Enrollment System (ES)

VET360

Technical Specification

**

Department of Veterans Affairs

Office of Information and Technology (OIT)

Product Development

Documentation Version 1.1

Revision History

| Date | Revision | Description | Author |
| --- | --- | --- | --- |
| 01/30/2019 | 1.1 | Address Validation | Chenjie Yu |
| 01/22/2019 | 1.0 | Initial document | Wen Lin and Trinadh Kavuri |

Table of Contents

[1 Introduction 4](#_Toc536620961)

[1.1 Purpose 4](#_Toc536620962)

[1.2 Scope 4](#_Toc536620963)

[2 Phone Number and Email Logical Deletion 5](#_Toc536620964)

[2.1 Change Request 790590 5](#_Toc536620965)

[2.2 Code Changes 5](#_Toc536620966)

[2.2.1 VET360InboundProcessService Code Changes 5](#_Toc536620967)

[3 Share ID with Vet360 7](#_Toc536620968)

[3.1 Change Request 795583 7](#_Toc536620969)

[3.2 Code Changes 7](#_Toc536620970)

[3.2.1 VET360 Model Changes 7](#_Toc536620971)

[4 ES Address Validation 10](#_Toc536620972)

[4.1 Change Request 10](#_Toc536620973)

[4.2 Code Changes 10](#_Toc536620974)

[4.2.1 User Interface (UI) Changes 10](#_Toc536620975)

[4.2.1.1 Address Validation Method 11](#_Toc536620976)

[4.2.1.2 Address Conversion From/To Model 12](#_Toc536620977)

[4.3 Model and Service Layer Changes 14](#_Toc536620978)

# Introduction

## Purpose

The purpose of this document is to discuss changes to system components and processes related to VET360 integration with the Enrollment System (ES).

## Scope

Technical specification describes the design related to VET360 integration.

# Phone Number and Email Logical Deletion

## Change Request 790590

CR 790590 business requirements describe the changes to ES to process logical deletes from Vet360 for phone number and email.

|  |  |
| --- | --- |
| **If ES receives an effective end date with an** | **Then** |
| Associated phone number from Vet 360 | ES will consider that a logical delete and move that phone number on file to history. |
| Associated email address from Vet 360 | ES will consider that a logical delete and move that email on file to history. |

## Code Changes

### VET360InboundProcessService Code Changes

If VET360 Phone entity has an effective end date, which is before the current timestamp, the phone type and phone number digits match with the on-file phone number and then deletes the phone number. Figures 1 and 2 below describes the similar changes for email object.

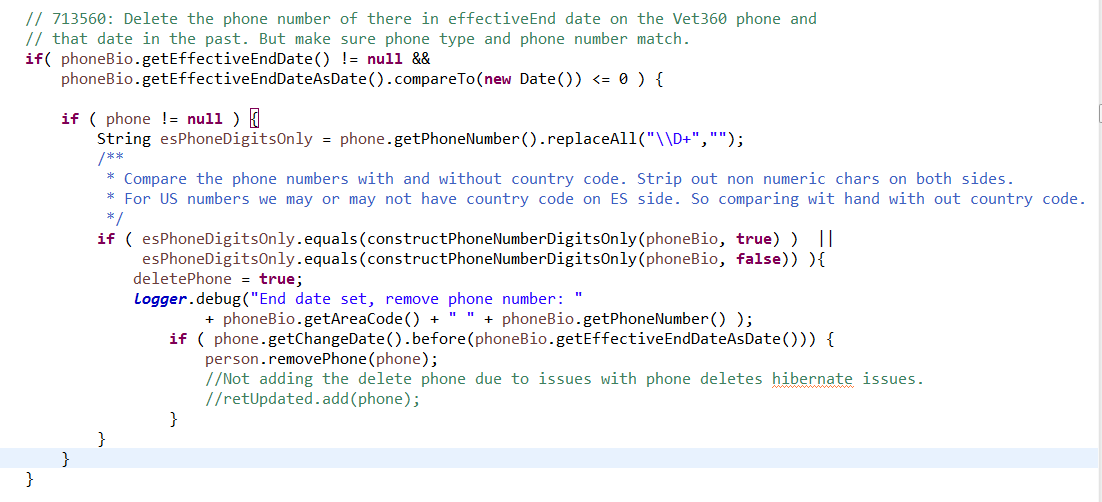


Figure 1: Code changes to VET360InboundProcessService.java updatePhones

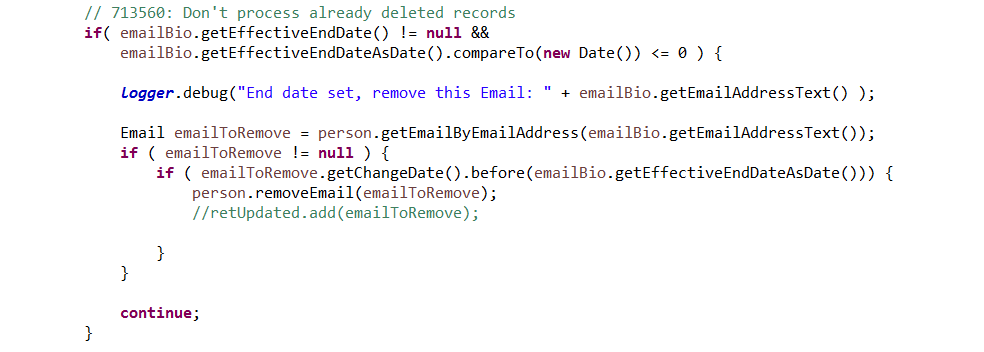


Figure 2: Code changes to VET360InboundProcessService.java updateEmails

# Share ID with Vet360

## Change Request 795583

CR 795583 business requirements describe the changes to the ES to share user ID and Site Identifier with VET360.

ES will send the user ID when a user edits/adds contact information (Address, Phone, email) as well as Site identifier to VET 360.

Table 1: VET360 Changes by Interface

|  |  |  |
| --- | --- | --- |
| **Interface** | **Current** | **To be** |
| **HCA** | originatingSourceSystem : HEC  sourceSystemUser" : anonymousUser | originatingSourceSystem" : HEC  sourceSystemUser" : **VOA event** |
| **VistA** | originatingSourceSystem" : VAMC  sourceSystemUser" : anonymousUser | originatingSourceSystem" : VAMC  ourceSystemUser" : **VAMC-608** |
| **ES UI** | originatingSourceSystem : HEC  sourceSystemUser" :  DNS | originatingSourceSystem : HEC  sourceSystemUser" :  DNS |

## Code Changes

### VET360 Model Changes

Previously, ES was sending the sourceSystemUser as anonymous when the object is modified by HCA or VistA. In the database, the record “modified by” will be either a “VOA Event” or “VAMV site code”. Because these are not real user accounts, developers changed the API to get the logical Id as shown in Figures 3-6 below.

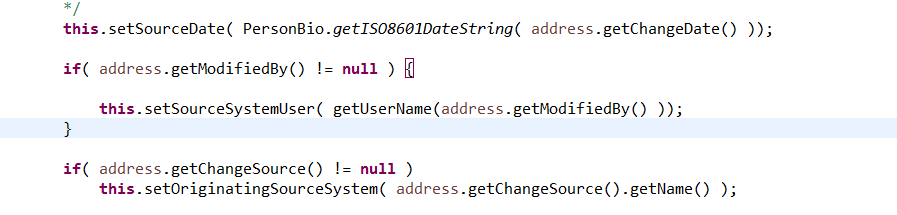


Figure 3: Code changes to AddresBio.java

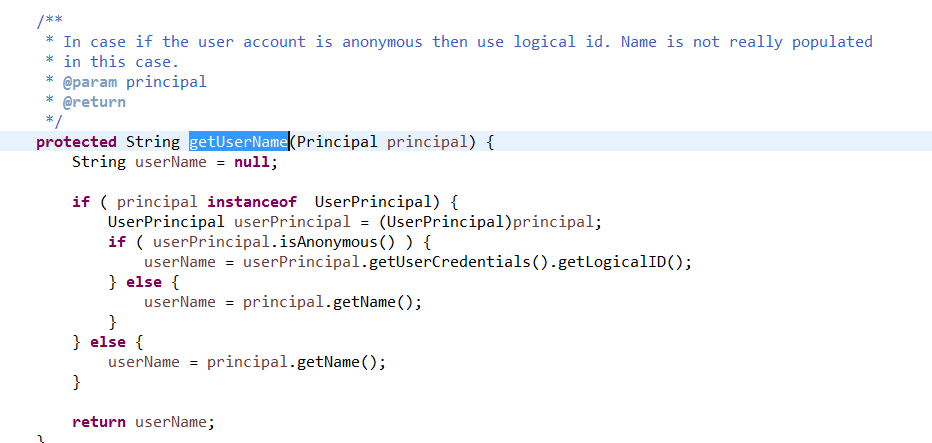


Figure 4: Code changes to BioBase.java

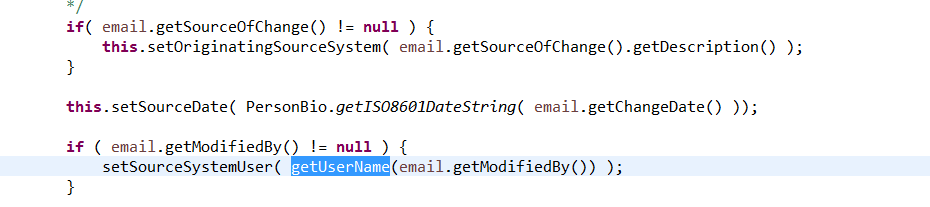


Figure 5: Code changes to EmailBio.java

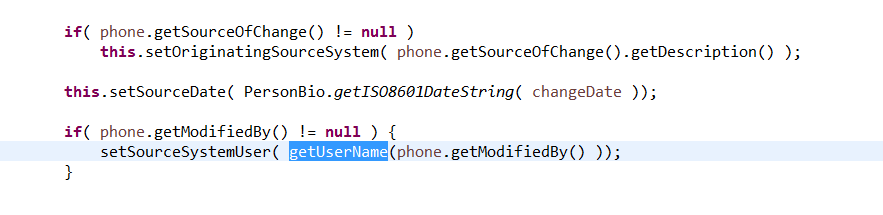


Figure 6: Code changes to PhoneBio.java

# ES Address Validation

## Change Request

User story 788100 Real Time Validation of Addresses. When an ES user enters address information, the ES provides a display that allows real-time validation of addresses.

## Code Changes

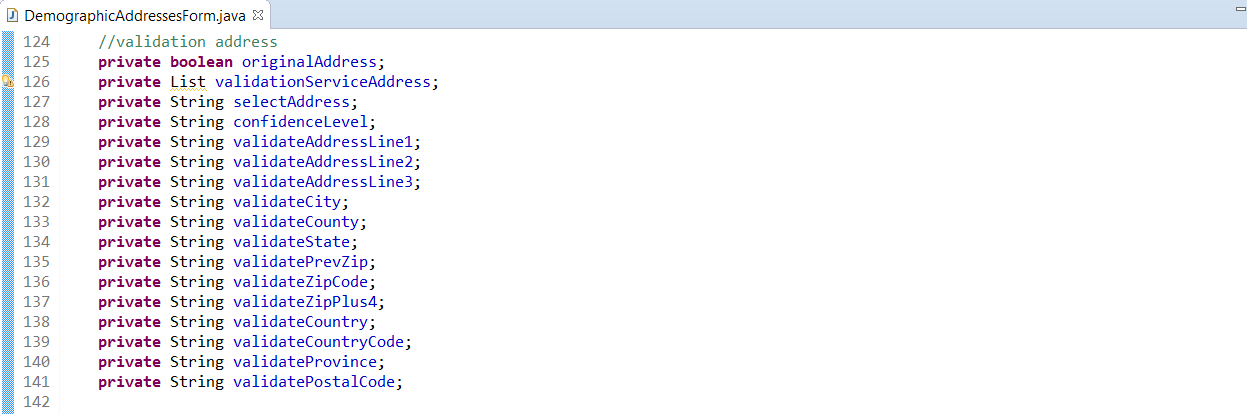
### User Interface (UI) Changes

Changes to the ES user interface (UI) on the Demographics, Addresses screen include adding new fields for real-time address validation. The new fields include three parts:

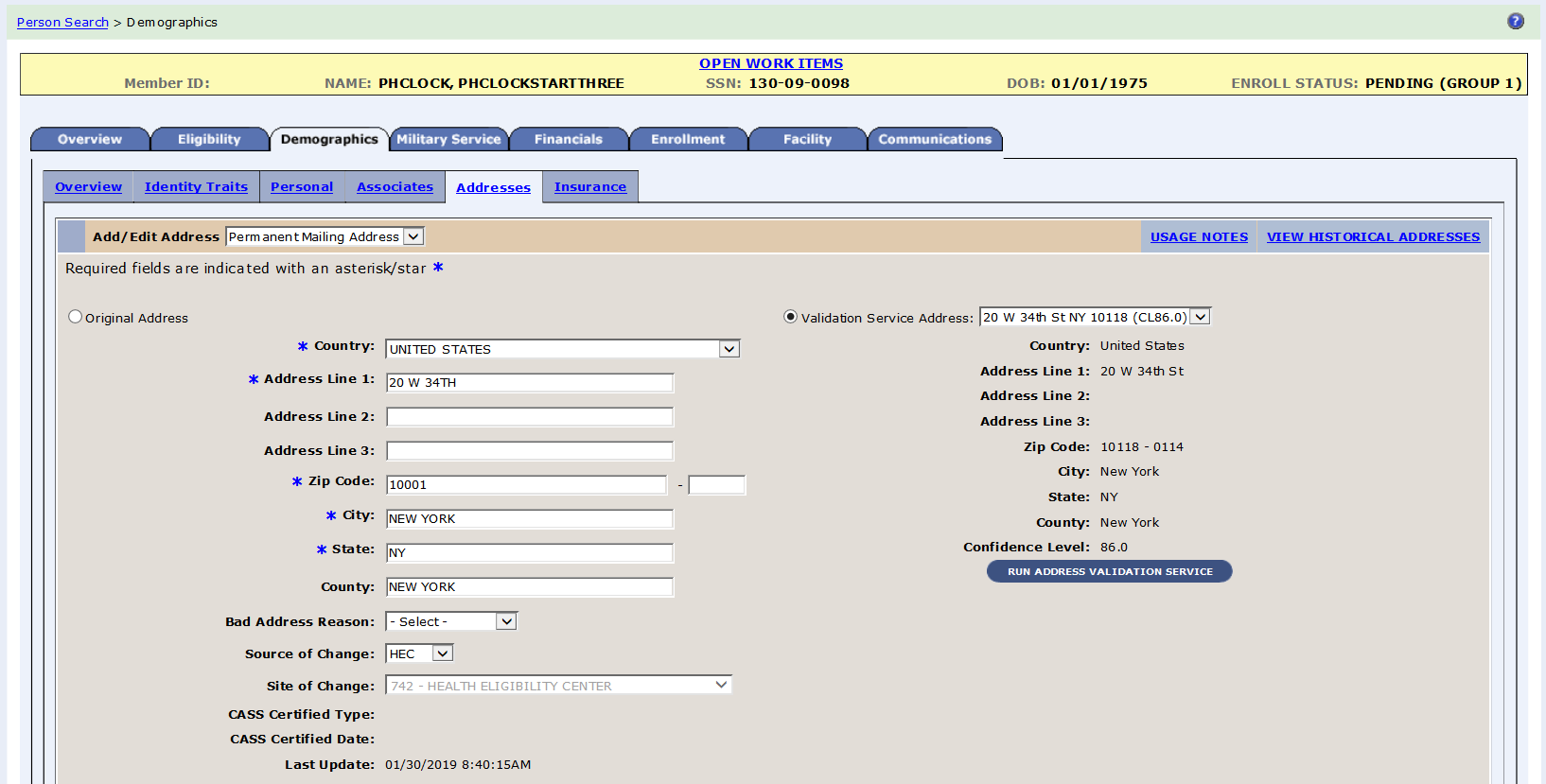
1. Two radio buttons for the ES user to select the address to update and a selection item for candidate addresses from the validation service response.
2. Validated address details display on the right-side of original address.
3. A Run Address Validation button displays under the validated address.

UI changes update the DemographicMessages.properties, demograhicAddAddress.jsp and DemographicAddressesForm.java files.

**File:** DemographicAddressesForm.java



**Figure 7: Validated Address Variables**

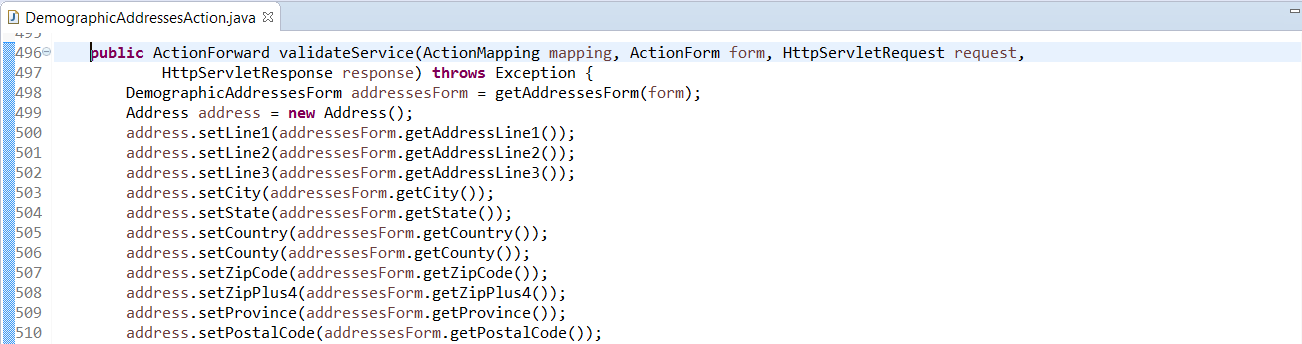
****

**Figure 8: Address Validation UI**

#### Address Validation Method

The Address Validation Function adds the new action method validateService() into DemographicAddressesAction.java file. This method sends the address data from the original address to call the validation server. After it receives the CandidateAddressResponse data, it will store the data into the new validated address fields in DemographicAddressesForm.

**File:** DemographicAddressesAction.java



**Figure 9: validateService() method**

#### Address Conversion From/To Model

Two changes were made in the DemographicAddressesConversionService.java file:

* Add a new method convertValidateAddressToAddress() into DemographicAddressesConversionService.java file.
* Update the convertFormToPerson() method.

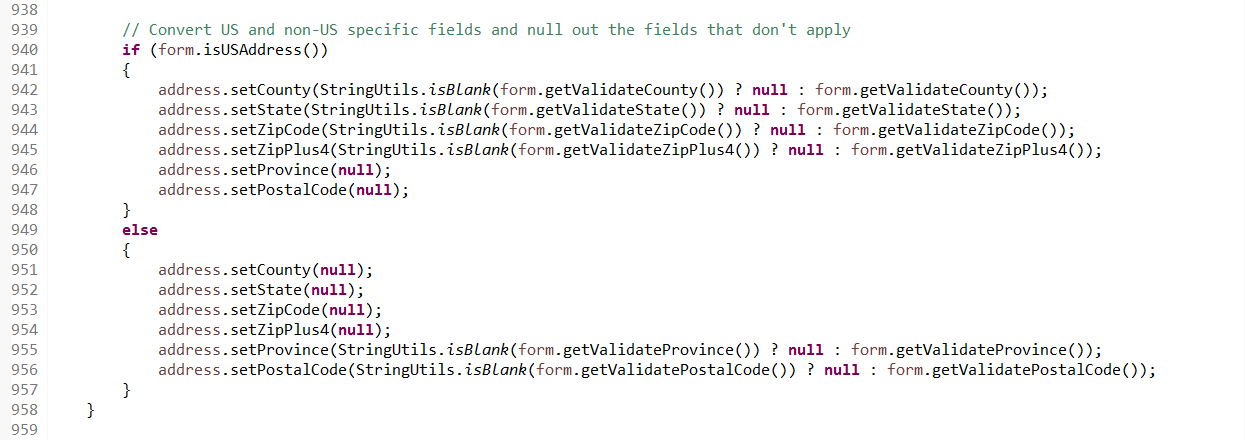
Perform the following steps below to store a validated address:

1. Check the value of the new Boolean variable isOriginalAddress().
2. Run the new convertValidateAddressToAddress() method to replace the original convertFormToAddress() method if it is false.
3. convertValidateAddressToAddress() method converts the validated address data in DemographicAddressesForm to an Address object.

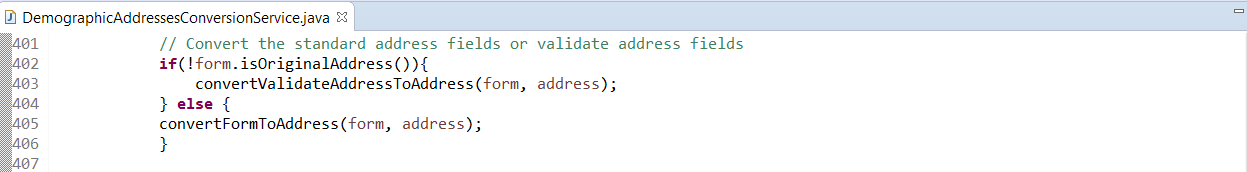
|  |  |
| --- | --- |
| **If** | **Then** |
| True | The original standard address will be convert to the Address object |
| False | The new convertValidateAddressToAddress() method will be used to convert the validated address to the Address Object. |

**File:** DemographicAddressesConversionService.java

****

****

**Figure 10: convertValidateAddressToAddress() method**

****

**Figure 11: Store Validated Address**

## Model and Service Layer Changes

TheVET360 Universal Addressing Module (UAM) service provides address validation service. The UAM service receives the request, which includes the original address, as well as sends the response that includes candidate addresses. In ES, new AddressValidationService.java and AddressValidationServiceImpl.java files were added for the address validation service. Furthermore, candidate address related class files were added for the objects using the request and response. Model classes are generated using Swagger, code-gen tool.

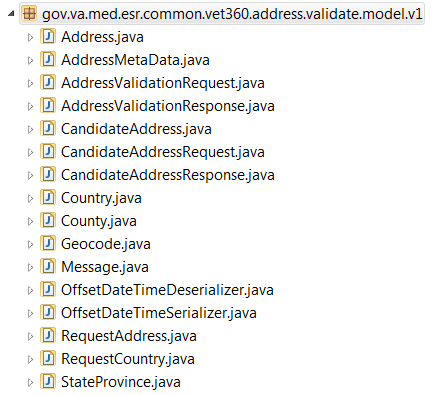
****

Figure 12: candidate address related class

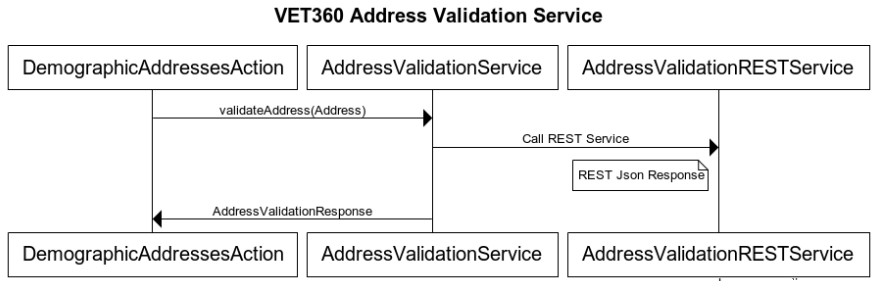
****

Figure 13: Vet360 Validation Service Sequence Diagram

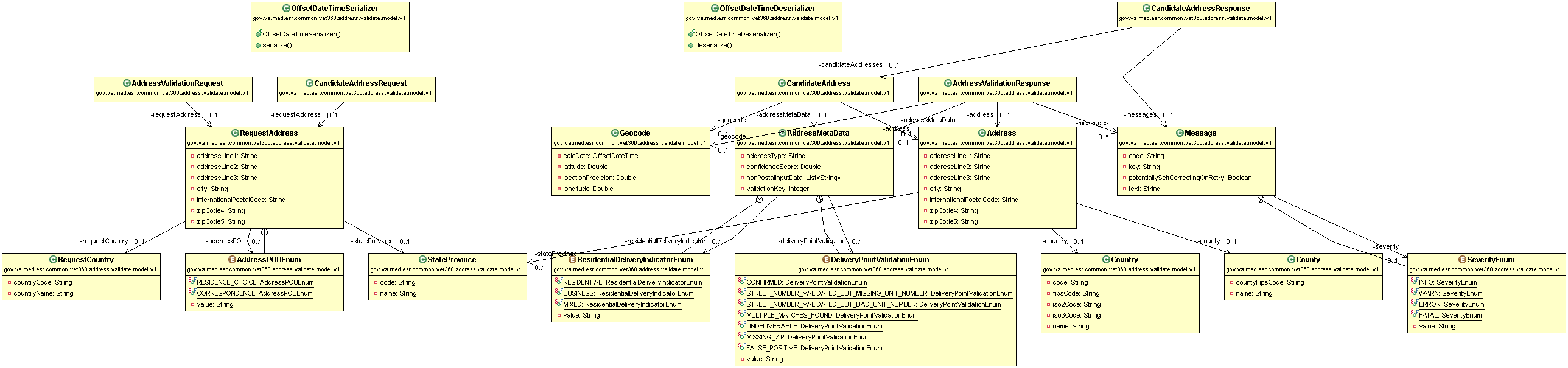
****

Figure 14: VET360 Class