

**Formal Requirements Specification**  
**Organizational Alert System**

**For**  
**CS 895 MSE Project**  
**Department of Computer Science**  
**Kansas State University**

**Submitted to**  
**Dr. Mitch Neilsen**  
**Dr. Torben Amtoft**  
**Dr. Scott Deloach**

**Submitted by**

**Angela Hall**

**2/10/2019**

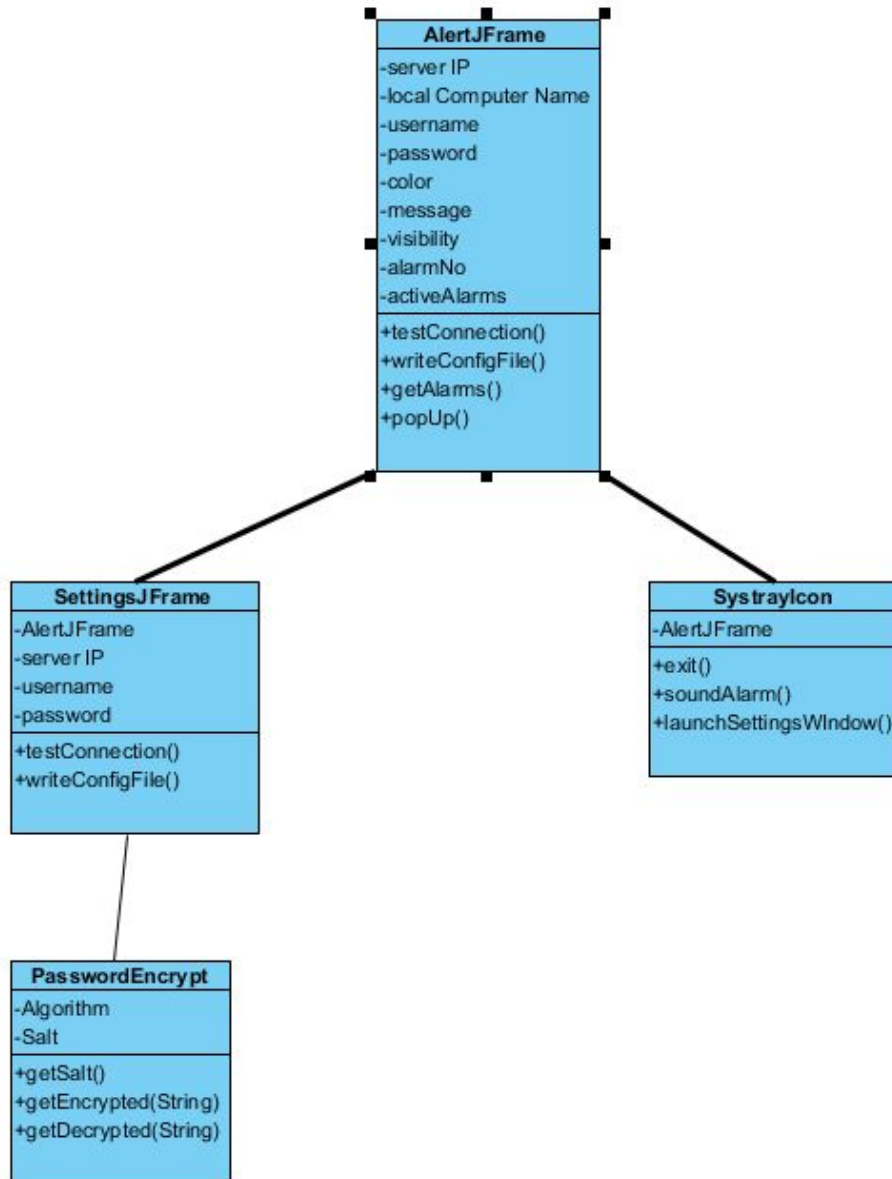
## 1. **Overview**

This document contains the formal specifications for the client and server of the Organizational Alert System. The Java libraries for window building, file operations, encrypting, and creating GET server requests are abstracted out for clarity.

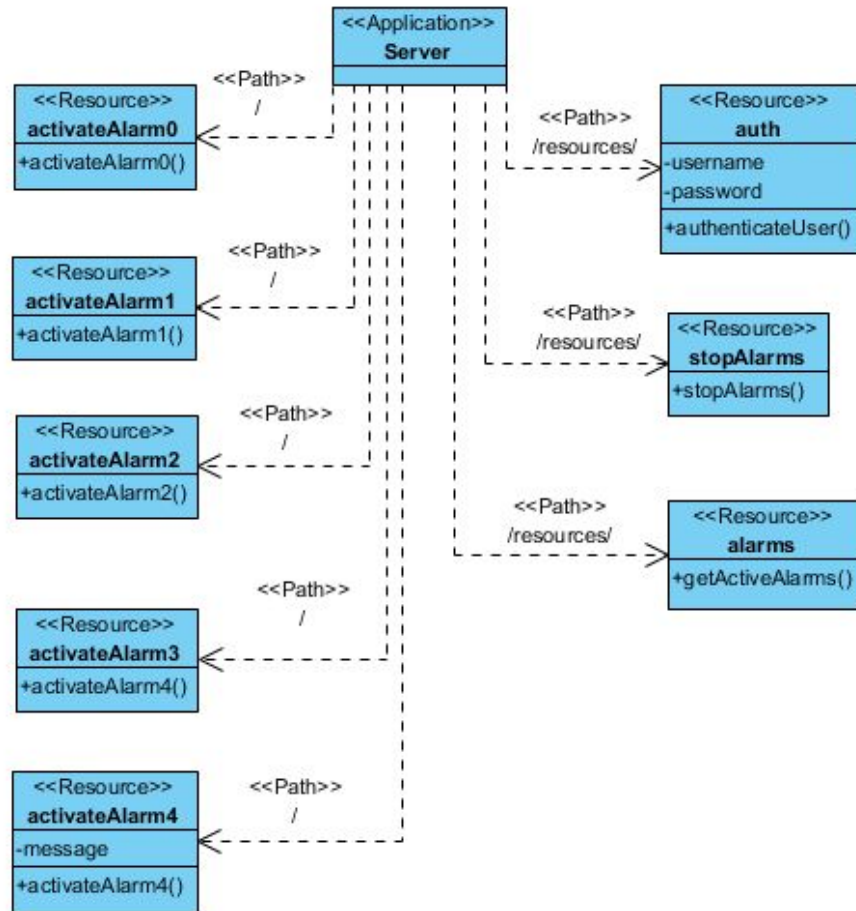
The formal specification language for the Java portion of the project is OCL. The consistency of the OCL specification was tested with the USE specification tool.

The formal specification language for the REST api is OpenAPI. The consistency of the REST specification was test with the Swagger specification toolset.

## 2. Class Diagrams Client (Java)



## Server (REST API implemented in PHP/XML)



### 3. OCL Specification

Specifies the classes, associations and constraints of the Java Client

model client

-- classes

--built in Java Classes

--that are abstracted from class diagram

--but needed for USE to compile

class Color

attributes

red: Integer

blue: Integer

green: Integer

end

class Cipher

attributes

name: String

end

--represents XML data

class Alarm

attributes

number: Integer

message: String

end

-- actual architecture

class AlertJFrame

attributes

serverIP: String

localComputerName: String

username: String

password: String

color: Color

message: String

visible: Boolean

alarmNo: Integer

alarms: Alarm

end

```
class SettingsJFrame
attributes
    AlertJFrame: AlertJFrame
    serverIP: String
    username: String
    password: String
end
```

```
class PasswordEncrypt
attributes
    Algorithm: Cipher
    Salt: String
end
```

```
class SystrayIcon
attributes
    AlertJFrame: AlertJFrame
end
```

```
-- associations
association alarmsList between
    AlertJFrame[1]
    Alarm[*]
end
```

```
association has between
    AlertJFrame[1]
    SettingsJFrame[1]
end
```

```
association uses between
    SettingsJFrame[1]
    PasswordEncrypt[1]
end
```

```
association shows between
    AlertJFrame[1]
    SystrayIcon[1]
end
```

```
-- OCL constraints
```

```

constraints
context AlertJFrame
    -- can't have more than 4 alarms
    inv only4Alarms:
        self.alarms->size <= 4

context AlertJFrame
    -- if there are active alarms,
    -- this JFrame is visible.
    inv activeAlarmsMakesVisible:
        self.visible implies self.alarms->size > 0

context AlertJFrame
    -- if there are no active alarms,
    -- this JFrame is invisible
    inv inactiveAlarmsMakesInvis:
        self.alarms->size < 1 implies not self.visible

```

#### 4. OpenAPI Specification

Formal Specification for PHP/XML REST API using OpenAPI specification language

```

swagger: "2.0"
info:
  description: "Headsup Organization-Wide Alert System Server REST API Specification"
  version: "1.0"
  title: "Headsup Server API"
  contact:
    email: angie.c.hall@gmail.com
  license:
    name: "Apache 2.0"
    url: "http://www.apache.org/licenses/LICENSE-2.0.html"
host: "192.168.0.5"
basePath: "/"
tags:
- name: "Activate Alarms"
  description: "activate alarms"
- name: "Get Active Alarms"
  description: "get active alarms"
- name: "Stop Alarms"
  description: "stop alarms"
- name: "Authenticate"

```

description: "authenticate users"

schemes:

- "http"

paths:

/activateAlarm0:

get:

tags:

- "Activate Alarms"

summary: "activate alarm 0"

description: ""

produces:

- "application/xml"

parameters: []

responses:

200:

description: "OK"

/activateAlarm1:

get:

tags:

- "Activate Alarms"

summary: "activate alarm 1"

description: ""

produces:

- "application/xml"

parameters: []

responses:

200:

description: "OK"

/activateAlarm2:

get:

tags:

- "Activate Alarms"

summary: "activate alarm 2"

description: ""

produces:

- "application/xml"

parameters: []

responses:

200:

description: "OK"

/activateAlarm3:

get:

tags:



- "Activate Alarms"  
summary: "activate alarm 3"  
description: ""  
produces:  
- "application/xml"  
parameters: []  
responses:  
200:  
description: "OK"

/activateAlarm4:

get:  
tags:  
- "Activate Alarms"  
summary: "activate alarm 4"  
description: ""  
produces:  
- "application/xml"  
parameters:  
- in: "body"  
name: "message"  
responses:  
200:  
description: "OK"

/resources/alarms:

get:  
tags:  
- "Get Active Alarms"  
summary: "get all currently active alarms"  
description: ""  
produces:  
- "application/xml"  
parameters:  
- in: "body"  
name: "body"  
description: "active alarms"  
schema:  
\$ref: "#/definitions/Alarm"  
responses:  
200:  
description: "OK"

/resources/stopAlarms:

get:  
tags:

- "Stop Alarms"  
summary: "stop all currently active alarms"  
description: ""  
produces:  
- "application/xml"  
parameters: []  
responses:  
200:  
description: "OK"

/resources/auth:

get:  
tags:  
- "Authenticate"  
summary: "authenticate users"  
description: "authenticate given password against database"  
produces:  
- "application/xml"  
parameters:  
- in: "body"  
name: "body"  
description: "check user against database"  
schema:  
\$ref: "#/definitions/Authenticated"  
responses:  
200:  
description: "OK"

definitions:

Alarm:  
type: "object"  
properties:  
alarms:  
type: "object"  
properties:  
pk:  
type: "integer"  
format: "int64"  
message:  
type: "string"  
example: "Active Shooter"

xml:  
name: "Alarm"

Authenticated:  
type: "object"

properties:

username:

type: "string"

example: "John Doe"

authenticated:

type: "boolean"

xml:

name: "Authed"