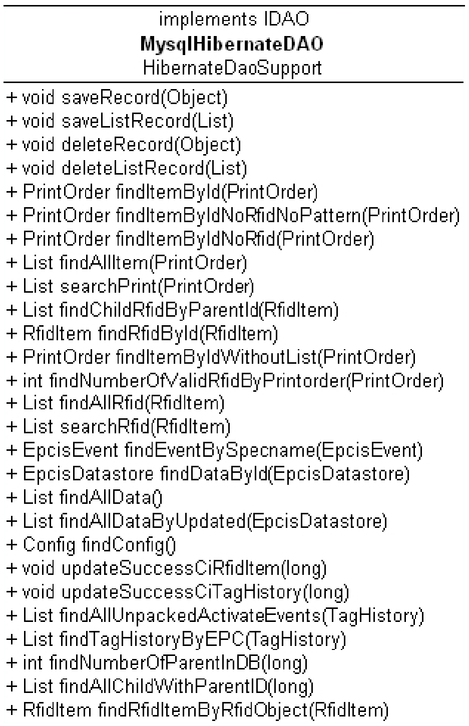
9.2.3.2 dao

9.2.3.2.1 iface



```
abstract
IDAO
Object
+ void saveRecord(Object)
+ void saveListRecord(List)
+ void deleteRecord(Object)
+ void deleteListRecord(List)
+ PrintOrder findItemById(PrintOrder)
+ PrintOrder findItemByIdNoRfidNoPattern(PrintOrder)
+ PrintOrder findItemByIdNoRfid(PrintOrder)
+ List findAllItem(PrintOrder)
+ List searchPrint(PrintOrder)
+ List findChildRfidByParentId(RfidItem)
+ RfidItem findRfidById(RfidItem)
+ List findAllRfid(RfidItem)
+ List searchRfid(RfidItem)
+ int findNumberOfValidRfidByPrintorder(PrintOrder)
+ EpcisEvent findEventBySpecname(EpcisEvent)
+ EpcisDatastore findDataById(EpcisDatastore)
+ List findAllData()
+ List findAllDataByUpdated(EpcisDatastore)
+ Config findConfig()
+ void updateSuccessCiRfidItem(long)
+ void updateSuccessCiTagHistory(long)
+ List findAllUnpackedActivateEvents(TagHistory)
+ List findTagHistoryByEPC(TagHistory)
+ int findNumberOfParentInDB(long)
+ List findAllChildWithParentID(long)
+ RfidItem findRfidItemByRfidObject(RfidItem)
```

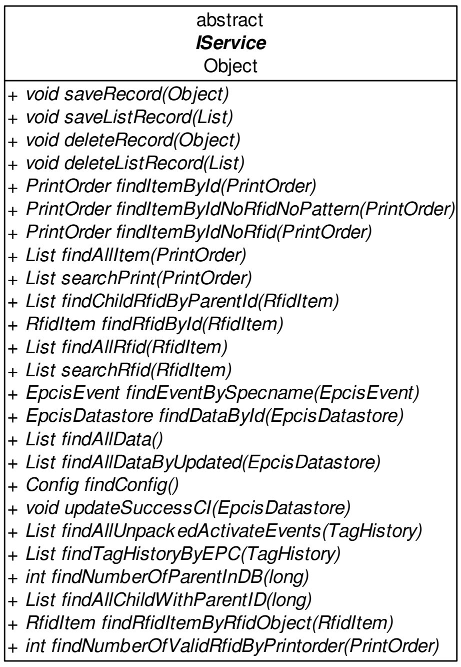
9.2.3.2.2 impl



```
implements IDAO
MysqlHibernateDAO
HibernateDaoSupport
+ void saveRecord(Object)
+ void saveListRecord(List)
+ void deleteRecord(Object)
+ void deleteListRecord(List)
+ PrintOrder findItemById(PrintOrder)
+ PrintOrder findItemByIdNoRfidNoPattern(PrintOrder)
+ PrintOrder findItemByIdNoRfid(PrintOrder)
+ List findAllItem(PrintOrder)
+ List searchPrint(PrintOrder)
+ List findChildRfidByParentId(RfidItem)
+ RfidItem findRfidById(RfidItem)
+ PrintOrder findItemByIdWithoutList(PrintOrder)
+ int findNumberOfValidRfidByPrintorder(PrintOrder)
+ List findAllRfid(RfidItem)
+ List searchRfid(RfidItem)
+ EpcisEvent findEventBySpecname(EpcisEvent)
+ EpcisDatastore findDataById(EpcisDatastore)
+ List findAllData()
+ List findAllDataByUpdated(EpcisDatastore)
+ Config findConfig()
+ void updateSuccessCiRfidItem(long)
+ void updateSuccessCiTagHistory(long)
+ List findAllUnpackedActivateEvents(TagHistory)
+ List findTagHistoryByEPC(TagHistory)
+ int findNumberOfParentInDB(long)
+ List findAllChildWithParentID(long)
+ RfidItem findRfidItemByRfidObject(RfidItem)
```

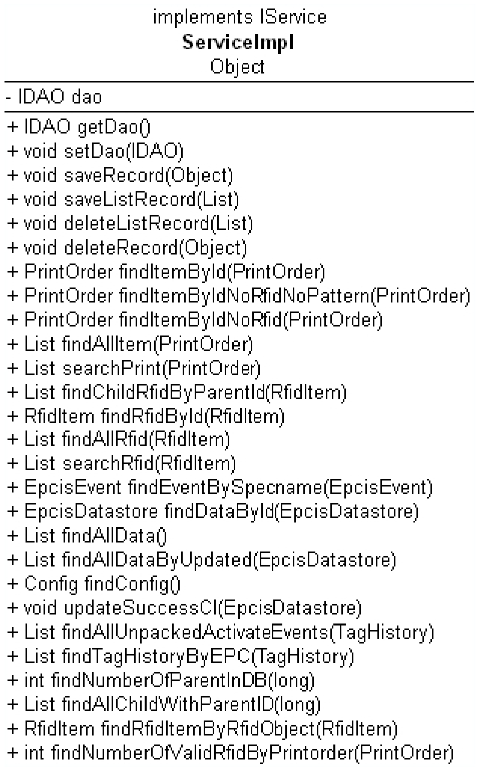
9.2.3.3 service

9.2.3.3.1 iface



```
abstract
IService
Object
+ void saveRecord(Object)
+ void saveListRecord(List)
+ void deleteRecord(Object)
+ void deleteListRecord(List)
+ PrintOrder findItemById(PrintOrder)
+ PrintOrder findItemByIdNoRfidNoPattern(PrintOrder)
+ PrintOrder findItemByIdNoRfid(PrintOrder)
+ List findAllItem(PrintOrder)
+ List searchPrint(PrintOrder)
+ List findChildRfidByParentId(RfidItem)
+ RfidItem findRfidById(RfidItem)
+ List findAllRfid(RfidItem)
+ List searchRfid(RfidItem)
+ EpcisEvent findEventBySpecname(EpcisEvent)
+ EpcisDatastore findDataById(EpcisDatastore)
+ List findAllData()
+ List findAllDataByUpdated(EpcisDatastore)
+ Config findConfig()
+ void updateSuccessCI(EpcisDatastore)
+ List findAllUnpackedActivateEvents(TagHistory)
+ List findTagHistoryByEPC(TagHistory)
+ int findNumberOfParentInDB(long)
+ List findAllChildWithParentID(long)
+ RfidItem findRfidItemByRfidObject(RfidItem)
+ int findNumberOfValidRfidByPrintorder(PrintOrder)
```

9.2.3.3.2 impl



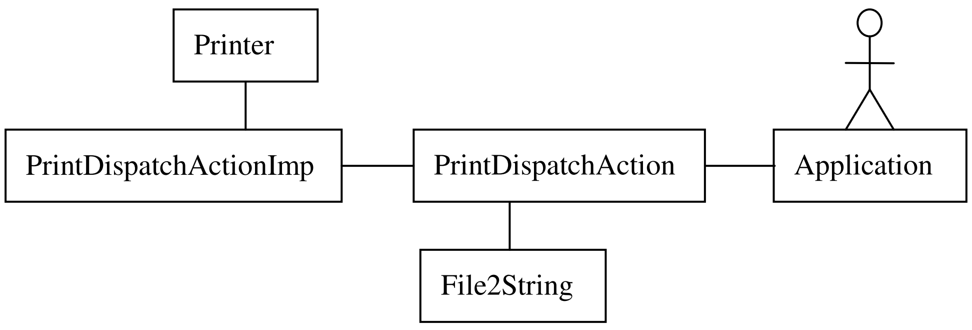
```
implements IService
ServiceImpl
Object
- IDAO dao
+ IDAO getDao()
+ void setDao(IDAO)
+ void saveRecord(Object)
+ void saveListRecord(List)
+ void deleteListRecord(List)
+ void deleteRecord(Object)
+ PrintOrder findItemById(PrintOrder)
+ PrintOrder findItemByIdNoRfidNoPattern(PrintOrder)
+ PrintOrder findItemByIdNoRfid(PrintOrder)
+ List findAllItem(PrintOrder)
+ List searchPrint(PrintOrder)
+ List findChildRfidByParentId(RfidItem)
+ RfidItem findRfidById(RfidItem)
+ List findAllRfid(RfidItem)
+ List searchRfid(RfidItem)
+ EpcisEvent findEventBySpecname(EpcisEvent)
+ EpcisDatastore findDataById(EpcisDatastore)
+ List findAllData()
+ List findAllDataByUpdated(EpcisDatastore)
+ Config findConfig()
+ void updateSuccessCI(EpcisDatastore)
+ List findAllUnpackedActivateEvents(TagHistory)
+ List findTagHistoryByEPC(TagHistory)
+ int findNumberOfParentInDB(long)
+ List findAllChildWithParentID(long)
+ RfidItem findRfidItemByRfidObject(RfidItem)
+ int findNumberOfValidRfidByPrintorder(PrintOrder)
```

9.3 Collaboration Diagram

Here the collaboration diagrams of the main services are shown. The major entities involved in each operation are depicted.

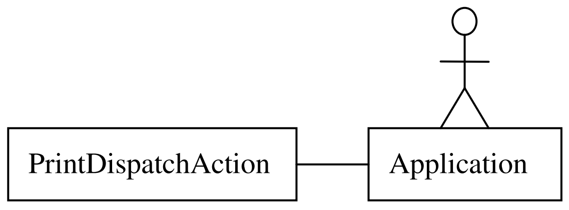
9.3.1 Print Service

In the Print Service, printing would be achieved by the class, PrintDispatchAction with the support classes, File2String and PrintDispatchActionImp. Finally, the class, PrintDispatchActionImp would send the data to the printer for printing.



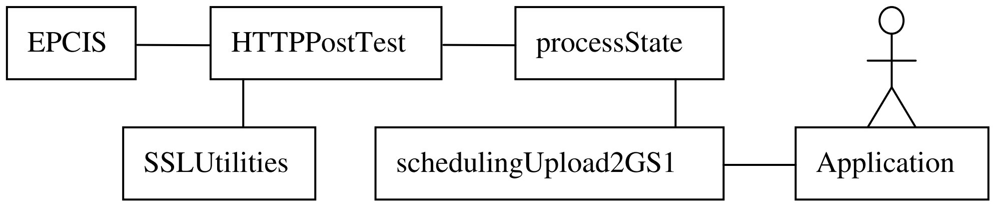
9.3.2 Activate Service

Internal flows are shown in the following graph.



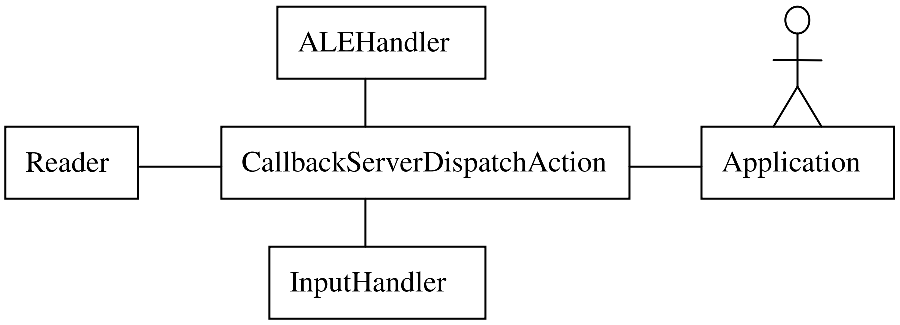
9.3.3 EPCIS Service

In the EPCIS Service, the class, schedulingUpload2GS1 would repeat to trigger the class, processState to run in certain minutes to detect any possible records which need to be uploaded to EPCIS. If there are some records available, the class, processState would upload those records to EPCIS through the class, HTTPPostTest with the support class, SSLUtilities.



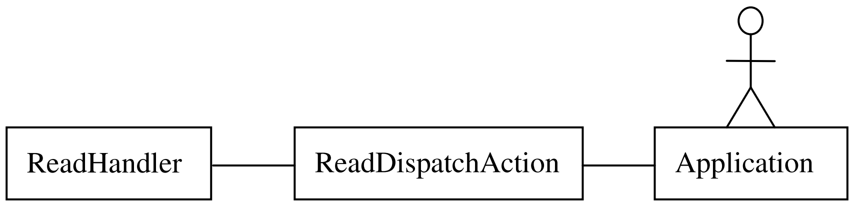
9.3.4 Connection Service

In the Connection Service, the class, CallbackServerDispatchAction is responsible for connecting the reader and receiving the data. Then, it would classify and dispatch to the classes, ALEHandler and InputHandler for validation before processing.



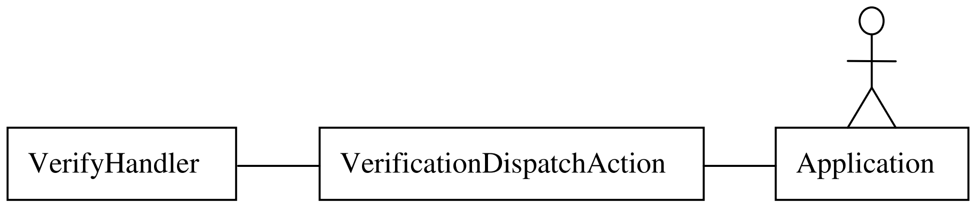
9.3.5 Read Service

In the Read Service, the class, ReadDispatchAction is responsible for classifying, processing and displaying the data from the reader as well as packing the data for uploading to EPCIS. All the action from the class, ReadDispatchAction would be implemented by the class, ReadHandler.



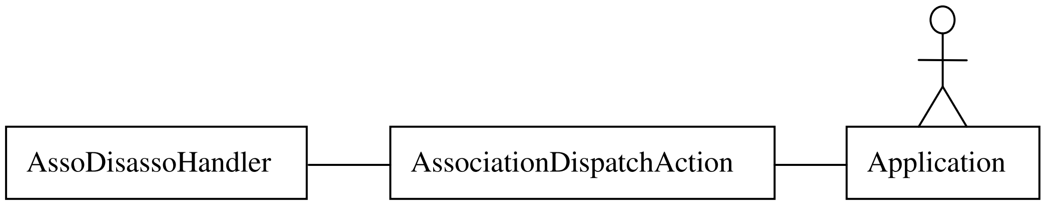
9.3.6 Verify Service

In the Verify Service, the class, VerificationDispatchAction is responsible for classifying, processing and displaying the data from the reader as well as packing the data for uploading to EPCIS. All the action from the class, VerificationDispatchAction would be implemented by the class, VerifyHandler.



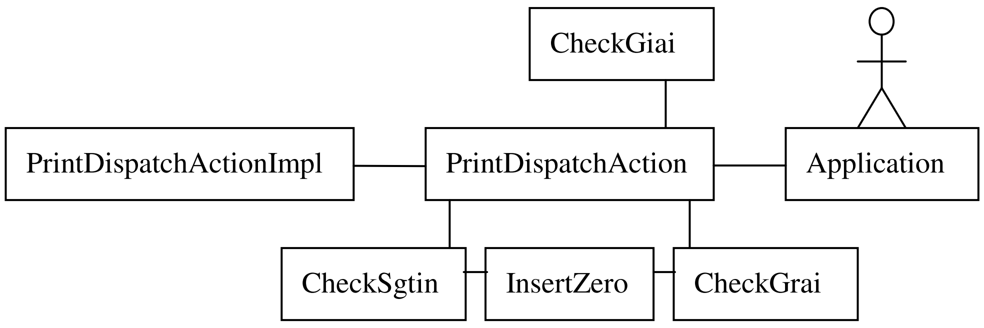
9.3.7 Association/Disassociation Service

In the Association/Disassociation Service, the class, AssociationDispatchAction is responsible for classifying, processing and displaying the data from the reader as well as packing the data for uploading to EPCIS. All the action from the class, AssociationDispatchAction would be implemented by the class, AssoDisassoHandler.



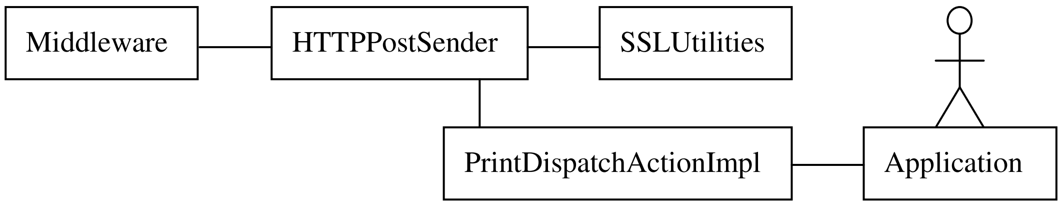
9.3.8 Batch Service

In the Batch Service, the class, PrintDispatchAction is responsible for displaying and processing the batch. All the action from the class, PrintDispatchAction would be implemented by the class, PrintDispatchActionImpl, the checking classes, CheckGiai, CheckSgtin and CheckGrai as well as the pre-process class, InsertZero.



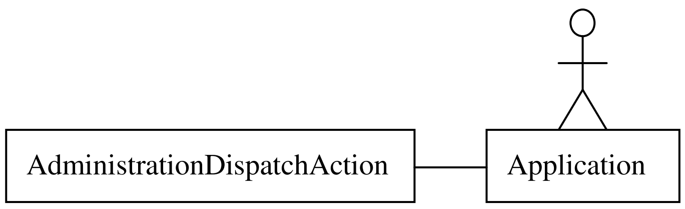
9.3.9 RFID Generation Service

In the RFID Generation Service, the class, PrintDispatchActionImpl handles generating the RFID tag serial number. However, it can also connect to middleware for generating according to the system configuration. The generation jobs would send to middleware and then receive the results through the class, HTTPPostSender with the support class, SSLUtilities.



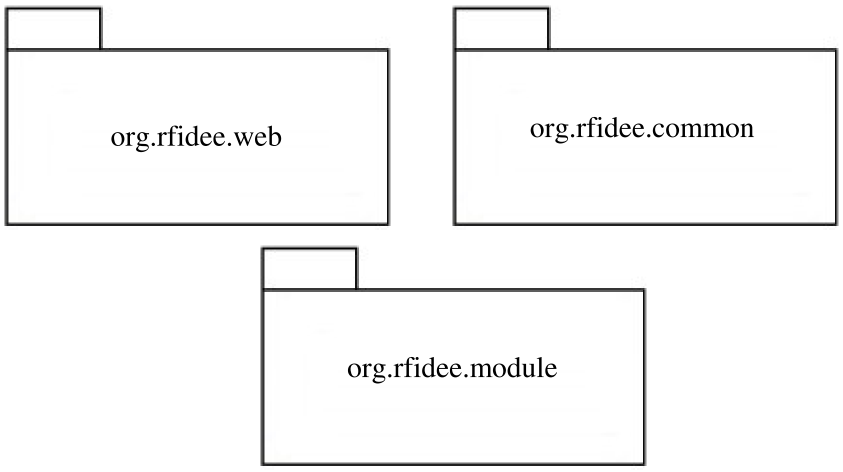
9.3.10 Configuration Service

Internal flows are shown in the following graph.



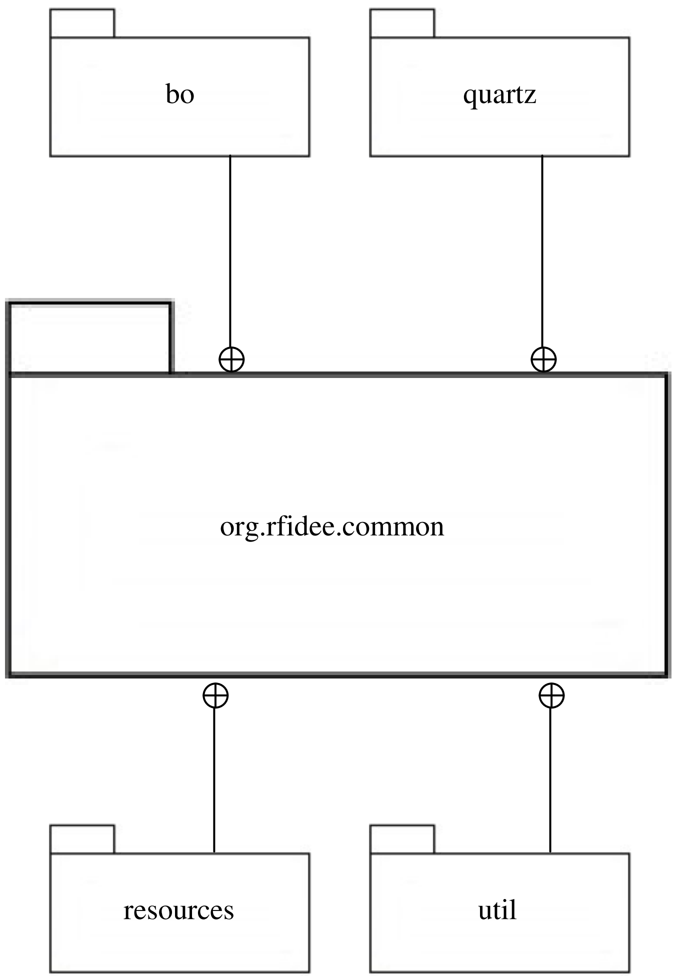
9.4 Package Diagram

RFID Enabling Engine is developed into 3 Java packages: org.rfidee.common, org.rfidee.module and org.rfidee.web.



9.4.1 org.rfidee.common

The "org.rfidee.common" package contains part of the classes which implement the RFID Enabling Engine.

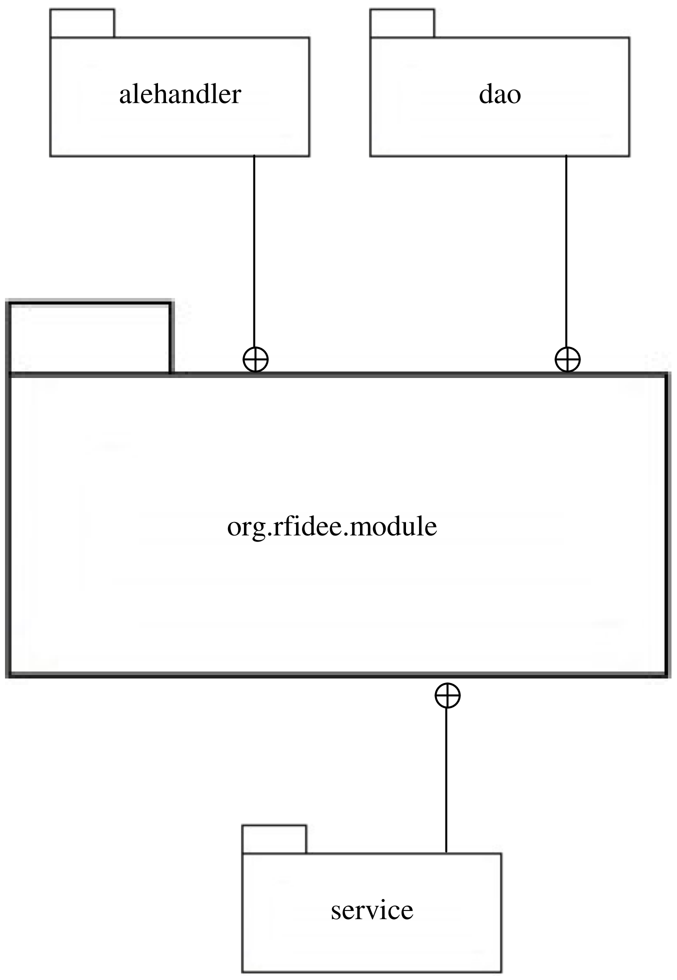


There are packages inside org.rfidee.common:

- bo: contains Data Access Objects, which are responsible for database storage and retrieval of data.
- quartz: contains the schedule classes which run repeatedly to support the main implementation classes to operate.
- resources: contains the multiple language files for layout and display. No any classes.
- util: contains the classes for supporting the main implementation classes to operate.

9.4.2 org.rfidee.module

The “org.rfidee.module” package contains the remaining part of the classes which implement the RFID Enabling Engine.

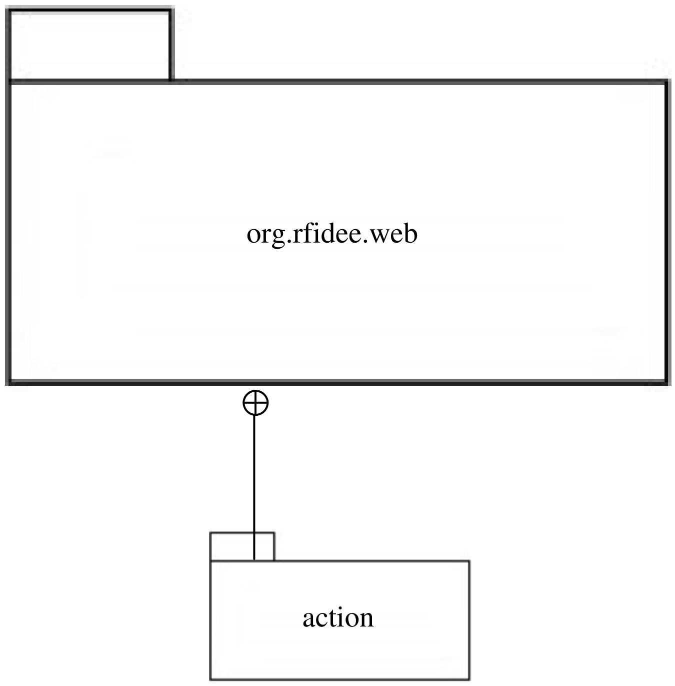


There are packages inside org.rfidee.module:

- alehandler: contains the support classes for supporting the main implementation classes.
- dao: contains the implementation classes for interacting with database storage.
- service: contains the main implementation classes for achieving the main services and functions.

9.4.3 org.rfidee.web

The “org.rfidee.web” package only contains the classes for user web interface and interaction.



There are packages inside org.rfidee.web:

- action: contains the classes for providing user web interface and interaction.

10 Deployment View

10.1 Environment

10.1.1 Setup

The standard development environments include:

- Apache Tomcat 5.5
- Java Development Kit (JDK) 1.5
- MySQL 4.1 +MySQLGUI 5.0

10.1.2 Build & Deployment

- Use Apache Ant 1.71 to build the project
- Put the war file into the webapps directory under Apache Tomcat 5.5 home directory

11 Data View

11.1 Table config

The config table is storing the system configuration.

Column name	Type	Description
Id	BIGINT(20)	Primary key
serverIP	VARCHAR(255)	The server IP of middleware
printerIP	VARCHAR(255)	Printer setting
printerPort	VARCHAR(255)	Printer setting
printerFilePath	VARCHAR(255)	Printer setting
epcisServerUrl	VARCHAR(255)	EPCIS setting
epcisServerLogin	VARCHAR(255)	EPCIS setting
epcisServerPassword	VARCHAR(255)	EPCIS setting
autoActivate	TINYINT(1)	Activate setting
printer	VARCHAR(255)	Printer setting

Primary Key: id

11.2 Table epics_datastore

The epcis_datastore table is storing the records which submit to EPCIS.

Column name	Type	Description
ci_id	BIGINT(20)	Primary key
epcis_event_id	BIGINT(20)	Foreign key – epcis_event
xmlcontent	LONGTEXT	The upload data to EPCIS in XML format
creation_date	DATETIME	The date and time of creation
submission_date	DATETIME	The date and time of submission to EPC network
updated	TINYINT(1)	Submission status
count	BIGINT(20)	Number of EPC elements
printorderno	VARCHAR(255)	Batch No related
printorderid	BIGINT(20)	Foreign key – printorder
submit_url	VARCHAR(255)	Submission destination
submit_login	VARCHAR(255)	Login name to EPC Network
return_code	VARCHAR(255)	HTTP Code of EPC server return

Primary Key: ci_id

11.3 Table epcis_event

The epics_event table is storing the records of EPCIS event.

Column name	Type	Description
eventID	BIGINT(20)	Primary key
specname	VARCHAR(255)	The internal system key, spec name of the event
bizstepname	VARCHAR(255)	The EPCIS key, business step name
action	VARCHAR(255)	The EPCIS key, action of the event
bizlocation	VARCHAR(255)	The EPCIS key, business location
readpoint	VARCHAR(255)	The EPCIS key, read point
dispositionId	VARCHAR(255)	The EPCIS key, disposition ID
template	LONGTEXT	Upload template file

Primary Key: eventID

11.4 Table pattern

The pattern table is storing the information for system internal use.

Column name	Type	Description
id	BIGINT(20)	Primary key
pattern_company_prefix	VARCHAR(255)	Company prefix for generating RFID
pattern_sgtin_indicator	VARCHAR(255)	SGTIN indicator for generating RFID
pattern_sgtin_item_reference	VARCHAR(255)	SGTIN item reference for generating RFID
pattern_sgtin_epc_filter	VARCHAR(255)	EPC filter for generating RFID
pattern_serial_reference	VARCHAR(255)	Serial for generating RFID
pattern_end_serial	VARCHAR(255)	The end of serial for generating RFID
pattern_type	VARCHAR(255)	RFID generation type, SGTIN, GIAI or GRAI
printorder_id	BIGINT(20)	Foreign key – printorder
pattern_sgtin_label_type	VARCHAR(255)	The type of SGTIN, Carton
pattern_upc	VARCHAR(255)	UPC information
pattern_case_qty	VARCHAR(255)	General information field
pattern_style	VARCHAR(255)	General information field
pattern_color	VARCHAR(255)	General information field
pattern_size	VARCHAR(255)	General information field
pattern_string	VARCHAR(255)	Generation result
pattern_giai_label_type	VARCHAR(255)	The type of GIAI, Pallet
pattern_grai_label_type	VARCHAR(255)	The type of GRAI, Container
pattern_grai_asset_type	VARCHAR(255)	The asset type of GRAI

Primary Key: id

11.5 Table printorder

The printorder table is storing the record of each tag input.

Column name	Type	Description
id	BIGINT(20)	Primary key
s_purchase_order_no	VARCHAR(255)	Batch number
s_factory_name	VARCHAR(255)	Sender address
s_address	VARCHAR(255)	Sender address
s_city	VARCHAR(255)	Sender address
s_state_region	VARCHAR(255)	Sender address
s_zip_postal_code	VARCHAR(255)	Sender address
s_country	VARCHAR(255)	Sender address
r_factory_name	VARCHAR(255)	Recipient address
r_address	VARCHAR(255)	Recipient address
r_city	VARCHAR(255)	Recipient address
r_state_region	VARCHAR(255)	Recipient address
r_zip_postal_code	VARCHAR(255)	Recipient address
r_country	VARCHAR(255)	Recipient address
r_ship_to_postal_code	VARCHAR(255)	Recipient address
creation_date	DATETIME	The creation time
total	VARCHAR(255)	The total number of tags
s_factory_name2	VARCHAR(255)	Sender address
r_factory_name2	VARCHAR(255)	Recipient address
s_factory_name_s	VARCHAR(255)	Sender address
r_factory_name_s	VARCHAR(255)	Recipient address
order_type	VARCHAR(255)	The internal system key

Primary Key: id

11.6 Table rfiditem

The rfiditem table is storing all information of each tag.

Column name	Type	Description
id	BIGINT(20)	Primary key
parent	BIGINT(20)	The internal system ID for association and Disassociation
epc	VARCHAR(255)	The EPC number of a tag
tag	VARCHAR(255)	The Tag number of a tag
rfid	VARCHAR(255)	The RFID hex number of a tag
application_identifier	VARCHAR(255)	Application identifier for GIAI and GRAI
barcode	VARCHAR(255)	The barcode of a tag
type	VARCHAR(255)	The type of EPC number
print_status	TINYINT(1)	//0=not printed; 1=printed
print_date	DATETIME	The date and time of printing
printorder_id	BIGINT(20)	Foreign key – printorder
upc	VARCHAR(255)	The UPC of a tag
case_qty	VARCHAR(255)	Tag information
style	VARCHAR(255)	Tag information
size	VARCHAR(255)	Tag information
color	VARCHAR(255)	Tag information
label_type	VARCHAR(255)	The level type //Item = item level; Carton = carton level; Pallet = pallet level; Container = container level
ci_id	BIGINT(20)	Foreign key – epcis_datastore
ci_date	DATETIME	Date of the foreign key
status	VARCHAR(255)	The internal system key
pattern_id	VARCHAR(255)	Foreign key – pattern
xml_id	BIGINT(20)	Reserved field
order_type	VARCHAR(255)	The internal system key
activate_status	TINYINT(1)	//0=not activated; 1=activated
activate_date	DATETIME	Date of activation

Primary Key: id

11.7 Table taghistory

The taghistory table is storing the information sent from reader.

Column name	Type	Description
id	BIGINT(20)	Primary key
parent	VARCHAR(255)	The internal system ID for association and Disassociation
epc	VARCHAR(255)	Data from reader, EPC
action	VARCHAR(255)	Data from reader, action of the event
bizStep	VARCHAR(255)	Data from reader, business step name
disposition	VARCHAR(255)	Data from reader, disposition ID
bizLocation	VARCHAR(255)	Data from reader, business location
readpoint	VARCHAR(255)	Data from reader, read point
timestamp	TIMESTAMP	Timestamp of record
ci_id	BIGINT(20)	Foreign key – epcis_datastore
ci_date	DATETIME	Date of the foreign key
printorder_id	BIGINT(20)	Foreign key – printorder
rfiditem_id	BIGINT(20)	Foreign key – rfiditem

Primary Key: id

12 System Properties

The RFID Enabling Engine is based on the J2EE technology and inherits the important characteristics of its standard.

12.1 Extensibility

The RFID Enabling Engine is built in the object-oriented architecture with well-defined class library. New feature can be added as sub-classes so that they can inherit the feature of the developed classes together with any customized features of the user.

12.2 Stability

As a group of well-tested specification & framework, the RFID Enabling Engine guarantees a stable and concrete architecture of both data and programming source. A specified hardware requirement & system environment setup will introduce as a guideline for keeping the engine running smoothly.

12.3 Diversity

The RFID Enabling Engine not only supports the communication in-between EPCIS interface, but also supports direct (API procedure call) & indirect (via RFID Middleware) communication with RFID equipments.

12.4 Portability

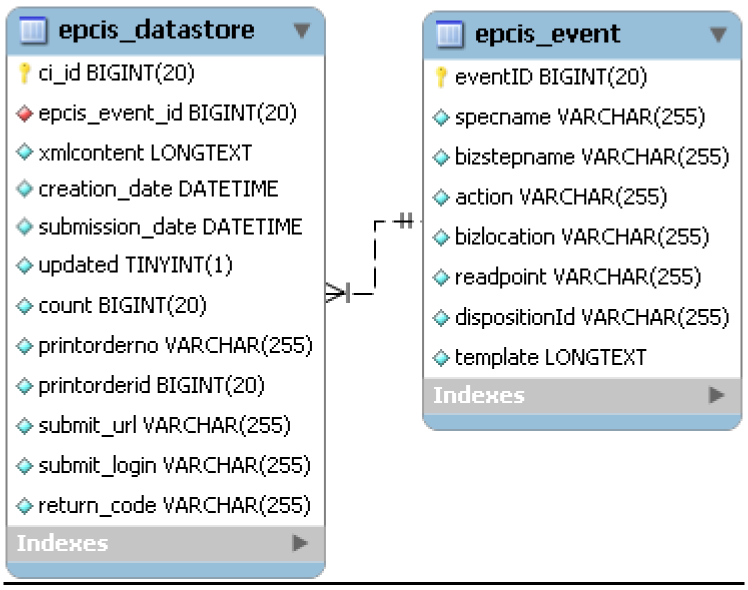
As a platform-independent java web program, the program can be run on different OS which includes Window XP & Linux. As a web program, user can access the RFID Enabling Engine via internet (browser) which ensure numerous users can access the program from all around the world throughout the supply chain.

Appendix A Database Specification

The database of the RFID Enabling Engine will be illustrated in ER Diagram as follow:

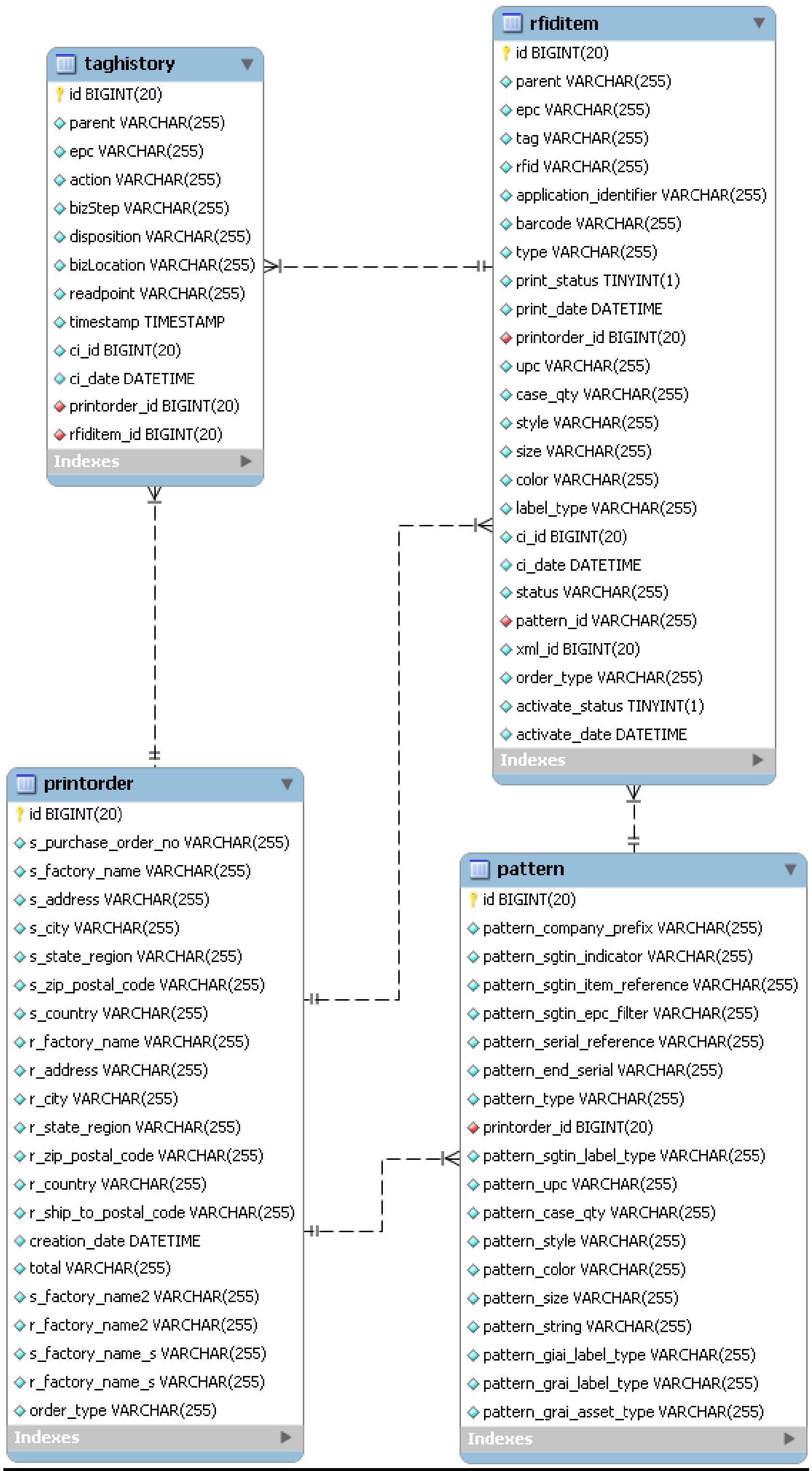
EPCIS Data Structure

The following diagram illustrates the data structure of the EPCIS event data:



Main Data Structure

The following diagram illustrates the data structure of the main modules:



General Data Structure

The following diagram illustrates the data structure of the system configuration:

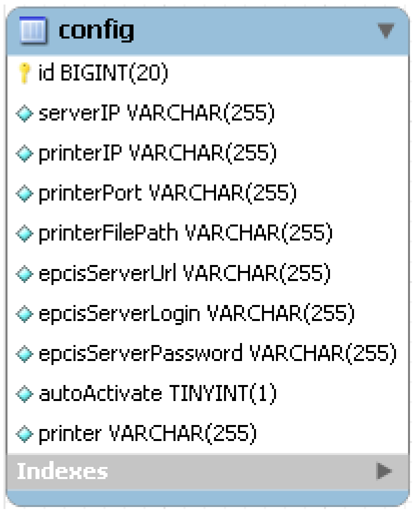


Table List:

Table	Data Description
config	The config table is storing the system configuration.
epcis_datastore	The epcis_datastore table is storing the records which submit to EPCIS.
epcis_event	The epics_event table is storing the records of EPCIS event.
pattern	The pattern table is storing the information for system internal use.
printorder	The printorder table is storing the record of each tag input.
rfiditem	The rfiditem table is storing all information of each tag.
taghistroy	The taghistory table is storing the information sent from reader.

Detailed Table Description:

1. config: The config table is storing the system configuration.

Field name	Data type	Length	Setting	Default	Description
Id	bigint	20	unsigned, Not NULL, Auto increment	NULL	Primary key
serverIP	varchar	255	-	NULL	The server IP of middleware
printerIP	varchar	255	-	NULL	Printer setting
printerPort	varchar	255	-	NULL	Printer setting
printerFilePath	varchar	255	-	NULL	Printer setting
epcisServerUrl	varchar	255	-	NULL	EPCIS setting
epcisServerLogin	varchar	255	-	NULL	EPCIS setting
epcisServerPassword	varchar	255	-	NULL	EPCIS setting
autoActivate	tinyint	1	-	NULL	Activate setting
printer	varchar	255	-	NULL	Printer setting

2. epcis_datastore: The epcis_datastore table is storing the records which submit to EPCIS.

Field name	Data type	Length	Setting	Default	Description
ci_id	bigint	20	unsigned, Not NULL, Auto increment	NULL	Primary key
epcis_event_id	bigint	20	unsigned, Not NULL	0	Foreign key – epcis_event
xmlcontent	longtext	-	-	NULL	The upload data to EPCIS in XML format
creation_date	datetime	-	-	NULL	The date and time of creation
submission_date	datetime	-	-	NULL	The date and time of submission to EPC network
updated	tinyint	1	unsigned	0	Submission status
count	bigint	20	unsigned	NULL	Number of EPC elements
printorderno	varchar	255	-	NULL	Batch No related
printorderid	bigint	20	unsigned	NULL	Foreign key – printorder
submit_url	varchar	255	-	NULL	Submission destination
submit_login	varchar	255	-	NULL	Login name to EPC Network
return_code	varchar	255	-	NULL	HTTP Code of EPC server return

3. epcis_event: The epics_event table is storing the records of EPCIS event.

Field name	Data type	Length	Setting	Default	Description
eventID	bigint	20	unsigned, Not NULL, Auto increment	NULL	Primary key
specname	varchar	255	-	NULL	The internal system key, spec name of the event
bizstepname	varchar	255	-	NULL	The EPCIS key, business step name
action	varchar	255	-	NULL	The EPCIS key, action of the event
bizlocation	varchar	255	-	NULL	The EPCIS key, business location
readpoint	varchar	255	-	NULL	The EPCIS key, read point
dispositionId	varchar	255	-	NULL	The EPCIS key, disposition ID
template	longtext	-	-	NULL	Upload template file

4. pattern: The pattern table is storing the information for system internal use.

Field name	Data type	Length	Setting	Default	Description
id	bigint	20	unsigned, Not NULL, Auto increment	NULL	Primary key
pattern_company_prefix	varchar	255	-	NULL	Company prefix for generating RFID
pattern_sgtin_indicator	varchar	255	-	NULL	SGTIN indicator for generating RFID
pattern_sgtin_item_reference	varchar	255	-	NULL	SGTIN item reference for generating RFID
pattern_sgtin_epc_filter	varchar	255	-	NULL	EPC filter for generating RFID
pattern_serial_reference	varchar	255	-	NULL	Serial for generating RFID
pattern_end_serial	varchar	255	-	NULL	The end of serial for generating RFID
pattern_type	varchar	255	-	NULL	RFID generation type, SGTIN, GIAI or GRAI
printorder_id	bigint	20	unsigned, Not NULL	0	Foreign key – printorder
pattern_sgtin_label_type	varchar	255	-	NULL	The type of SGTIN, Carton
pattern_upc	varchar	255	-	NULL	UPC information
pattern_case_qty	varchar	255	-	NULL	General information field
pattern_style	varchar	255	-	NULL	General information field
pattern_color	varchar	255	-	NULL	General information field
pattern_size	varchar	255	-	NULL	General information field
pattern_string	varchar	255	-	NULL	Generation result
pattern_giai_label_type	varchar	255	-	NULL	The type of GIAI, Pallet
pattern_grai_label_type	varchar	255	-	NULL	The type of GRAI, Container
pattern_grai_asset_type	varchar	255	-	NULL	The asset type of GRAI

5. printorder: The printorder table is storing the record of each tag input.

Field name	Data type	Length	Setting	Default	Description
id	bigint	20	unsigned, Not NULL, Auto increment	NULL	Primary key
s_purchase_order_no	varchar	255	-	NULL	Batch number
s_factory_name	varchar	255	-	NULL	Sender address
s_address	varchar	255	-	NULL	Sender address
s_city	varchar	255	-	NULL	Sender address
s_state_region	varchar	255	-	NULL	Sender address
s_zip_postal_code	varchar	255	-	NULL	Sender address
s_country	varchar	255	-	NULL	Sender address
r_factory_name	varchar	255	-	NULL	Recipient address
r_address	varchar	255	-	NULL	Recipient address
r_city	varchar	255	-	NULL	Recipient address
r_state_region	varchar	255	-	NULL	Recipient address
r_zip_postal_code	varchar	255	-	NULL	Recipient address
r_country	varchar	255	-	NULL	Recipient address
r_ship_to_postal_code	varchar	255	-	NULL	Recipient address
creation_date	datetime	-	-	NULL	The creation time
total	varchar	255	-	NULL	The total number of tags
s_factory_name2	varchar	255	-	NULL	Sender address
r_factory_name2	varchar	255	-	NULL	Recipient address
s_factory_name_s	varchar	255	-	NULL	Sender address
r_factory_name_s	varchar	255	-	NULL	Recipient address
order_type	varchar	255	Not NULL	-	The internal system key

6. rfiditem: The rfiditem table is storing all information of each tag.

Field name	Data type	Length	Setting	Default	Description
id	bigint	20	unsigned, Not NULL, Auto increment	NULL	Primary key
parent	varchar	255	-	NULL	The internal system ID for association and Disassociation
epc	varchar	255	-	NULL	The EPC number of a tag
tag	varchar	255	-	NULL	The Tag number of a tag
rfid	varchar	255	-	NULL	The RFID hex number of a tag
application_identifier	varchar	255	-	NULL	Application identifier for GIAI and GRAI
barcode	varchar	255	-	NULL	The barcode of a tag
type	varchar	255	-	NULL	The type of EPC number
print_status	tinyint	1	-	0	//0=not printed; 1=printed
print_date	datetime	-	-	NULL	The date and time of printing
printorder_id	bigint	20	unsigned, Not NULL	0	Foreign key – printorder
upc	varchar	255	-	NULL	The UPC of a tag
case_qty	varchar	255	-	NULL	Tag information
style	varchar	255	-	NULL	Tag information
size	varchar	255	-	NULL	Tag information
color	varchar	255	-	NULL	Tag information
label_type	varchar	255	-	NULL	The level type //Item = item level; Carton = carton level; Pallet = pallet level; Container = container level
ci_id	bigint	20	unsigned	NULL	Foreign key – epcis_datastore
ci_date	datetime	-	-	NULL	Date of the foreign key
status	varchar	255	-	NULL	The internal system key
pattern_id	varchar	255	-	NULL	Foreign key – pattern
xml_id	bigint	20	unsigned	NULL	Reserved field
order_type	varchar	255	Not NULL	-	The internal system key
activate_status	tinyint	1	-	0	//0=not activated; 1=activated
activate_date	datetime	-	-	NULL	Date of activation

7. taghistory: The taghistory table is storing the information sent from reader.

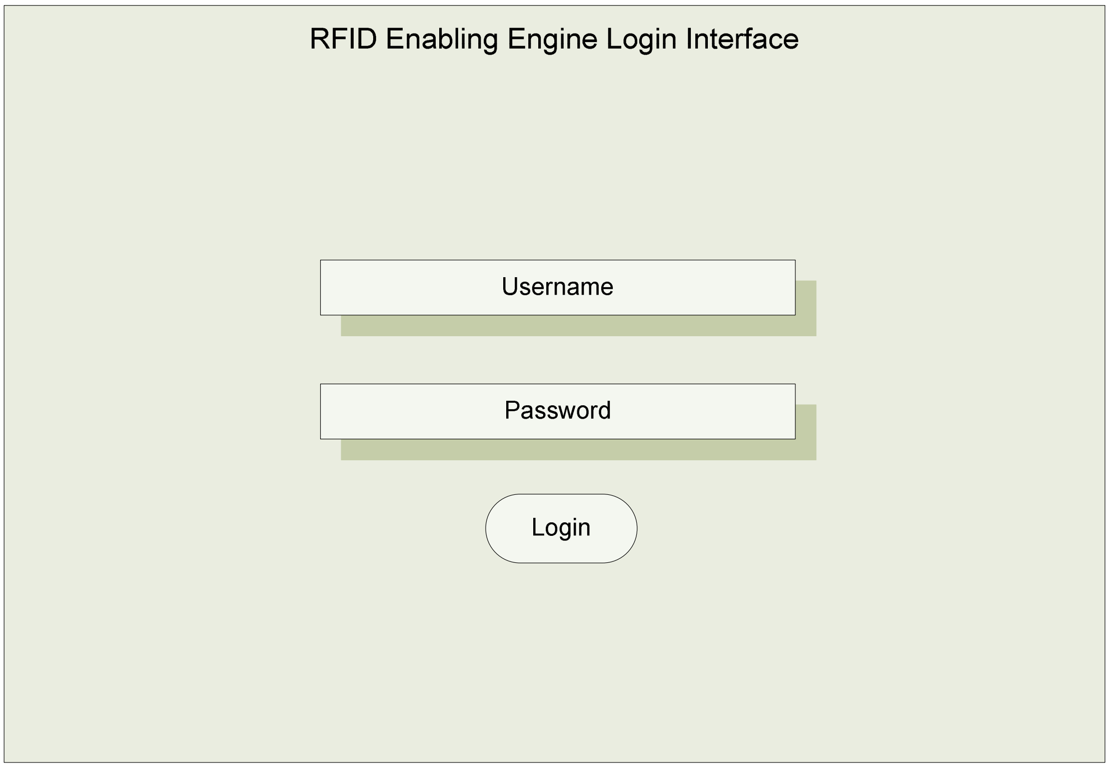
Field name	Data type	Length	Setting	Default	Description
id	bigint	20	unsigned, Not NULL, Auto increment	NULL	Primary key
parent	varchar	255	Not NULL	-	The internal system ID for association and Disassociation
epc	varchar	255	Not NULL	-	Data from reader, EPC
action	varchar	255	Not NULL	-	Data from reader, action of the event
bizStep	varchar	255	Not NULL	-	Data from reader, business step name
disposition	varchar	255	Not NULL	-	Data from reader, disposition ID
bizLocation	varchar	255	Not NULL	-	Data from reader, business location
readpoint	varchar	255	Not NULL	-	Data from reader, read point
timestamp	timestamp	-	-	'0000-00-00 00:00:00'	Timestamp of record
ci_id	bigint	20	unsigned, Not NULL	0	Foreign key – epcis_datastore
ci_date	datetime	-	-	NULL	Date of the foreign key
printorder_id	bigint	20	unsigned, Not NULL	0	Foreign key – printorder
rfiditem_id	bigint	20	unsigned, Not NULL	0	Foreign key – rfiditem

Appendix B UI Design

Base on the view model stated above, the conceptual UI Design will be defined and all major function that includes specific interface will be identified as below:

- **Login Interface**

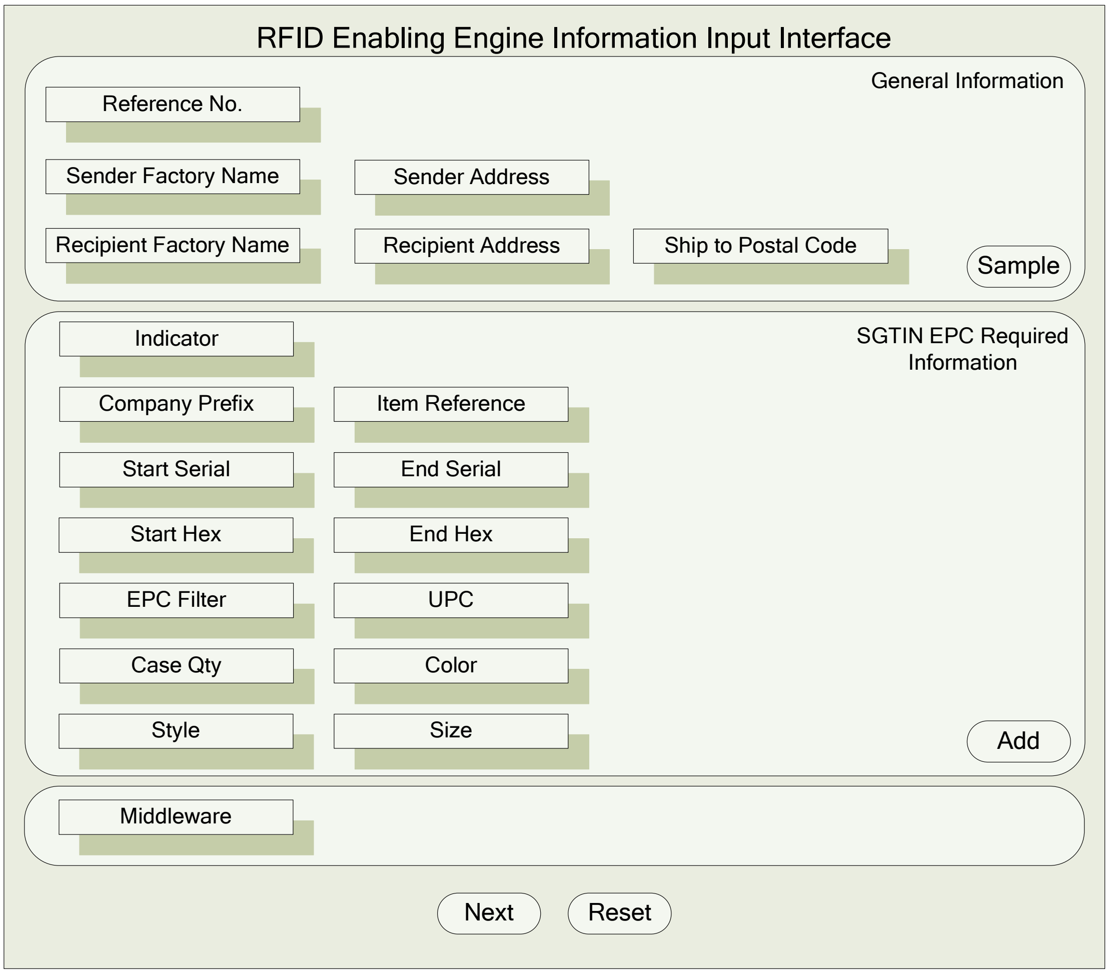
  The RFID Enabling Engine includes a login interface that requires the user to input username & password:



Field Name	Description	Validation
Username	User login ID of the RFID Enabling Engine	tomcat-users.xml under Tomcat Installation Path
Password	User password corresponding to username	tomcat-users.xml under Tomcat Installation Path
Button Name	**Description**	
Login	Login RFID Enabling Engine	

➢ **Information Input Interface (SGTIN)**

The Information Input Interface will include a UI that the user can input tag required information to generate necessary data for tag printing & validation. The UI sample of inputting SGTIN information is illustrated as below:



Field Name	Description	Validation
Reference No*	Reference No. as defined by user (PO no., internal reference, etc.)	-
Sender Factory Name	Shipper Name	-
Sender Address	Address of Shipper	-
Recipient Factory Name	Consignee Name	-
Recipient	Address of Consignee	-

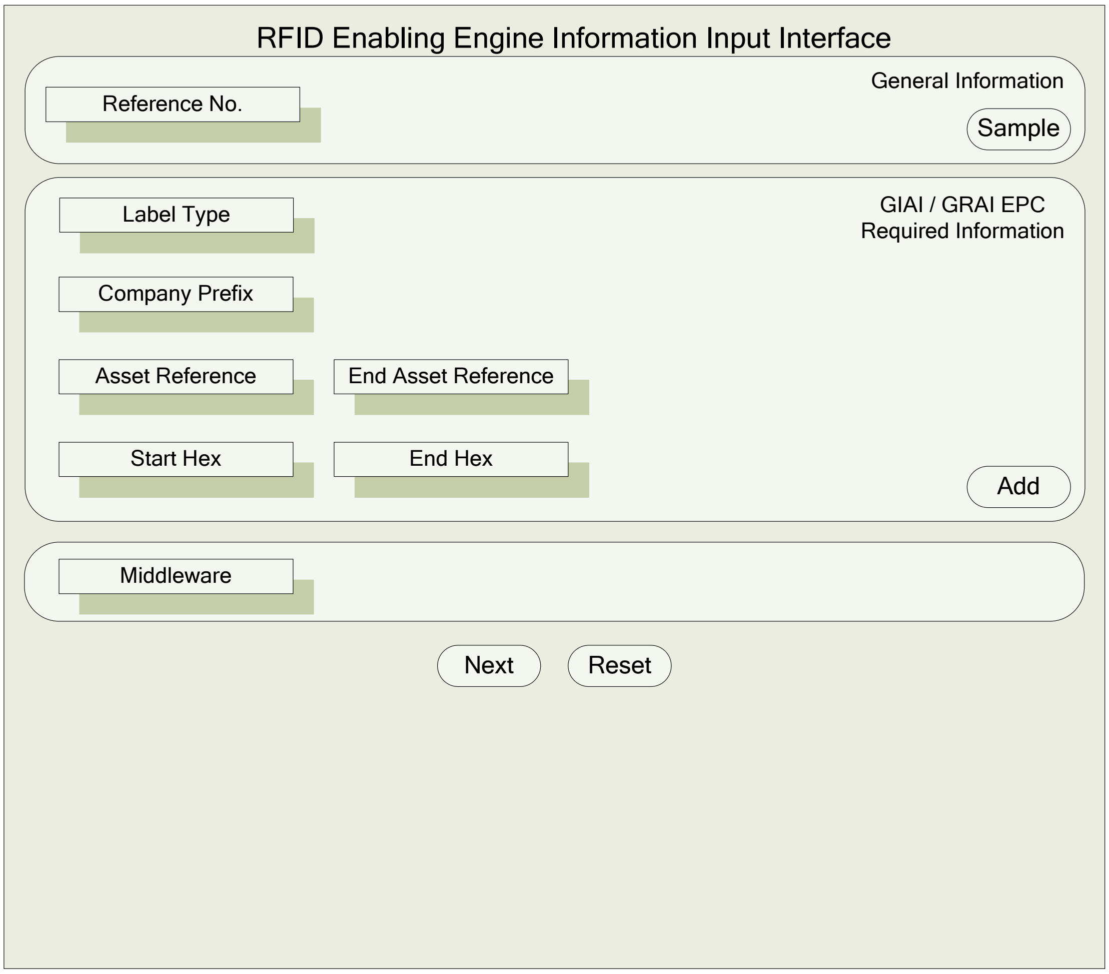
Address		
Ship to Postal Code	Ship to postal code	-
Indicator*	GTIN Indicator (select from 0 to 9)	0 to 9
Company Prefix*	Company prefix of corresponding item UPC	-
Item Reference*	Item reference of corresponding item UPC	-
Start Serial*	Starting serial no. for generating EPC serial	-
End Serial*	Ending serial no. for generating EPC serial	-
Start Hex	Starting EPC Hex for not using RFID Middleware conversion	-
End Hex	Ending EPC Hex for not using RFID Middleware conversion	-
EPC Filter*	EPC Filtering value defined by end user	0 to 3
UPC	UPC of the item	-
Case Qty	Case Qty of the item	-
Style	Style of the item	-
Color	Color of the item	-
Size	Size of the item	-
Middleware*	Select to connect RFID Middleware or not	-

*Mandatory field

Button Name	Description
Sample	Provide sample information of the input fields
Add	Add a batch of labels for this particular reference no. (order)
Next	Next to confirmation page and then save the record
Reset	Clear all the filled field before Next

➢ **Information Input Interface (GIAI / GRAI)**

The Information Input Interface will include a UI that the user can input tag required information to generate necessary data for tag printing & validation. The UI sample of inputting GIAI / GRAI information is illustrated as below:

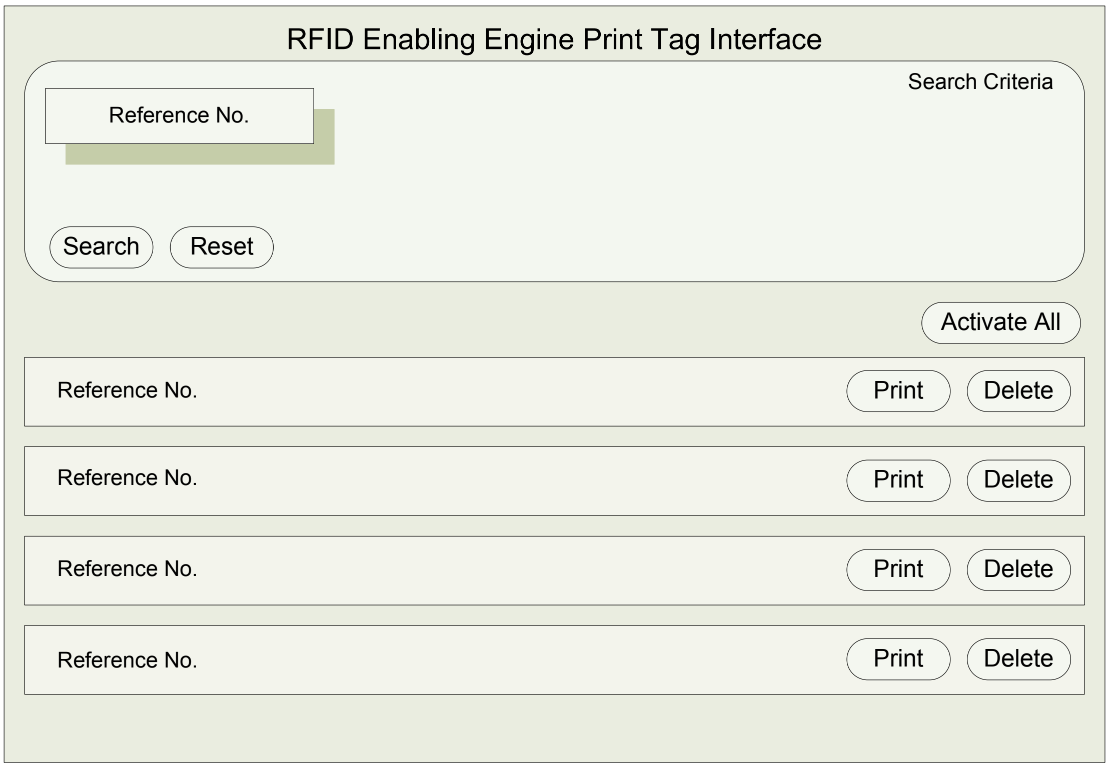


Field Name	Description	Validation
Reference No*	Reference No. as defined by user (PO no., internal reference, etc.)	-
Label Type	Select Label Type (GIAI / GRAI)	-
Company Prefix	Company prefix of corresponding asset	-
Asset Reference	The starting asset reference	-
End Asset Reference	The ending asset reference	-

Start Hex	The Start EPC Hex (Compulsory manual input for not using Middleware)	-
End Hex	The End EPC Hex (Compulsory manual input for not using Middleware)	-
Middleware	Select connecting to RFID Middleware or not	-
Button Name	**Description**	
Sample	Provide sample information of the input fields	
Add	Add a batch of labels for this particular reference no. (order)	
Next	Next to confirmation page and then save the record	
Reset	Clear all the filled field before Next	

- **Print Tag Interface**

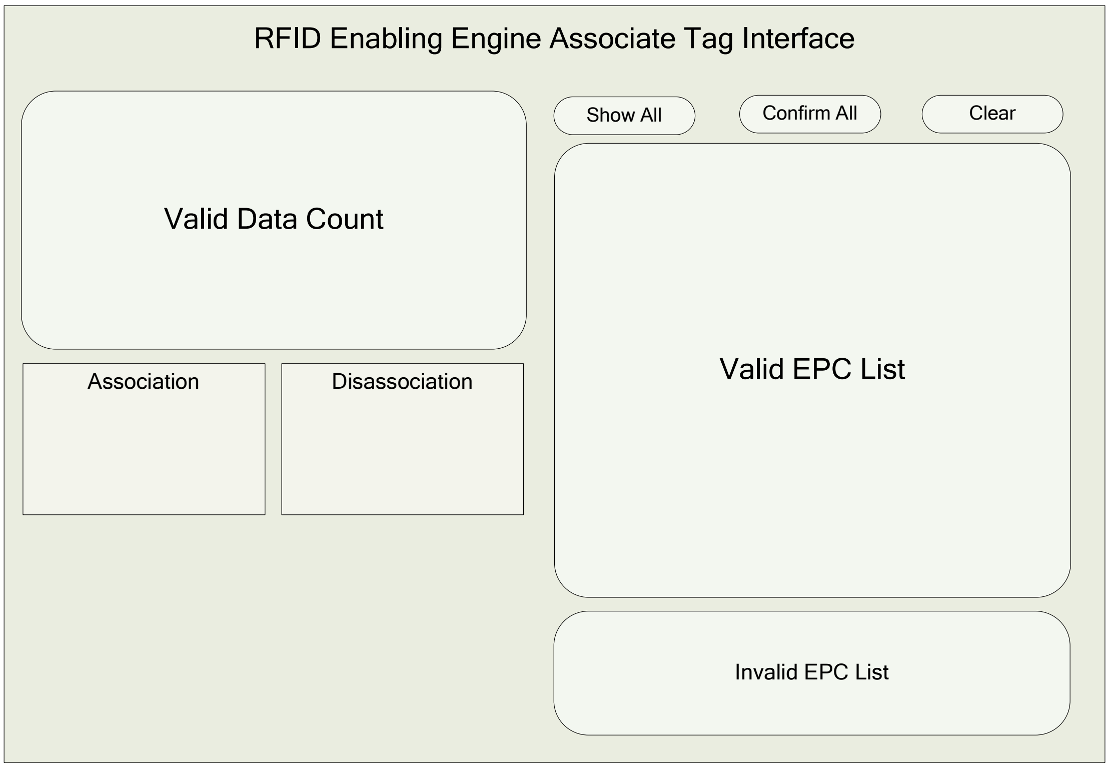
  The Print Tag Interface will include a UI that the user can input search criteria (Reference No.) to retrieve the targeted Order. The filtered result (Order List) will then be demonstrated as illustrated below (Activate tag function can be supported as backend running procedure or manual activate as below):



Field Name	Description	Validation
Reference No.	Reference No. as defined by user (PO no., etc.)	-
Button Name	**Description**	
Search	Search for Reference No.	
Reset	Reset for the input value of Reference No.	
Activate All	Manual activate all template	
Print	Selected order printing	
Delete	Selected order removal	

➢ **Associate / Disassociate Tag Interface**

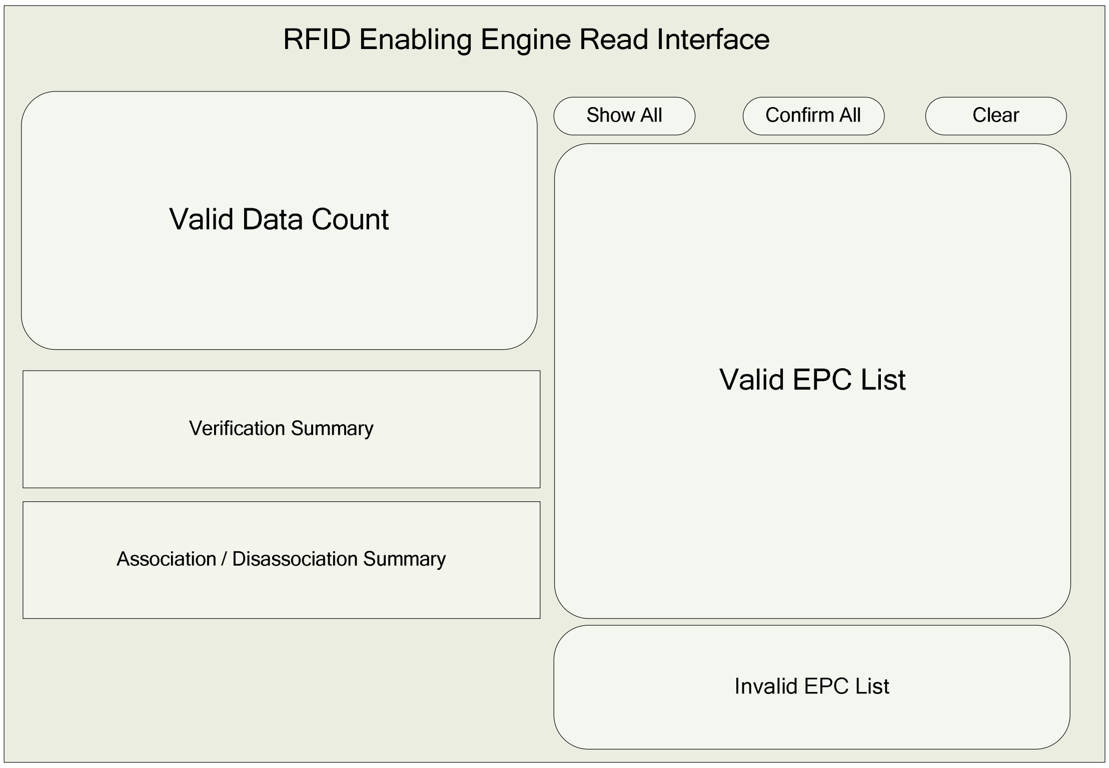
The Associate Disassociate Tag Interface will include 3 main bodies: Data Count (*upper left area)*, Associated / Disassociated EPC Summary (*lower left area),* Associated / Disassociated EPC List (*right area*). The Data Count displayed the valid no. of EPC data associated / disassociated in the RFID Enabling Engine (as captured from handheld device). The Associated / Disassociated EPC Summary illustrate the sub-total associated / disassociated EPC. Detail Parent's & Child EPC will be displayed in EPC List and pop-up window after clicking "Show All". (Verify tag function will be supported here as backend running procedure)



Field Name	Description	Validation
EPC	Parent's EPC (carton / pallet / container)	Valid in DB / Duplicate checking
EPC (Child)	Child's EPC (item / carton / pallet) of corresponding parent's EPC	Valid in DB / Duplicate checking
Type	Type of EPC (SGTIN / GIAI / GRAI)	-
Button Name	**Description**	
Show All	Show pop-up Parent EPC / Child EPC List	
Confirm All	Confirm all valid EPC association / disassociation data	
Clear	Clear EPC List	

- **Read Tag Interface**

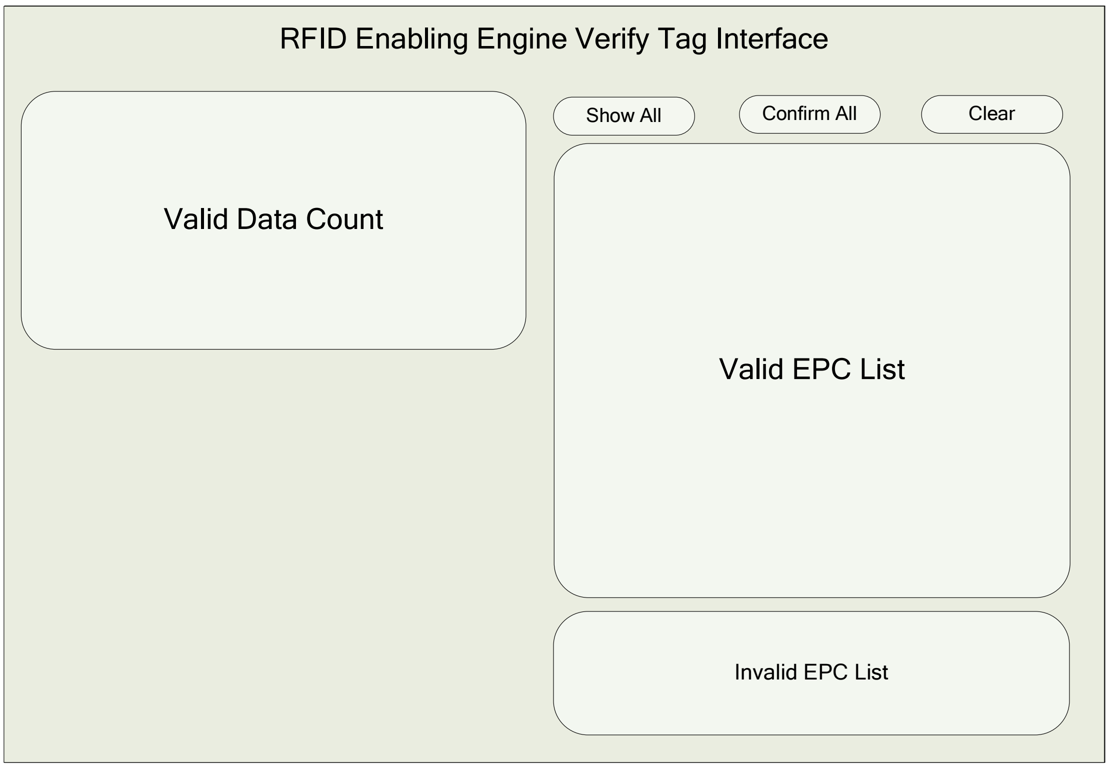
  The Read interface acts as the summary of all captured EPCIS event data from RFID handheld devices. The Verification Summary and Association / Disassociation Summary display subtotal of captured EPCIS event data that regarded as verification, association & disassociation.



Field Name	Description	Validation
EPC	EPC data with EPCIS event object	Valid in DB
Button Name	**Description**	
Show All	Show all valid EPC data	
Confirm All	Confirm all valid EPCIS event data	
Clear	Clear EPC List	

- **Verify Tag Interface**

  The tag verification function will follow the specification of verification event. Successful verified data will be display in the valid EPC List.



Field Name	Description	Validation
EPC	EPC (item / carton / pallet / container)	-
Type	Type of EPC (SGTIN / GIAI / GRAI)	-
Button Name	**Description**	
Show All	Show all valid EPC data	
Confirm All	Confirm all valid EPC data	
Clear	Clear EPC List	

Appendix C Simple Guideline on generating JAR

Some of the JAR (e.g. ALE_Schemas.jar , AleExt_xsd.jar , convert_schemas.jar, handheld_xsd.jar) in the programming source are generated by making use of Apache XMLBean. The following procedure acts as an example / simple guideline on demonstrating how the JAR has been created (The procedure is the same for the JARs with different xsd files):

- Confirm the path of "ANT" has already added to the (System variable/系統變數) call "PATH"
- Confirm the "ANT_HOME" has also already added to the (System variable/系統變數)
- Extract "xmlbeans-2.5.0.zip" to "xmlbeans-2.5.0"
- First, right click (My Computer/我的電腦) and select (Properties/內容):
- Select the (Advanced/進階) tab
- Click the (Environment variable/環境變數)
- Find the (System variable/系統變數) call "PATH"
- Click (Modify/編輯) and append the location of the "bin" directory such as "...\xmlbeans-2.5.0\bin" in (Variable Value/變數值)
- click (Confirm/確定)
- Click (Add/新增) in the bottom
- Type "XMLBEANS_HOME" in (Variable Name/變數名稱)
- Type the home directory location such as "...\xmlbeans-2.5.0" in (Variable Value/變數值)
- click (Confirm/確定)
- Go to the directory "...\xmlbeans-2.5.0\samples\XsdConfig"
- Remove the file in those folders "xml", "schemas" and "src"
- Open the "build.xml" file
- Search the key "catalog.xsd"
- Only one search result should be found
- Modify from
    - <xmlbean schema="schemas/catalog.xsd"
- To
    - <xmlbean schema="schemas"

- Copy the target xsd file to the folder "schemas" (Note: Only place 1 xsd file into the folder)
- Open Window command prompt (Type "cmd" in Startup->Run /開始->執行)

- Change the directory to the location where the "build.xml" locates
- Type "ant" in the command prompt and the success message will be shown after a moment
- After, there is a new folder "build"
- Go to the path "build/lib"
- Copy the file "schemas_xsdconfig.jar" to the target location
- Rename it to the target name

Appendix D Additional Reference Information on RFID Enabling Engine & EPCIS

The following technical information is for reference purpose:

Package: org.rfidee.web.action

Class: schedulingUpload2GS1

Package: org.rfidee.common.quartz

Class: processState

Package: org.rfidee.common.util

Class: HTTPPostSender

Package: org.rfidee.common.util

Class: SSLUtilities

Reference XML (Sample) for uploading Object Event through HTTP:

```
- <epcis:EPCISDocument creationDate="$creationDate" schemaVersion="1.0"
    xsi:schemaLocation="urn:epcglobal:epcis:xsd:1 EPCIS\EPCIS.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:epcis="urn:epcglobal:epcis:xsd:1"
    xmlns:gs1hk="urn:epcglobal:gs1hk:xsd:ext">
- <EPCISBody>
- <EventList>
- <ObjectEvent>
  <eventTime>$eventTime</eventTime>
  <eventTimeZoneOffset>+08:00</eventTimeZoneOffset>
  <epcList>$epc</epcList>
  <action>$action</action>
  <bizStep>$bizStep</bizStep>
  <disposition>$disposition</disposition>
- <bizLocation>
  <id>$bizLocation</id>
    </bizLocation>
  <gs1hk:key01>$PO#</gs1hk:key01>
  <gs1hk:key02>$Upc</gs1hk:key02>
  <gs1hk:key03>$shipToPortalCode</gs1hk:key03>
```

```
<gs1hk:numeric01>$CaseQty</gs1hk:numeric01>
  </ObjectEvent>
  </EventList>
  </EPCISBody>
  </epcis:EPCISDocument>
```

Reference XML (Sample) for uploading Aggregation Event through HTTP:

```
- <epcis:EPCISDocument creationDate="$creationDate" schemaVersion="1.0"
    xsi:schemaLocation="urn:epcglobal:epcis:xsd:1 EPCIS\EPCIS.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:epcis="urn:epcglobal:epcis:xsd:1"
    xmlns:gs1hk="urn:epcglobal:gs1hk:xsd:ext">
- <EPCISBody>
- <EventList>
- <AggregationEvent>
  <eventTime>$eventTime</eventTime>
  <eventTimeZoneOffset>+08:00</eventTimeZoneOffset>
  <parentID>$parent</parentID>
  <childEPCs>$epc</childEPCs>
  <action>$action</action>
  <bizStep>$bizStep</bizStep>
  <disposition>$disposition</disposition>
- <bizLocation>
  <id>$bizLocation</id>
    </bizLocation>
    </AggregationEvent>
    </EventList>
    </EPCISBody>
          </epcis:EPCISDocument>
```

Appendix E Additional Reference Information on RFID Enabling Engine & DATAPLEX

The following technical information is for reference purpose:

Package: org.rfidee.web.action

Class (Method): PrintDispatchActionImpl.java

Reference XML (Sample) for sending to DATAPLEX through HTTP:

```
- <Order xmlns="com.sedna.web.epcconvert">
- <EPCList>
- <SgtinItem>
   <indicator>1</indicator>
   <company_prefix>0037001</company_prefix>
   <item_reference>89012</item_reference>
   <serial_reference>123856</serial_reference>
   <serial_end_reference>123861</serial_end_reference>
   <epc_filter>3</epc_filter>
     </SgtinItem>
     </EPCList>
     </Order>
```

Reference XML (Sample) for receiving from DATAPLEX through HTTP:

```
- <Order xmlns="com.sedna.web.epcconvert">
- <EPCListResult>
- <Item>
   <barcode>10037001890122</barcode>
   <epc>urn:epc:id:sgtin:0037001.189012.123856</epc>
   <tag>urn:epc:tag:sgtin-96:3.0037001.189012.123856</tag>
   <rfid>3074024224B895000001E3D0</rfid>
     </Item>
- <Item>
   <barcode>10037001890122</barcode>
   <epc>urn:epc:id:sgtin:0037001.189012.123861</epc>
   <tag>urn:epc:tag:sgtin-96:3.0037001.189012.123861</tag>
   <rfid>3074024224B895000001E3D5</rfid>
     </Item>
     </EPCListResult>
       </Order>
```