

CMS SOAP CLIENT

SOFTWARE REQUIREMENTS
SPECIFICATION

CONTENTS

1. [Introduction](#)
 - 1.1. Purpose
 - 1.2. Scope Of Project
 - 1.3. Glossary
 - 1.4. References
 - 1.5. Overview Of Document
2. [Overall Description](#)
 - 2.1. [System Environment](#)
 - 2.2. [Functional Requirements Specification](#)
 - 2.2.1 [Get Queued Filings Use Case](#)
 - 2.2.2 [Get Case Filing Use Case](#)**
 - 2.2.3 [Import Acknowledgment Use Case](#)
 - 2.2.4 [Get Documents Use Case](#)
 - 2.2.5 [Import Documents Acknowledgment Use Case](#)
 - 2.3. [User Characteristics](#)
 - 2.4. [Non-functional Requirements](#)
 - 2.5. [External Interface Requirements](#)
 - 2.6 [Detailed Non-functional Requirements](#)
 - 2.7 [Security](#)
3. [Appendix A – E-Filing Service .WSDL](#)
4. [Appendix B – SOAP Client Flowchart](#)

INTRODUCTION

PURPOSE

The purpose of this document is to describe the requirements for a CMS SOAP Client (hereinafter called system). This document describes the features and the interfaces of the system. This document is for both the developers who are to design the CMS SOAP Client and the developers who will use the system to aide in the integration of their CMS with the AOC's E-Filing project.

SCOPE OF PROJECT

The CMS SOAP Client is a middleware module which will act as an interpreter between a CMS and the AOC's E-filing web application. A User with the intention of interacting with the CMS or a program will, on demand or programmatically on a batch mode, ask the system to retrieve either a list of case filings ready to be imported or the case information for a particular case filing. The response from the E-filing service will be provided to CMS in accordance to the CMS specification.

GLOSSARY

The following terms and abbreviations will be used throughout the document, and are referenced here for the ease of the reader:

Term	Definition
The System	CMS SOAP Client
AOC	Georgia Administrative Office of the Courts
CMS	Case Management System
SOAP	Simple Object Access Protocol
XML	Extensible Markup Language
WSDL	Web Service Definition Language
User	A user that intends to enter a case or documents in the CMS
Case ready to be imported	In GAJE Child Support Efiling a case is marked as ready to be imported when the case has never been imported before and the clerk is

satisfied and determines that the case can be imported or a case has been already imported together with its leading and supporting documents but additional documents have been introduced into the Efiling system and the clerk is satisfied and determines that the additional documents must be imported

REFERENCES

IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998

OVERVIEW OF DOCUMENT

The Overall Description section of this document provides an overview of the system in its entirety. It describes the context for the technical requirements that follow. The third section, the Requirements Specification section gives the exact technical specifications to which the system functions.

OVERALL DESCRIPTION

2.1 SYSTEM ENVIRONMENT

The system interacts directly only with other programs and/or a User. The system will wait for a command from a program or a User and then form the proper SOAP request message to invoke the AOC's E-filing service. Once the system receives a response from the E-filing service, it will take the appropriate action and provide an output for the CMS in accordance with the CMS specification

2.2 FUNCTIONAL REQUIREMENTS SPECIFICATION

This section describes the general configuration of the system. In this configuration all method calls are initiated by a User with the intention of integrating with the CMS or by a program directly or on batch mode.

2.2.1 GET QUEUED FILINGS USE CASE

BRIEF DESCRIPTION

The User or a program will fire up the system to ask the E-filing service for a list of case filing identifiers for cases that are ready to be imported into their system.

Step-By-Step Description

1. The user on demand, or a program on either demand or batch mode will fire up the system for a list of case filing identifiers for cases that are ready to be imported into their system.
2. The system will form the SOAP request message to invoke the E-filing service's method 'getQueuedFilings'.
3. The system will parse the response from the E-filing service and provide the output to the CMS in accordance with the CMS specification.

The formal definition of the use case is as follows:

GET QUEUED FILINGS

Use Case Name	Get Queued Filings
Reference	Section 2.2.1
Trigger	The User or a program notifies the system to invoke the getQueuedFilings method of the E-Filing service.
Precondition	None.
Basic Path	<ol style="list-style-type: none">1. The User or a program notifies the system to invoke the getQueuedFilings method of the E-Filing system.2. The system forms and sends the SOAP request message to AOC's E-filing service method 'getQueuedFilings.'3. The system consumes the response, which is a list of case filing identifiers, and provides the list to the User or a program in accordance with the CMS specification
Alternative Paths	None.
Post-condition	None.
Exception Paths	If there is a fault returned from the E-filing service or the SOAP response is malformed, the system will attempt two subsequent retries. If these are unsuccessful, then the system will return an error message to the CMS.

2.2.2 GET CASE FILING USE CASE

Brief Description

The user on demand or the system automatically will ask the E-filing service for all case filing information for a particular case identifier.

Step-By-Step Description

1. The user on demand, ask the system for a particular case filing identifier or the system

2. The system will form the SOAP request message to invoke the E-filing service's method 'getCaseFiling' for the particular identifier.
3. The system will take the payload of the response from the E-filing service and provide it to the CMS in accordance with the CMS specification.

The formal definition of the use case is as follows:

GET CASE FILING

Use Case Name	Get Case Filing
Reference	Section 2.2.2.
Trigger	The User or a program will ask the system to invoke the getCaseFiling method of the E-Filing system for a given case identifier.
Precondition	The system has previously invoked the getQueuedFilings method (described above) to get the unique case filing identifiers of cases ready to be imported to the CMS.
Basic Path	<ol style="list-style-type: none"> 1. The User or the program in accordance with the CMS specification notifies the system to invoke the getCaseFiling method of the E-Filing system and passes a unique case filing identifier. 2. The system forms the SOAP request message to invoke the AOC's E-filing service method 'getCaseFiling.' 3. The system consumes the response from the AOC's E-Filing service, and provides it to the CMS in accordance to the CMS specification. The AOC's E-Filing SOAP response message constitutes the meta data of the case filing for the particular unique case filing identifier.
Alternative Paths	None.
Post-condition	None.
Exception Paths	If there is a fault returned from the E-filing service or the SOAP response is malformed, the system will attempt two subsequent retries. If these are unsuccessful, then the system will return an error message to the CMS.

2.2.3 IMPORT ACKNOWLEDGEMENT USE CASE

Brief Description

CMS conveys an acknowledgement message to the AOC'S e-Filing service.

Step-By-Step Description

1. The CMS will ask the system, in accordance with the CMS specification, to send a SOAP request message to the AOC's E-Filing service in regards with the effort of the CMS to create a case record for a particular unique case filing identifier.
2. The system will form the SOAP request message to invoke the E-filing service's method "importAck".
3. The system will parse the response from the E-filing service and may or may not provide the output to the CMS in accordance with the CMS specification.

The formal definition of the use case is as follows:

IMPORT ACKNOWLEDGMENT USE CASE

Use Case Name	Import acknowledgment use case
Reference	Section 2.2.3
Trigger	The CMS notifies the system to invoke the importAck method of the E-Filing system.
Precondition	The User or a program has previously invoked the getCaseFiling method (described above) to get the case filing meta data of a particular case identifier.
Basic Path	<ol style="list-style-type: none">1. The CMS in accordance with the CMS specification notifies the system to invoke the importAck method of the E-Filing service.2. The system forms the SOAP request message to invoke the AOC's E-filing service method "importAck".3. The system consumes the response from the AOC's E-Filing service, and provides it to the CMS in accordance to the CMS specification.
Alternative Paths	None.
Post-condition	None.
Exception Paths	If there is a fault returned from the E-filing service or the SOAP response is malformed, the system will attempt two subsequent retries. If these are unsuccessful, then the system will return an error message to the CMS.

2.2.4 GET DOCUMENTS USE CASE

Brief Description

The CMS will ask the E-filing service for all documents associated with a particular imported case.

Step-By-Step Description

1. The CMS will ask the system, in accordance with the CMS specification, to send a SOAP request message to the AOC's E-Filing service for all documents associated with a particular imported case.
2. The system will form the SOAP request message to invoke the E-filing service's method "getDocuments".
3. The system will parse the response from the E-filing service and provide the output to the CMS.

The formal definition of the use case is as follows:

GET DOCUMENTS USE CASE

Use Case Name	Get documents use case
Reference	Section 2.2.4

Trigger	The User or a program fires up the system to invoke the getDocuments method of the E-Filing service.
Precondition	The system has previously invoked the importAck method (described above) to notify the E-Filing service that the CMS had successfully imported a case.
Basic Path	<ol style="list-style-type: none"> 1. The User or a program in accordance with the CMS specification notifies the system to invoke the getDocuments method of the E-Filing service. 2. The system forms the SOAP request message to invoke the AOC's E-filing service method "getDocuments". 3. The system consumes the response from the AOC's E-Filing service, and provides it to the CMS in accordance to the CMS specification.
Alternative Paths	None.
Post-condition	None.
Exception Paths	If there is a fault returned from the E-filing service or the SOAP response is malformed, the system will attempt two subsequent retries. If these are unsuccessful, then the system will return an error message to the CMS.

2.2.5 IMPORT DOCUMENTS ACKNOWLEDGEMNT USE CASE

Brief Description

CMS conveys an acknowledgement message to the AOC'S e-Filing service.

Step-By-Step Description

1. The CMS will ask the system, in accordance with the CMS specification, to send a SOAP request message to the AOC's E-Filing service in regards with the effort of the CMS to import all documents for a particular imported case.
2. The system will form the SOAP request message to invoke the E-filing service's method "docAck".
3. The system will parse the response from the E-filing service and may or may not provide the output to the CMS in accordance with the CMS specification.

The formal definition of the use case is as follows:

IMPORT DOCUMENTS ACKNOWLEDGMENT USE CASE

Use Case Name	Import Documents acknowledgment use case
Reference	Section 2.2.5
Trigger	The CMS notifies a program to fire up the system to invoke the docAck method of the E-Filing service.
Precondition	The system has previously invoked the getDocuments method (described above) to import all the documents associated with a case.
Basic Path	<ol style="list-style-type: none"> 1. The CMS in accordance with the CMS notifies the system to invoke the docAck method of the E-Filing service. 2. The system forms the SOAP request message to invoke the AOC's E-

		filing service method "docAck".
		3. The system consumes the response from the AOC's E-Filing service, and provides it to the CMS in accordance to the CMS specification.
Alternative Paths	None.	
Post-condition	None.	
Exception Paths		If there is a fault returned from the E-filing service or the SOAP response is malformed, the system will attempt two subsequent retries. If these are unsuccessful, then the system will return an error message to the CMS.

2.3 USER CHARACTERISTICS

The user of the system is expected to develop a program that will invoke the methods of the system.

The user of the system is also expected to develop a program that integrates the system's consumption of the payload of the response from the E-filing service and provide it to the CMS in accordance with the CMS specification.

2.4 NON-FUNCTIONAL REQUIREMENTS

The system will have write permissions to whichever directory the user configures the system to save files to. The system will have Internet connectivity in order to communicate with the E-filing service.

2.5 EXTERNAL INTERFACE REQUIREMENTS

The system interfaces with two separate entities:

1. The system interfaces with CMS.

The CMS invokes the system's request methods and receives the consumed by the system AOC's E-Filing service in accordance with the CMS specification.

2. The system interfaces with the AOC's E-filing service. The system both sends and receives SOAP messages to and from the AOC's E-Filing service

2.6 DETAILED NON-FUNCTIONAL REQUIREMENTS

Please see Appendix A for the .WSDL of the service which gives the XML structure of all entities.

2.7 SECURITY

Standard system administrative practices should be used when integrating the system into the CMS system. The system will authenticate with the E-filing service through HTTPS.

APPENDIX A - E-FILING SERVICE .WSDL

This .WSDL is also available online at the following URL:

https://www.gaje.us/ws/GAJE_ImportWS?wsdl

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://GAJEWS.gaaoc.us/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wsoap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:t="http://GAJEWS.gaaoc.us/types/"
  xmlns:ws="http://GAJEWS.gaaoc.us/"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd"
  xmlns:fi="http://java.sun.com/xml/ns/wsit/2006/09/policy/fastinfoset/service"
  xmlns:tcp="http://java.sun.com/xml/ns/wsit/2006/09/policy/soaptcp/service"
  xmlns:wsp="http://www.w3.org/ns/ws-policy"
  xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
  xmlns:sc="http://schemas.sun.com/2006/03/wss/server"
  xmlns:wsppl="http://java.sun.com/xml/ns/wsit/policy"
  xmlns:sp="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702"

  >

  <wsdl:documentation>This service allows for the retrieval of case records that are
ready to be
imported into a case management system.</wsdl:documentation>

  <wsdl:types>
    <xs:schema targetNamespace="http://GAJEWS.gaaoc.us/types/"
      xmlns="http://GAJEWS.gaaoc.us/types/"
      elementFormDefault="unqualified"
      attributeFormDefault="unqualified">

      <xs:element name="getQueuedFilingsRequest">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="courtID" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>

      <xs:element name="getQueuedFilingsResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="cases" type="uuidList"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
  </wsdl:types>
</wsdl:definitions>
```

```

        </xs:sequence>
    </xs:complexType>
</xs:element>

    <xs:element name="getCaseFilingRequest">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="uuid" type="xs:string"
minOccurs="1" maxOccurs="1"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element name="getCaseFilingResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="case_filing"
type="caseFiling"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element name="importAckRequest">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="ack"
type="importAck_type"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element name="importAckResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="success" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element name="getDocumentsRequest">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="caseNum"
type="xs:string"/>
                <xs:element name="uuid" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element name="getDocumentsResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="documents"
type="documentList"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

```

```

        </xs:complexType>
    </xs:element>

    <xs:element name="docAckRequest">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="caseUuid"
type="xs:string"/>
                <xs:element name="docAck" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element name="docAckResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="docSuccess"
type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element name="resetRequest">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="reset" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element name="resetResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="resetSuccess"
type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:complexType name="importAck_type">
        <xs:sequence>
            <xs:element name="success" type="xs:string"/>
            <xs:element name="uuid" type="xs:string"/>
            <xs:element name="caseNum" type="xs:string"/>
            <xs:element name="judge" type="xs:string"/>
            <xs:element name="reason" type="xs:string"
minOccurs="0" maxOccurs="1"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="organizationList">
        <xs:sequence minOccurs="0" maxOccurs="unbounded">
            <xs:element name="organization"
type="organization_type"/>
        </xs:sequence>
    </xs:complexType>

```

```

</xs:complexType>

<xs:complexType name="caseFiling">
  <xs:sequence>
    <xs:element name="docketID" type="xs:string"
nillable="true"/>
    <xs:element name="divID" type="xs:int"/>
    <xs:element name="type" type="xs:int"/>
    <xs:element name="category" type="xs:int"/>
    <xs:element name="filingType" type="xs:int"/>
    <xs:element name="caption" type="xs:string"/>
    <xs:element name="date" type="xs:string"/>
    <xs:element name="dispositionDate"
type="xs:string"/>
    <xs:element name="status" type="xs:string"/>
    <xs:element name="prosecutionAttorneys"
type="caseOfficialList"/>
    <xs:element name="defenseAttorneys"
type="caseOfficialList"/>
    <xs:element name="initiatingParties"
type="casePartyList"/>
    <xs:element name="caseInitiatingPartyOrganizations"
type="casePartyList"/>
    <xs:element name="caseDefendantParties"
type="casePartyList"/>
    <xs:element name="witnesses" type="casePartyList"/>
    <xs:element name="events" type="caseEventList"/>
    <xs:element name="courtID" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="caseOfficialList">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:element name="caseOfficial" type="case_official"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="case_official">
  <xs:sequence>
    <xs:element name="person" type="person_type"/>
    <xs:element name="barID" type="xs:string"/>
    <xs:element name="cmsCode" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="person_type">
  <xs:sequence>
    <xs:element name="lastName" type="xs:string"/>
    <xs:element name="middleName" type="xs:string"/>
    <xs:element name="firstName" type="xs:string"/>
    <xs:element name="suffix" type="xs:string"/>
    <xs:element name="addresses" type="addressList"/>
  </xs:sequence>
</xs:complexType>

```

```

<xs:complexType name="addressList">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:element name="address" type="address_type"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="address_type">
  <xs:sequence>
    <xs:element name="street1" type="xs:string"/>
    <xs:element name="street2" type="xs:string"/>
    <xs:element name="city" type="xs:string"/>
    <xs:element name="state" type="xs:string"/>
    <xs:element name="zip" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="casePartyList">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:element name="caseParty"
type="case_party_type"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="case_party_type">
  <xs:sequence>
    <xs:element name="person" type="person_type"/>
    <xs:element name="cmsTypeCode" type="xs:string"/>
    <xs:element name="organization"
type="organization_type"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="organization_type">
  <xs:sequence>
    <xs:element name="name" type="xs:string"/>
    <xs:element name="addresses" type="addressList"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="caseEventList">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:element name="caseEvent"
type="case_event_type"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="case_event_type">
  <xs:sequence>
    <xs:element name="type" type="xs:string"/>
    <xs:element name="date" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="documentList">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">

```

```

        <xs:element name="document"
type="document_type"/>
    </xs:sequence>
</xs:complexType>

    <xs:complexType name="document_type">
        <xs:sequence>
            <xs:element name="uuid" type="xs:string"/>
            <xs:element name="title" type="xs:string"/>
            <xs:element name="courtID" type="xs:string"/>
            <xs:element name="type" type="xs:string"/>
            <xs:element name="content"
type="xs:base64Binary"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="uuidList">
        <xs:sequence>
            <xs:element name="uuid" type="xs:string"
minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>

    <xs:element name="uuid_list" type="uuidList"/>

    <xs:element name="putDocument">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="document"
                type="document_type">
            </xs:element>
        </xs:sequence>
    </xs:complexType>
    </xs:element>

    <xs:element name="putDocumentResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="out" type="xs:string"></xs:element>
        </xs:sequence>
    </xs:complexType>
    </xs:element>

    </xs:schema>
</wsdl:types>

    <wsdl:message name="getQueuedFilingsRequest">
        <wsdl:part name="parameters" element="t:getQueuedFilingsRequest"/>
    </wsdl:message>

    <wsdl:message name="getQueuedFilingsResponse">
        <wsdl:part name="parameters" element="t:getQueuedFilingsResponse"/>
    </wsdl:message>

    <wsdl:message name="getCaseFilingRequest">

```

```

        <wsdl:part name="parameters" element="t:getCaseFilingRequest"/>
    </wsdl:message>

    <wsdl:message name="getCaseFilingResponse">
        <wsdl:part name="parameters" element="t:getCaseFilingResponse"/>
    </wsdl:message>

    <wsdl:message name="importAckRequest">
        <wsdl:part name="parameters" element="t:importAckRequest"/>
    </wsdl:message>

    <wsdl:message name="importAckResponse">
        <wsdl:part name="parameters" element="t:importAckResponse"/>
    </wsdl:message>

    <wsdl:message name="getDocumentsRequest">
        <wsdl:part name="parameters" element="t:getDocumentsRequest"/>
    </wsdl:message>

    <wsdl:message name="getDocumentsResponse">
        <wsdl:part name="parameters" element="t:getDocumentsResponse"/>
    </wsdl:message>

    <wsdl:message name="docAckRequest">
        <wsdl:part name="parameters" element="t:docAckRequest"/>
    </wsdl:message>

    <wsdl:message name="docAckResponse">
        <wsdl:part name="parameters" element="t:docAckResponse"/>
    </wsdl:message>

    <wsdl:message name="resetRequest">
        <wsdl:part name="parameters" element="t:resetRequest"/>
    </wsdl:message>

    <wsdl:message name="resetResponse">
        <wsdl:part name="parameters" element="t:resetResponse"/>
    </wsdl:message>

    <wsdl:message name="putDocumentRequest">
        <wsdl:part name="parameters" element="t:putDocument"></wsdl:part>
    </wsdl:message>
    <wsdl:message name="putDocumentResponse">
        <wsdl:part name="parameters"
element="t:putDocumentResponse"></wsdl:part>
    </wsdl:message>
    <wsdl:portType name="queuedFilings">
        <wsdl:operation name="getQueuedFilings">
            <wsdl:input message="ws:getQueuedFilingsRequest"
name="getQueuedFilingsRequest"/>
            <wsdl:output message="ws:getQueuedFilingsResponse"
name="getQueuedFilingsResponse"/>
        </wsdl:operation>
        <wsdl:operation name="getCaseFiling">

```



```

        <wsdl:input message="ws:getCaseFilingRequest"
name="getCaseFilingRequest"/>
        <wsdl:output message="ws:getCaseFilingResponse"
name="getCaseFilingResponse"/>
    </wsdl:operation>
    <wsdl:operation name="importAck">
        <wsdl:input message="ws:importAckRequest"
name="importAckRequest"/>
        <wsdl:output message="ws:importAckResponse"
name="importAckResponse"/>
    </wsdl:operation>
    <wsdl:operation name="getDocuments">
        <wsdl:input message="ws:getDocumentsRequest"
name="getDocumentsRequest"/>
        <wsdl:output message="ws:getDocumentsResponse"
name="getDocumentsResponse"/>
    </wsdl:operation>
    <wsdl:operation name="docAck">
        <wsdl:input message="ws:docAckRequest" name="docAckRequest"/>
        <wsdl:output message="ws:docAckResponse"
name="docAckResponse"/>
    </wsdl:operation>
    <wsdl:operation name="reset">
        <wsdl:input message="ws:resetRequest" name="resetRequest"/>
        <wsdl:output message="ws:resetResponse" name="resetResponse"/>
    </wsdl:operation>
    <wsdl:operation name="putDocument">
        <wsdl:input message="ws:putDocumentRequest"
name="putDocumentRequest" ></wsdl:input>
        <wsdl:output message="ws:putDocumentResponse"
name="putDocumentResponse"></wsdl:output>
    </wsdl:operation>
</wsdl:portType>

    <wsdl:binding name="soap12Binding" type="ws:queuedFilings">
        <wsp:PolicyReference URI="#NoClientSSLUsernameAuthPolicy"/>

        <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
style="document"/>

        <wsdl:operation name="getQueuedFilings">
            <soap:operation soapAction=""/>
            <wsdl:input name="getQueuedFilingsRequest">
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="getQueuedFilingsResponse">
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>

        <wsdl:operation name="getCaseFiling">
            <soap:operation soapAction=""/>
            <wsdl:input name="getCaseFilingRequest">
                <soap:body use="literal"/>
            </wsdl:input>

```

```
        <wsdl:output name="getCaseFilingResponse">
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>

    <wsdl:operation name="importAck">
        <soap:operation soapAction=""/>
        <wsdl:input name="importAckRequest">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="importAckResponse">
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>

    <wsdl:operation name="getDocuments">
        <soap:operation soapAction=""/>
        <wsdl:input name="getDocumentsRequest">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="getDocumentsResponse">
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>

    <wsdl:operation name="docAck">
        <soap:operation soapAction=""/>
        <wsdl:input name="docAckRequest">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="docAckResponse">
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>

    <wsdl:operation name="reset">
        <soap:operation soapAction=""/>
        <wsdl:input name="resetRequest">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="resetResponse">
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>

    <wsdl:operation name="putDocument">
        <soap:operation soapAction=""/>
        <wsdl:input name="putDocumentRequest">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="putDocumentResponse">
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
```

```
</wsdl:binding>
<wsdl:service name="GAJE_ImportWSService">
  <wsdl:port name="GAJE_ImportWS" binding="ws:soap12Binding">
    <soap:address location="https://www.gaje.ws/ws/GAJE_ImportWS"/>
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

APPENDIX B - SOAP CLIENT FLOWCHART

