Course Project

Java Programming

MSSE 670 Section XPS40

Ishmael L Thomas

February 28, 2013

Regis University

Table of Contents

REVISION HISTORY

Revision Number	Date	Author	Description
6	2/28/2013	Ishmael Thomas	Initial Version

INTRODUCTION AND BUSINESS CASE

This is a Java e-Pedigree application design document to track and trace drug prescriptions with pedigrees (electronic documents) throughout a pharmaceutical supply chain. Members of a pharmaceutical supply chain are manufacturers, distributors, and retailers. According to GS1, each supply chain member is responsible for producing separate pedigrees for their logistical responsibilities (shipping and receiving). In addition, each supply chain member

is responsible for completing the pedigree chain of custody. For example, the retailer would complete the chain of custody with their own pedigree and two additional pedigrees from the manufacturer and distributor. This application design document builds upon the following high priority use cases:

- **Manufacturer creates an initial pedigree**
- **Manufacturer creates a pedigree envelope for customer order**

The scope of this application design document is to develop the following four application domain classes for the high priority use cases:

- **Envelope**
- **Manufacturer**
- **Pedigree**
- **Product**

DESIGN ISSUES

In this application version, the views and controllers are coupled. This means they share a single module to deliver the Model-View-Controller architecture at the expense of the Object-Oriented Encapsulation Principle. The design benefit is that each view is independent of the next view. For example, each view can be opened, minimized, and maximized without affecting the other views. However, the design liability is that if one view closes by clicking the “x” button

located on the right corner frame of the view in focus, then all the open views will close. Views and controllers have separate modules in the next version. In addition, there is a design liability with the application exception handling mechanism. For example, the exception handling mechanism binds the presentation layer to model layer semantics. A return-status code mechanism will be incorporated in the next version.

USE CASES

This section restates the fully dressed uses cases for the e-Pedigree application design. It summarizes the use case goals (main and alternative), functionality, and use case diagrams.

Use Case 1: Manufacturer creates an initial pedigree

Main goal path:

1.1. Manufacturer generates the initial pedigree, which contains the serial number product information and item information (identifies the specific items represented by the pedigree) elements.

1.2. Manufacturer adds transaction information for the sale and signs the pedigree.

Alternative goal path:

1.1. Manufacturer generates an initial pedigree in a hard-copy format.

1.2. Manufacturer creates an electronic format using a word processing system.

1.3. Manufacturer scans and uploads either option into the e-Pedigree application when it comes available.

Functionality:

The functionality bridges requirements and features with this use case. The following depicts the features for this use case:

1.1. Create pedigree.

1.2. Add information to pedigree.

1.3. Certify (digitally sign) pedigree.

1.4. Electronically authenticate pedigrees.

1.5. Manually authenticate transactions that were not electronic.

Goal	Manufacturer creates an initial pedigree
Actor(s)	Manufacturer
Description	Initial pedigree
Frequency/ Trigger	Occurs when manufacturer is ready to ship prescription drug items to a distributer or directly to a retailer.
Pre-processing	The application has been successfully installed on the manufacturer's machine.
Post-processing	The pedigree has been successfully created with initial pedigree elements representing a manufacturer's pedigree.
Included use case	None
Extended use case	None
Technology	JAVA graphical user interface

Use Case 2: Manufacturer creates a pedigree envelope for customer order

Main goal path:

2.1 Manufacturer generates the pedigree envelope, which contains the version, serial number, date, source routing code, and destination routing code elements.

2.2 Manufacturer adds a container element for the case or tote.

2.3 Manufacturer adds one pedigree handle element for each pedigree associated with products in the case. Repeat for each case in the shipment.

2.4 Manufacturer adds each pedigree representing each physical prescription drug item in the shipment to the pedigree envelope.

Alternative goal path:

2.1 Manufacturer generates a pedigree envelope in a hard-copy format containing version, serial number, date, source routing code, and destination routing code element. In addition, the manufacturer adds container and handle elements to the hard-copy formats. Last, the manufacturer adds a pedigree (standard or alternative) representing each physical prescription drug item in the shipment to the pedigree envelope.

2.2 Manufacturer creates an electronic format using a word processing system containing – version, serial number, date, source routing code, and destination routing code element. In addition, the manufacturer adds container and handle elements to the alternative electronic formats. Last, the manufacturer adds a pedigree (standard or alternative) representing each physical prescription drug item in the shipment to the pedigree envelope.

2.3 Manufacturer scans and uploads either option into the e-Pedigree application when it comes available.

Functionality:

The functionality bridges requirements and features with this use case. The following depicts the features for this use case:

2.1 Send pedigrees for products in shipment to customer.

2.2 Verify products received against authenticated pedigrees.

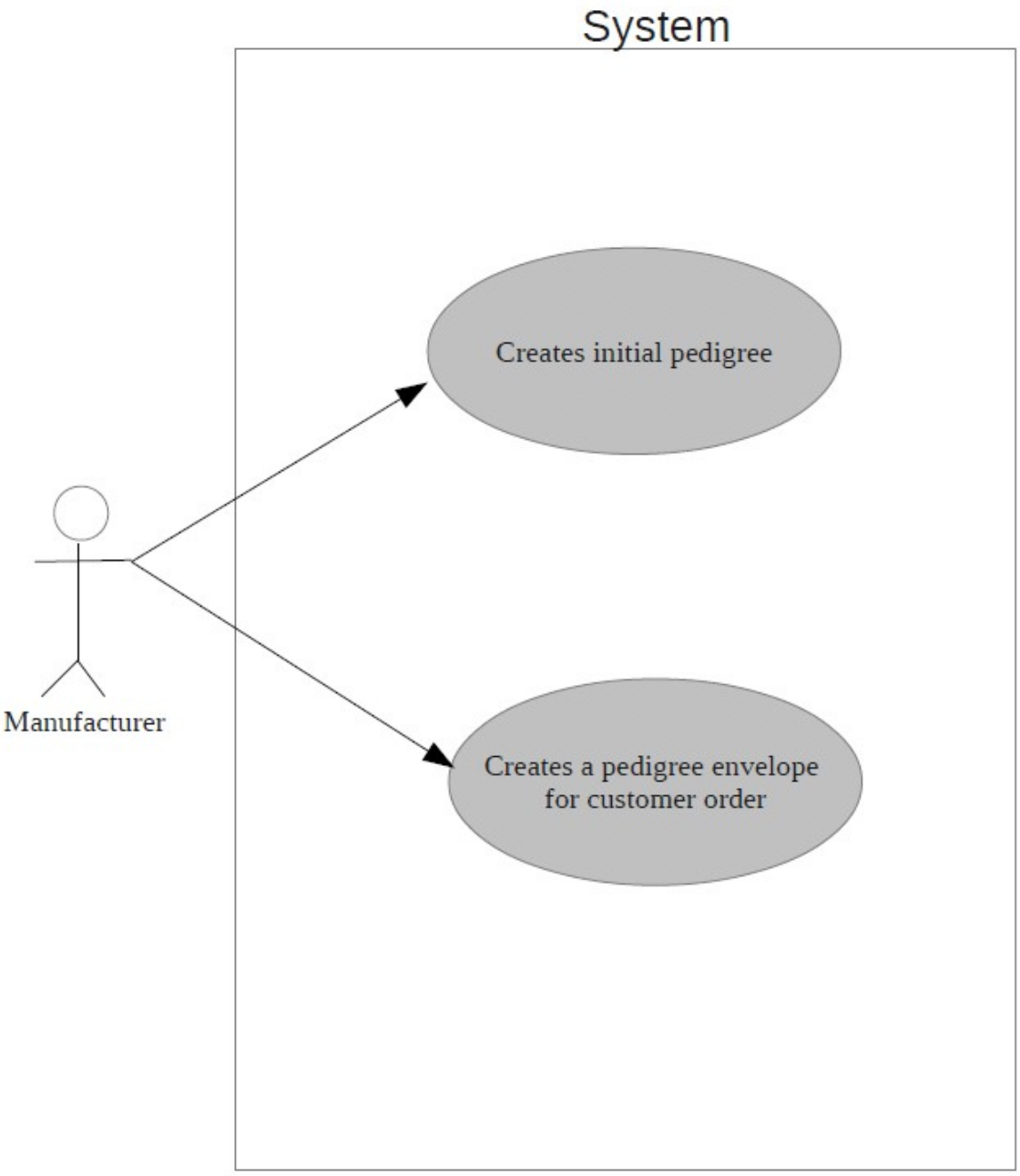
2.3 Certify (digitally sign) pedigree for receipt and authentication.

2.4 Electronically authenticate pedigrees.

2.5 Manually authenticate transactions that were not electronic.

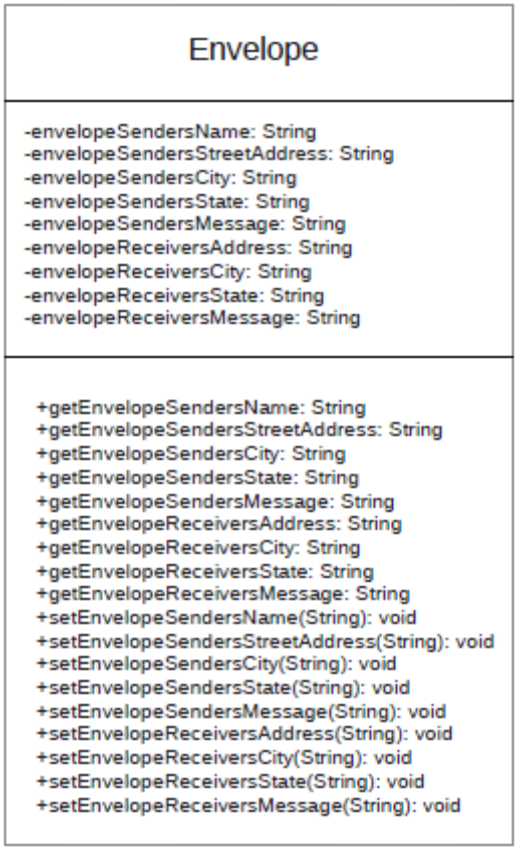
Goal	Manufacturer creates a pedigree envelope for customer order
Actor(s)	Manufacturer
Description	Pedigree envelope
Frequency/ Trigger	Occurs when the manufacturer transmits a collection of pedigrees associated with an outbound customer shipment.
Pre-processing	Successfully installation of application and creation of an initial pedigree.
Post-processing	The pedigree envelope has been successfully created and contains a copy of the manufacturer's initial pedigree.
Included use case	None
Extended use case	None
Technology	JAVA graphical user interface

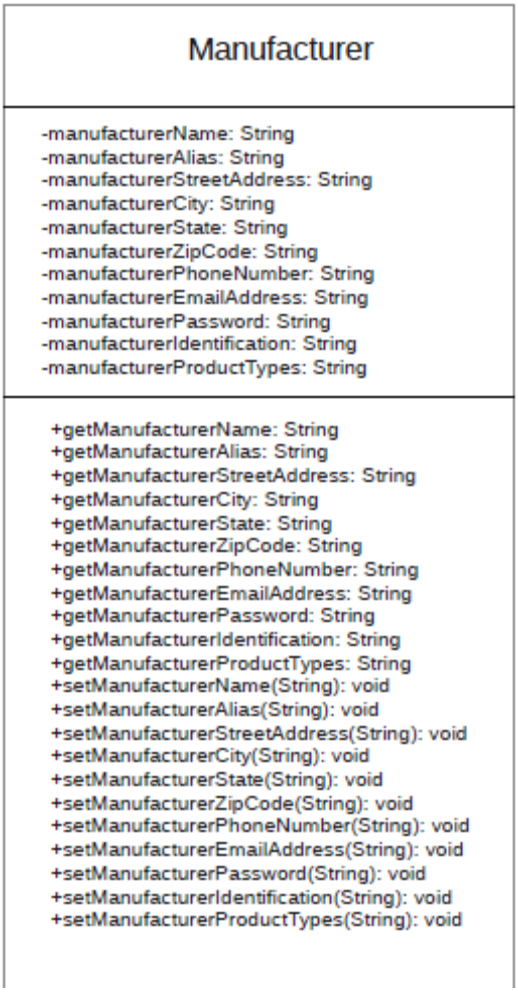
FIGURE -1 USE CASE DIAGRAMS

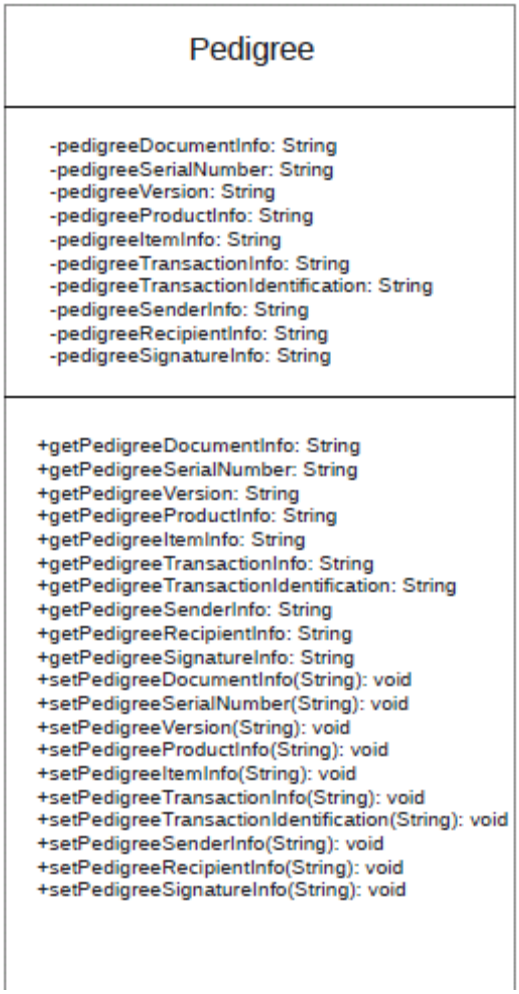


DOMAIN LAYER CLASSES

This section depicts the domain classes for the e-pedigree application.


Envelope
-envelopeSendersName: String
-envelopeSendersStreetAddress: String
-envelopeSendersCity: String
-envelopeSendersState: String
-envelopeSendersMessage: String
-envelopeReceiversAddress: String
-envelopeReceiversCity: String
-envelopeReceiversState: String
-envelopeReceiversMessage: String
+getEnvelopeSendersName: String
+getEnvelopeSendersStreetAddress: String
+getEnvelopeSendersCity: String
+getEnvelopeSendersState: String
+getEnvelopeSendersMessage: String
+getEnvelopeReceiversAddress: String
+getEnvelopeReceiversCity: String
+getEnvelopeReceiversState: String
+getEnvelopeReceiversMessage: String
+setEnvelopeSendersName(String): void
+setEnvelopeSendersStreetAddress(String): void
+setEnvelopeSendersCity(String): void
+setEnvelopeSendersState(String): void
+setEnvelopeSendersMessage(String): void
+setEnvelopeReceiversAddress(String): void
+setEnvelopeReceiversCity(String): void
+setEnvelopeReceiversState(String): void
+setEnvelopeReceiversMessage(String): void


Manufacturer
-manufacturerName: String
-manufacturerAlias: String
-manufacturerStreetAddress: String
-manufacturerCity: String
-manufacturerState: String
-manufacturerZipCode: String
-manufacturerPhoneNumber: String
-manufacturerEmailAddress: String
-manufacturerPassword: String
-manufacturerIdentification: String
-manufacturerProductTypes: String
+getManufacturerName: String
+getManufacturerAlias: String
+getManufacturerStreetAddress: String
+getManufacturerCity: String
+getManufacturerState: String
+getManufacturerZipCode: String
+getManufacturerPhoneNumber: String
+getManufacturerEmailAddress: String
+getManufacturerPassword: String
+getManufacturerIdentification: String
+getManufacturerProductTypes: String
+setManufacturerName(String): void
+setManufacturerAlias(String): void
+setManufacturerStreetAddress(String): void
+setManufacturerCity(String): void
+setManufacturerState(String): void
+setManufacturerZipCode(String): void
+setManufacturerPhoneNumber(String): void
+setManufacturerEmailAddress(String): void
+setManufacturerPassword(String): void
+setManufacturerIdentification(String): void
+setManufacturerProductTypes(String): void


Pedigree
-pedigreeDocumentInfo: String
-pedigreeSerialNumber: String
-pedigreeVersion: String
-pedigreeProductInfo: String
-pedigreeItemInfo: String
-pedigreeTransactionInfo: String
-pedigreeTransactionIdentification: String
-pedigreeSenderInfo: String
-pedigreeRecipientInfo: String
-pedigreeSignatureInfo: String
+getPedigreeDocumentInfo: String
+getPedigreeSerialNumber: String
+getPedigreeVersion: String
+getPedigreeProductInfo: String
+getPedigreeItemInfo: String
+getPedigreeTransactionInfo: String
+getPedigreeTransactionIdentification: String
+getPedigreeSenderInfo: String
+getPedigreeRecipientInfo: String
+getPedigreeSignatureInfo: String
+setPedigreeDocumentInfo(String): void
+setPedigreeSerialNumber(String): void
+setPedigreeVersion(String): void
+setPedigreeProductInfo(String): void
+setPedigreeItemInfo(String): void
+setPedigreeTransactionInfo(String): void
+setPedigreeTransactionIdentification(String): void
+setPedigreeSenderInfo(String): void
+setPedigreeRecipientInfo(String): void
+setPedigreeSignatureInfo(String): void

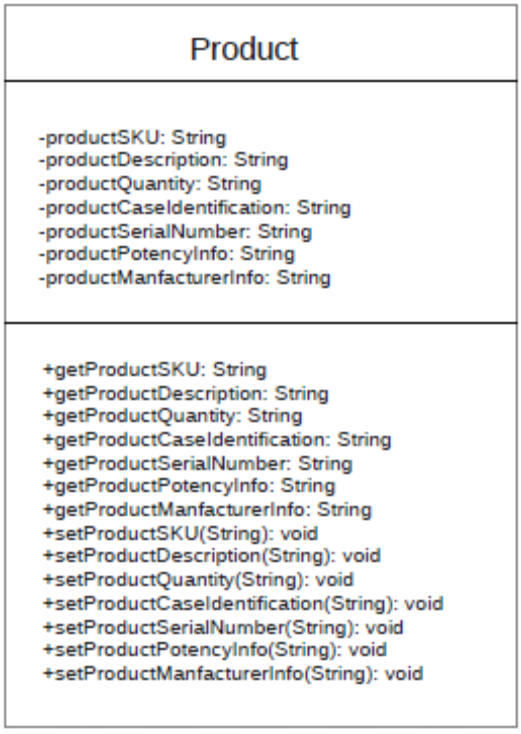
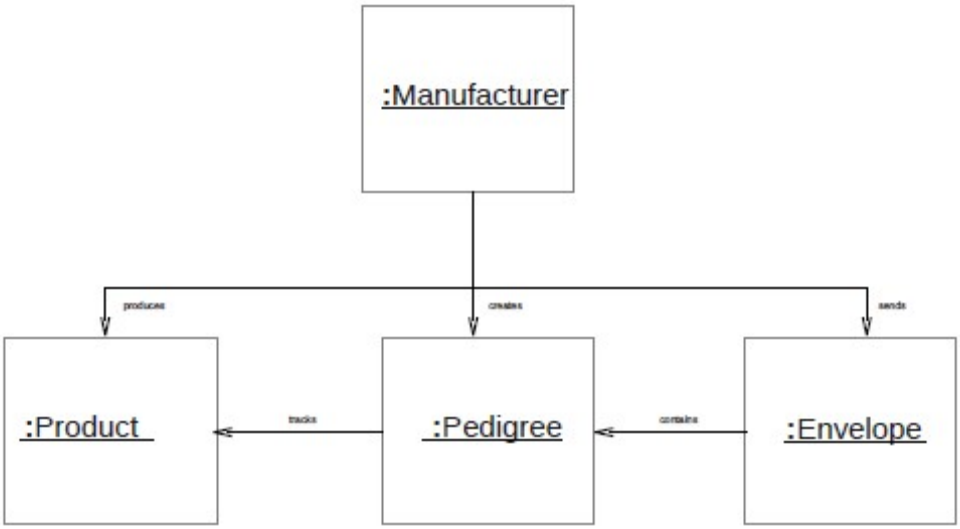
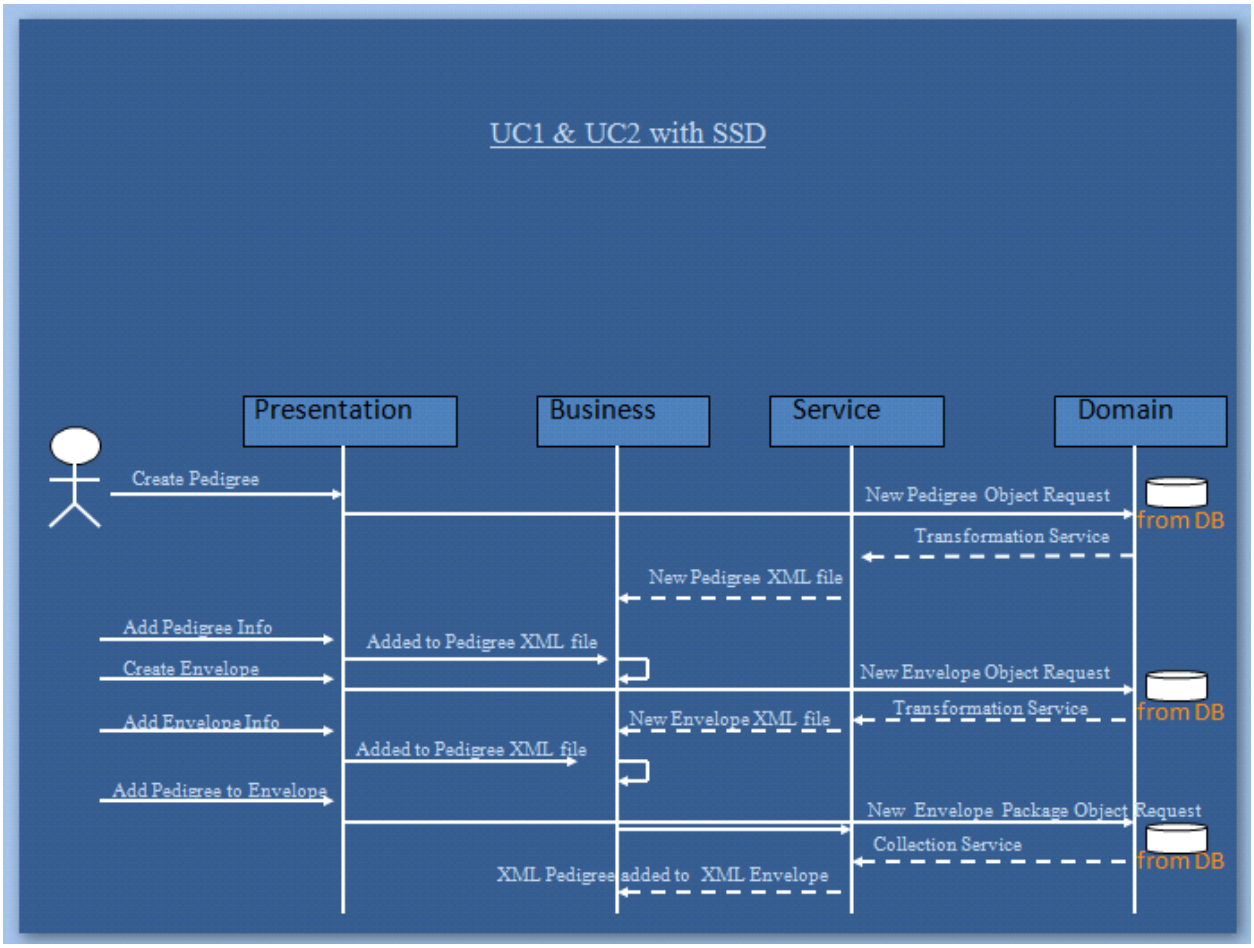


FIGURE -2 DOMAIN CLASS DIAGRAMS

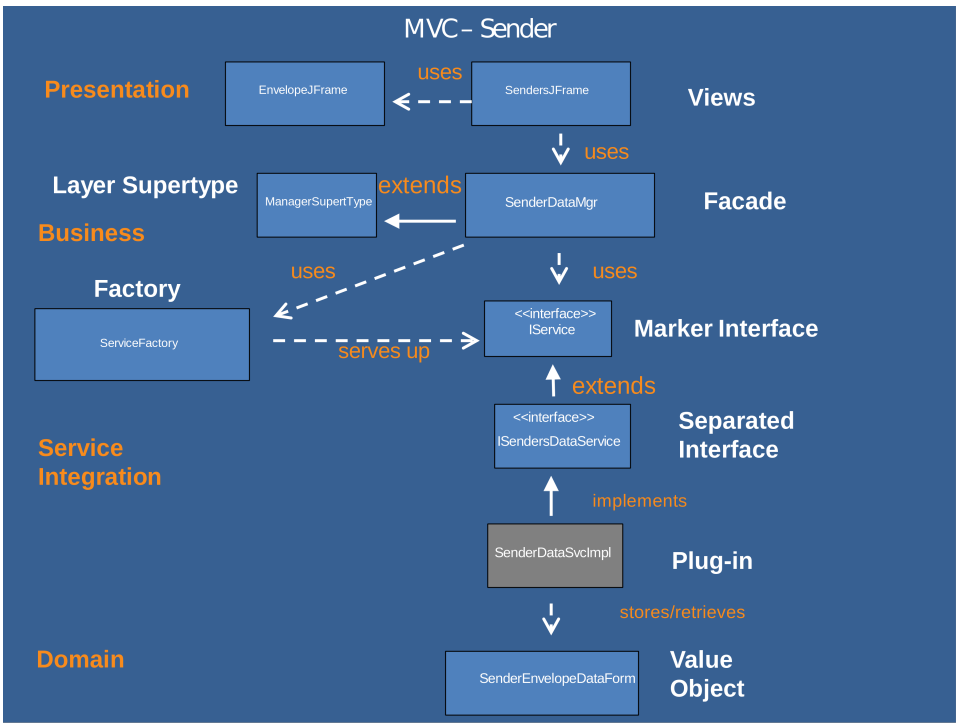


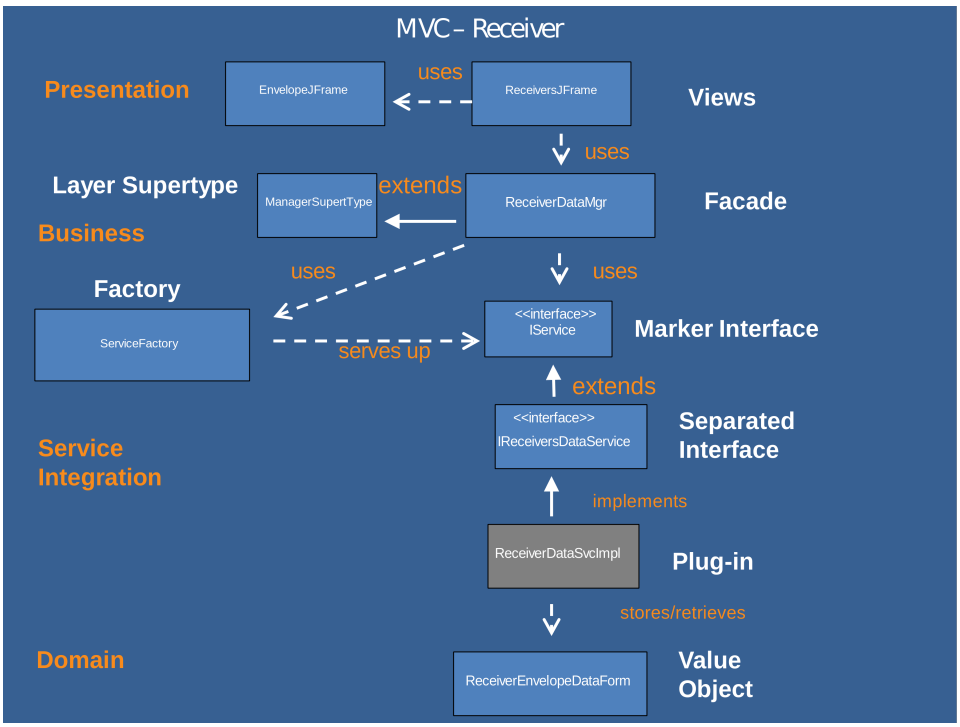
USE CASE DRIVEN SYSTEM SEQUENCE DIAGRAM

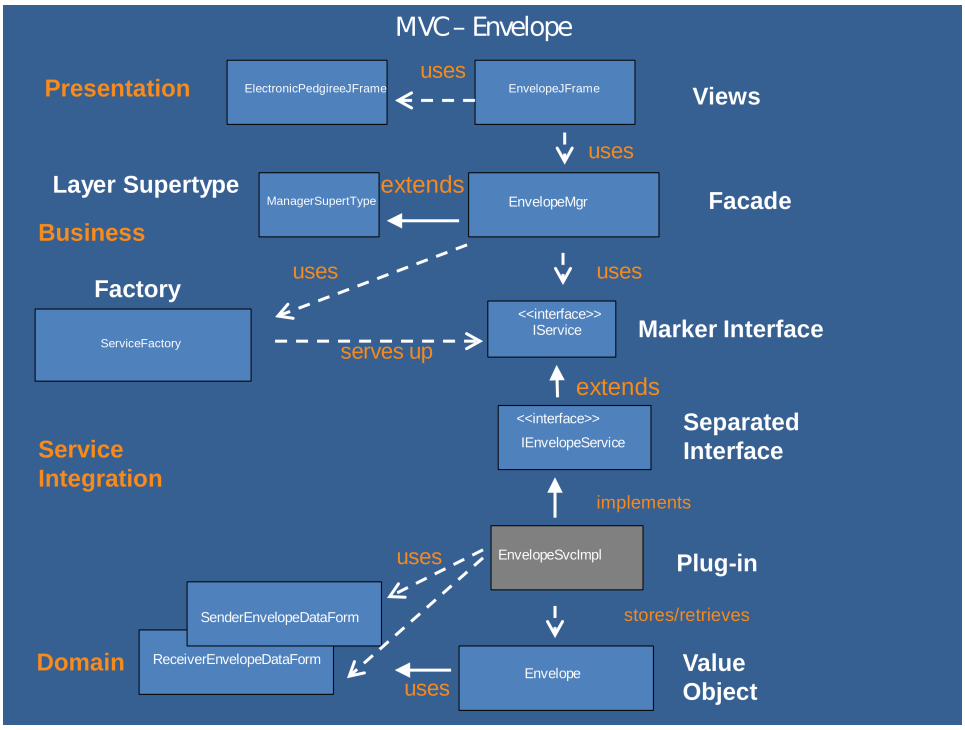


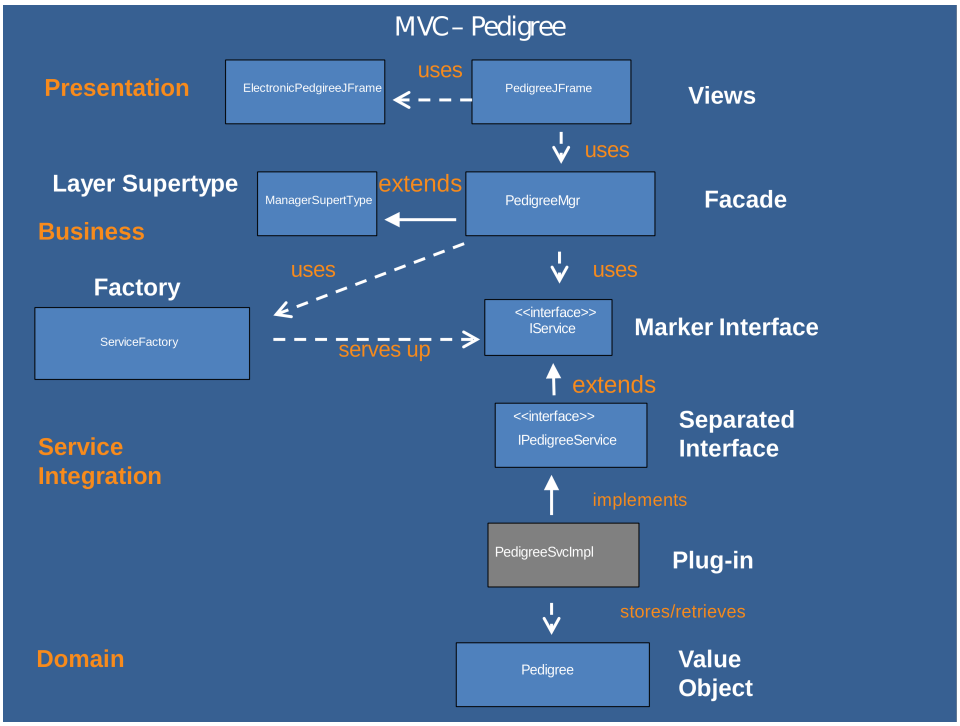
This viewpoint depicts functionality in use case one and two through the system sequence diagram (SSD). Note the object request to the domain layer. In addition, note the service requests from the domain layer to the services layer (transformation and collection). Furthermore, requests are finalized at the business layer after object conversion to XML through serialization.

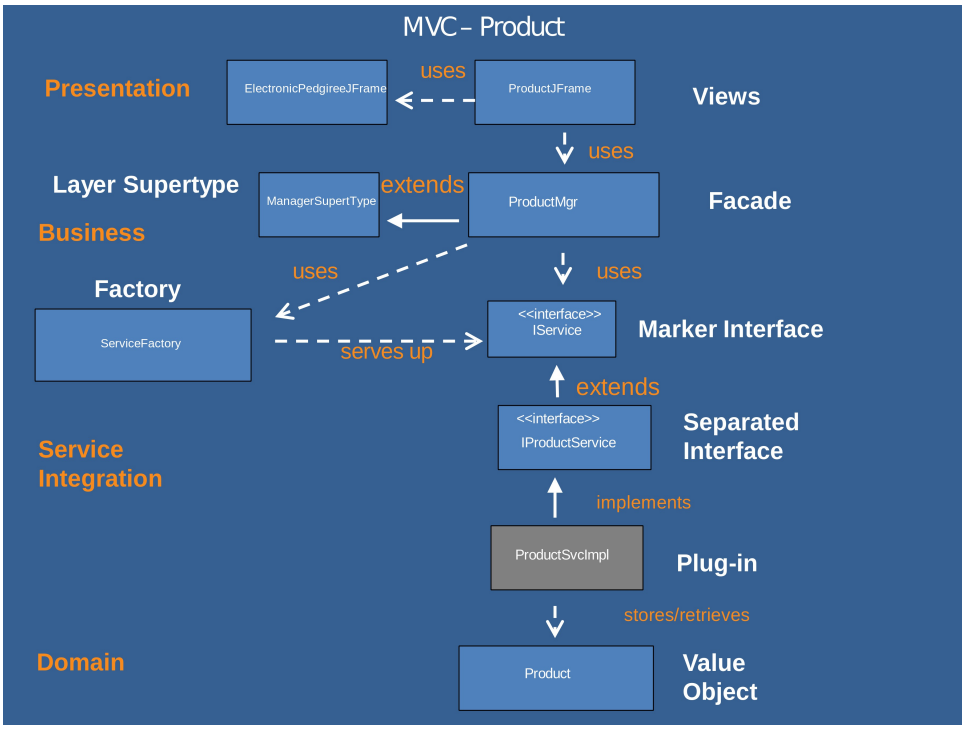
MVC ARCHITECTURE

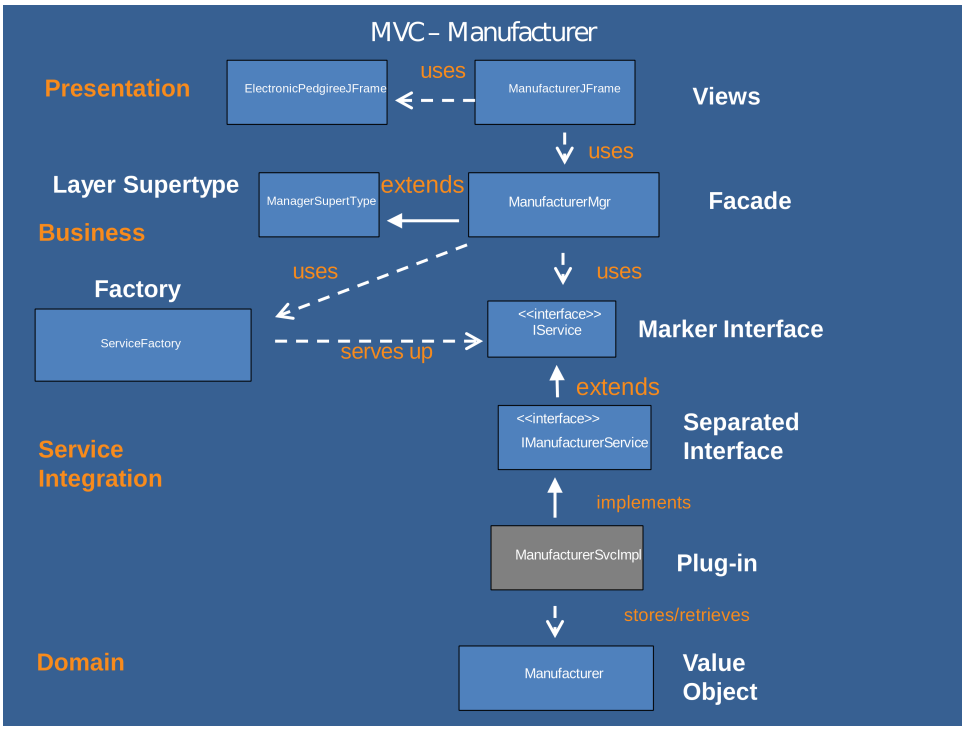





MVC – Envelope
Presentation
ElectronicPedgireeJFrame
uses
EnvelopeJFrame
Views
uses
Layer Supertype
ManagerSupertType
extends
EnvelopeMgr
Facade
Business
Factory
uses
uses
ServiceFactory
serves up
<<interface>>
IService
Marker Interface
extends
<<interface>>
IEnvelopeService
Separated
Interface
Service
Integration
implements
uses
EnvelopeSvcImpl
Plug-in
SenderEnvelopeDataForm
stores/retrieves
Domain
ReceiverEnvelopeDataForm
Envelope
Value
Object
uses


MVC – Pedigree
Presentation
ElectronicPedgireeJFrame
uses
PedigreeJFrame
Views
uses
Layer Supertype
ManagerSupertType
extends
PedigreeMgr
Facade
Business
Factory
uses
uses
ServiceFactory
serves up
<<interface>>
IService
Marker Interface
extends
<<interface>>
IPedigreeService
Separated
Interface
Service
Integration
implements
PedigreeSvcImpl
Plug-in
stores/retrieves
Domain
Pedigree
Value
Object


MVC – Product
Presentation
ElectronicPedgireeJFrame
uses
ProductJFrame
Views
uses
Layer Supertype
ManagerSupertType
extends
ProductMgr
Facade
Business
Factory
uses
uses
ServiceFactory
serves up
<<interface>>
IService
Marker Interface
extends
<<interface>>
IProductService
Separated
Interface
Service
Integration
implements
ProductSvcImpl
Plug-in
stores/retrieves
Domain
Product
Value
Object

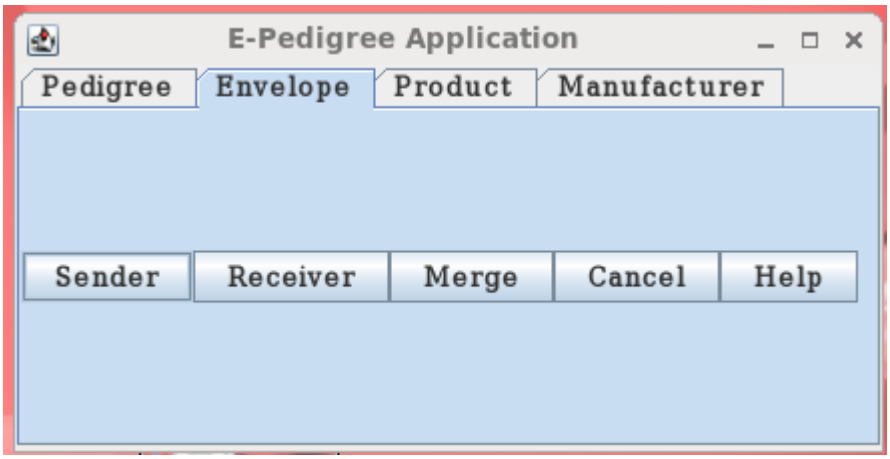

MVC – Manufacturer
Presentation
ElectronicPedgireeJFrame
uses
ManufacturerJFrame
Views
uses
Layer Supertype
ManagerSupertType
extends
ManufacturerMgr
Facade
Business
Factory
uses
uses
ServiceFactory
serves up
<<interface>>
IService
Marker Interface
extends
<<interface>>
IManufacturerService
Separated
Interface
Service
Integration
implements
ManufacturerSvcImpl
Plug-in
stores/retrieves
Domain
Manufacturer
Value
Object

GRAPHICAL USER INTERFACE

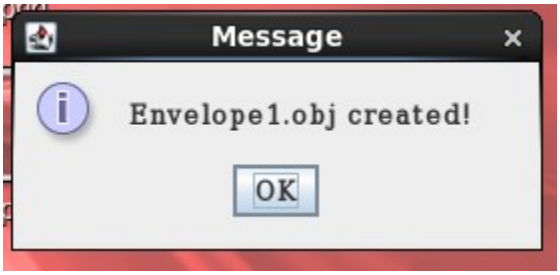
There are six views for application usability and use case execution. The six views are identified as follows:

ElectronicPedigreeJFrame

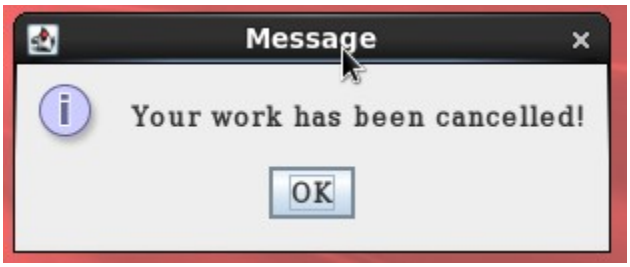
The ElectronicPedigreeJFrame is the main entry point into the e-Pedigree application. It is has five functional buttons: Sender, Receiver, Merge, Cancel, and Help. It contains two sub-frames: SenderJFrame and ReceiversJFrame. The sub-frames are dependent to the ElectronicPedigreeJFrame but are independent to each other. The Sender button launches the SendersJFrame. The Receiver button launches the ReceiversJFrame.



The Merge button doesn’t launch a view. Its function is to called upon the Envelope Manager to extract data captured within the SendersJFrame and ReceiversJFrame. Through the Envelope Manager, the Merge button consolidates sender and receiver data sources and creates a complete envelope.



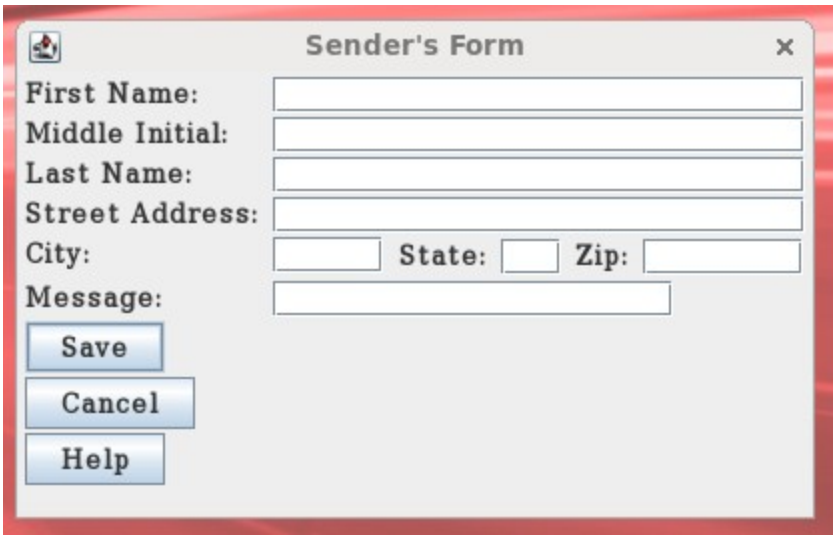
The Cancel and Help buttons provide support messages. Furthermore, they provide the same support message for each cancel and help button located in each application frame.



SendersJFrame

The SendersJFrame represents the sender's section of an envelope. It operates independently of the ReceiversJFrame. It is launched with the Sender button on the

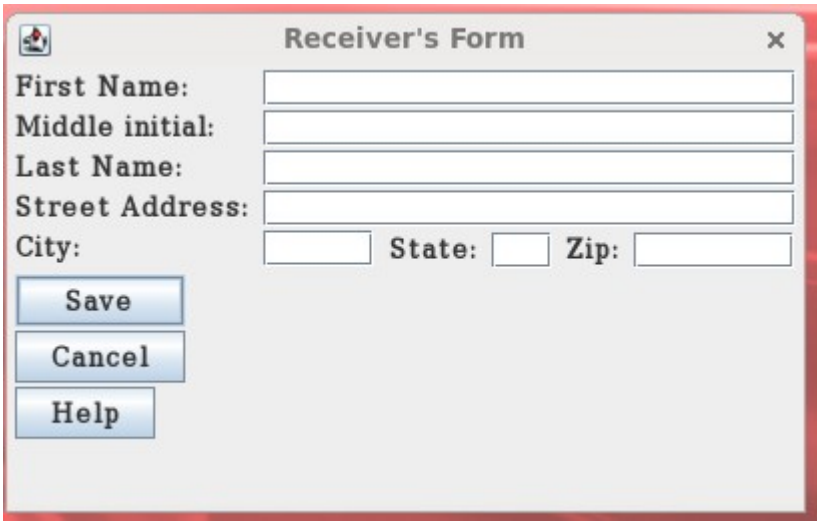
ElectronicPedigreeJFrame. The SenderJFrame calls upon the Sender Data Manager to hold sender information and to create the sender's section of an envelope. This support the realization of use case two: add new envelope.



ReceiversJFrame

The ReceiversJFrame represents the receiver's section of an envelope. It operates independently of the SendersJFrame. It is launched with the Receiver button on the

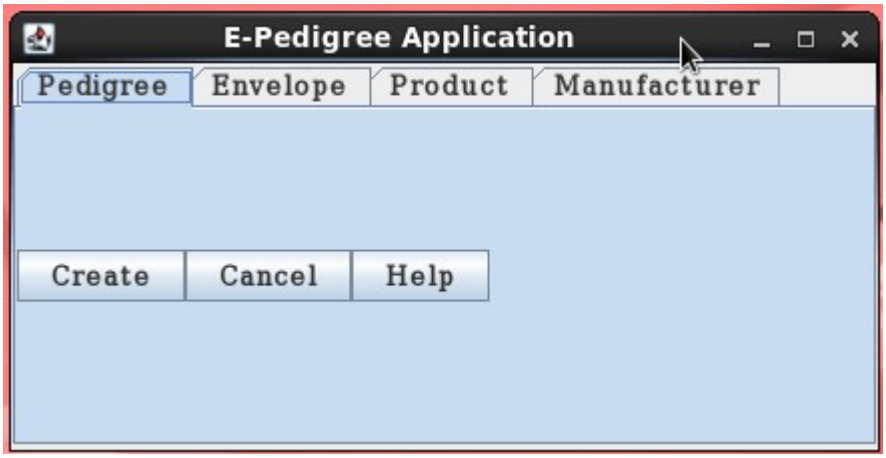
ElectronicPedigreeJFrame. It captures receiver's information and saves the information with the Save button through the Receiver Data Manager. The information is held in a receiver's form to create receiver's section of an envelope. This support the realization of use case two: add new envelope

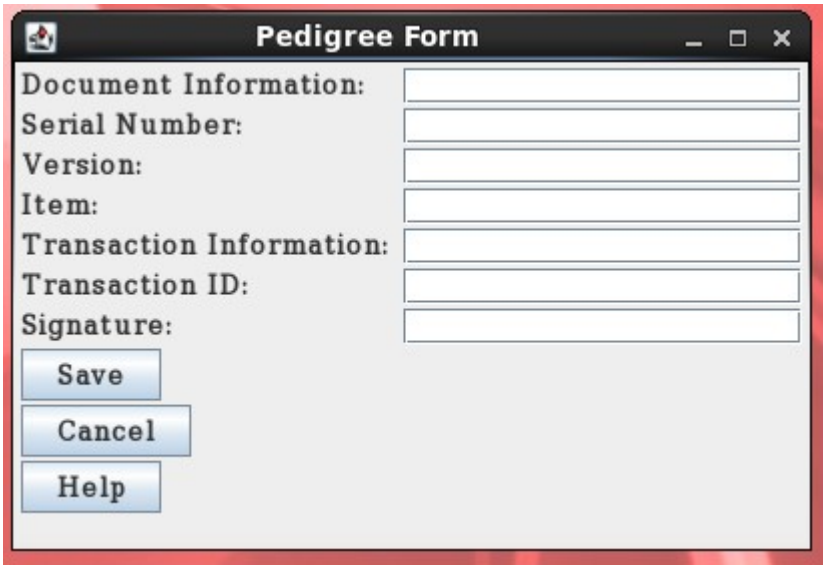


PedigreeJFrame

The PedigreeJFrame represents the pedigree form used to capture information to create a manufacturer's pedigree. It operates independently of the frames: ManufacturerJFrame,

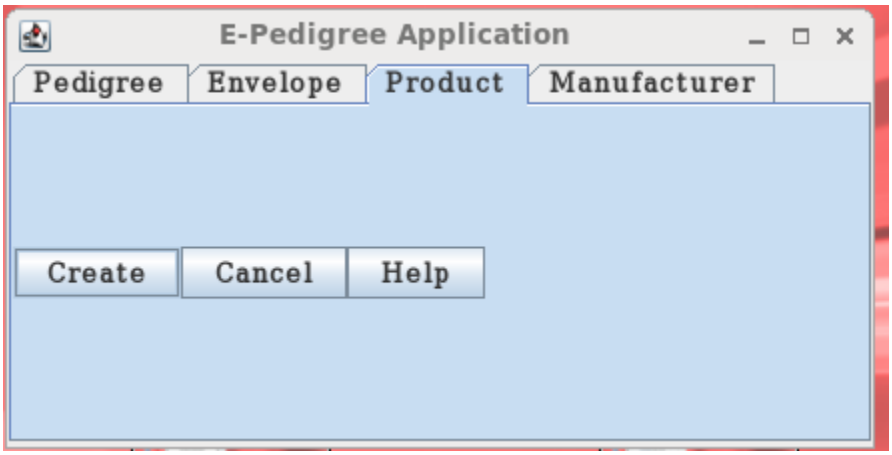
ProductJFrame, ReceiversJFrame, and SendersJFrame . It is part of the Pedigree Tab located in the ElectronicPedigreeJFrame. It is launched from the Pedigree Tab with the Create button. The information is held in a pedigree form with the Save button through the Pedigree Manager. The pedigree form is used to create a new pedigree. This support the realization of use case one: add new pedigree.

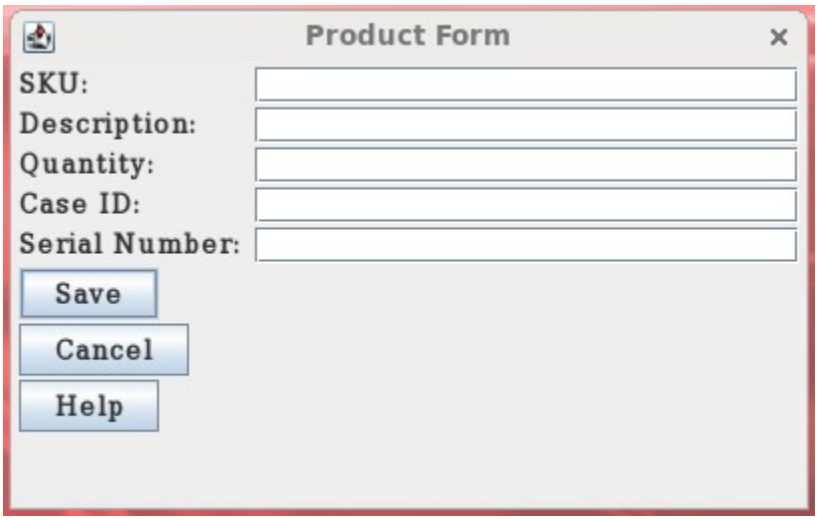




ProductJFrame

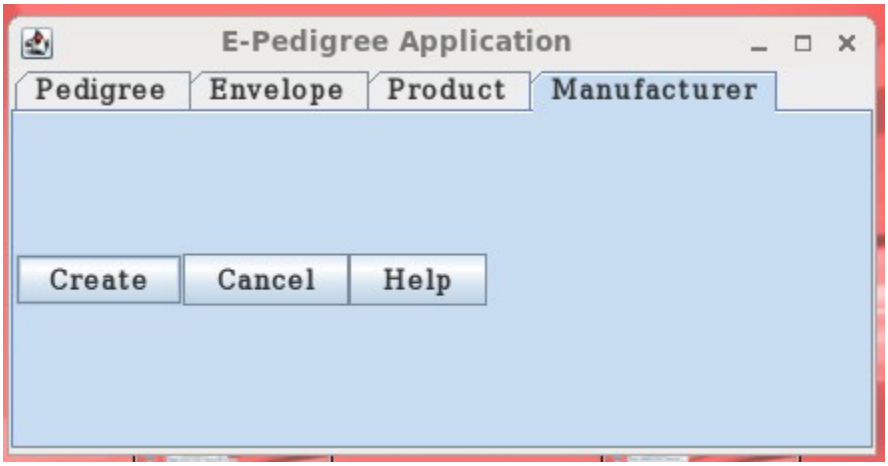
The ProductJFrame represents the product form used to capture information about a manufactured prescription drug that tracked by a manufacturer's pedigree. It operates independently of the frames: ManufacturerJFrame, PedigreeJFrame, ReceiversJFrame, and SendersJFrame . It is part of the Product Tab located in the ElectronicPedigreeJFrame. It is launched from the Product Tab with the Create button. The information is held in a product form with the Save button through the Product Manager. The product form is used to create a new product. This support the realization of use case one: add new pedigree.

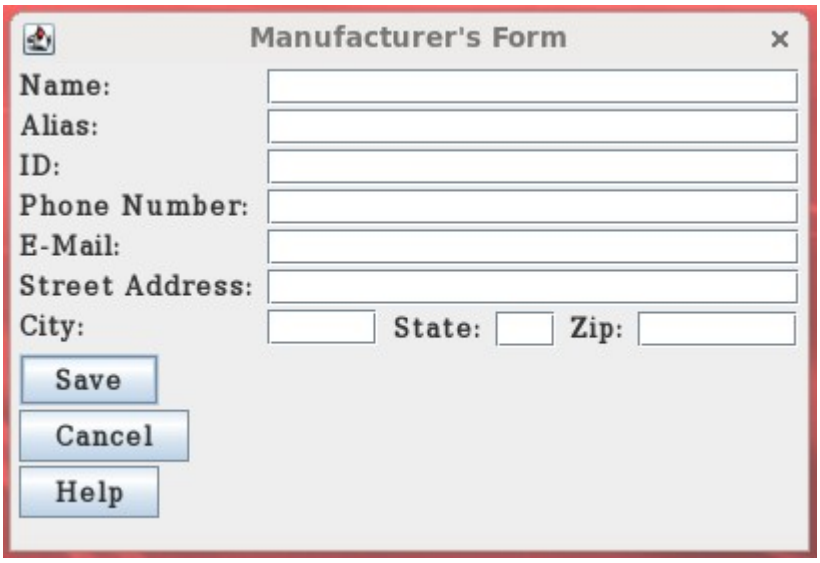




ManufacturersJFrame

The ManufacturersJFrame represents the manufacturer's profile. It used to register the manufacturer with the application and objects created during that session. It operates independently of the frames: ProductJFrame, PedigreeJFrame, ReceiversJFrame, and SendersJFrame . It is part of the Manufacturer Tab located in the ElectronicPedigreeJFrame. It is launched from the Manufacturer Tab with the Create button. The information is held in a manufacturer's profile form with the Save button through the Manufacturer Manager. The manufacturer's profile form is used to create a new manufacturer (user) to add new pedigrees and envelopes. This support the realization of use cases one and two: manufacturer add new pedigree and add new envelope.





REFERENCES

GS1 (2013). “Pedigree Standard: Pedigree Standard v1.0”. Retrieved January 11, 2013 from http://www.gs1.org/gsmp/kc/epcglobal/pedigree

GS1 (2009). “What is GS1? The importance of standards”. Retrieved January 11, 2013 from http://www.gs1.org/docs/what_is_gs1.pdf