AXON CFS

Software Design Document

Change Log

|  |  |  |
| --- | --- | --- |
| Date | Change By | Detail |
| 17 Mar 2021 | Danh | Init document |
| 18 Mar 2021 | Danh | Add introduction and table of contents |
| 19 Mar 2021 | Danh | Finish document |

Table of Contents

[1. Overview 4](#_Toc67089043)

[1.1. Document Scope and Purpose 4](#_Toc67089044)

[1.2. Acronyms / Abbreviations 4](#_Toc67089045)

[1.3. System Environment 4](#_Toc67089046)

[1.4. Architecture Design 4](#_Toc67089047)

[1.5. High-level View 5](#_Toc67089048)

[2. Modules 5](#_Toc67089049)

[2.1. Identity Service 5](#_Toc67089051)

[2.2. API Service 5](#_Toc67089052)

[2.2.1. Events 5](#_Toc67089053)

[3. Entity Diagram 5](#_Toc67089054)

[4. Class Diagrams 6](#_Toc67089055)

[5. Sequence Diagrams 8](#_Toc67089056)

[5.1. Login 8](#_Toc67089060)

[5.2. Search event 8](#_Toc67089061)

[5.3. Create event 9](#_Toc67089062)

[6. API Format 9](#_Toc67089063)

[6.1. Get user token 9](#_Toc67089070)

[Headers 9](#_Toc67089071)

[Request Body 9](#_Toc67089072)

[Response 10](#_Toc67089073)

[6.2. Search event 10](#_Toc67089074)

[Headers 10](#_Toc67089075)

[Parameters 10](#_Toc67089076)

[Response 11](#_Toc67089077)

[6.3. Create event 11](#_Toc67089078)

[Headers 11](#_Toc67089079)

[Request body 11](#_Toc67089080)

[Request sample 12](#_Toc67089081)

[Response 12](#_Toc67089082)

# Overview

This introduction provides an overview of the Software Design Document for Axon Call For Service (CFS) management system. It includes the purpose, scope, design approach, main component design and high-level system design considerations of the system.

## Document Scope and Purpose

This document provides a description of the technical design for Call For Service - API. This document’s primary purpose is to describe the technical vision for how business requirements will be realized. This document provides an architectural overview of the system to depict different aspects of the system. This document also functions as a foundational reference point for developers.

## Acronyms / Abbreviations

|  |  |
| --- | --- |
| **Acronym** | **Meaning** |
| **CFS** | Call For Service |

## System Environment

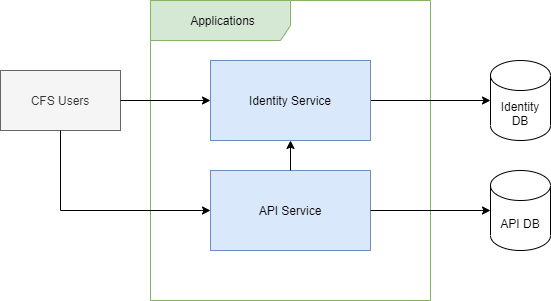
* Development: Visual Studio 2019 / Visual Studio Mac / Visual Studio Code
* Unit Test: xUnit
* Diagrams: draw.io
* Database: MSSQL
* Deployment: Azure App Services

## Architecture Design

CFS Management System will follow N-Tier Architecture. A principal advantage to this design is the relative stability of the components as seen by the applications developer. Implementations may change considerably to enhance the performance or in response to changes in the architecture. These changes are less likely to cause major impact to the applications’ programs.

* Identity Service: manage users, as well as handling user’s token
* API tier: exposing APIs to users, as well as API configurations
* Application tier: contains business logic of application
* Domain tier: contains domain models of the application
* Infrastructure tier: contains database context, code-first model mappings data migrations and DI injectors

## High-level View



# Modules



## Identity Service

Provide a common way to authenticate requests to CFS API Service, as well as future applications.

## API Service

## Events

Events management. User can create and view events based on assigned agency.

# Entity Diagram

Diagram

Description automatically generated

# Class Diagrams

Graphical user interface, diagram

Description automatically generated

Figure 1 Application tier class diagram

Diagram

Description automatically generated

Figure 2 Domain tier class diagram

Graphical user interface, diagram

Description automatically generated

Figure 3 Data tier class diagram

# Sequence Diagrams



## Login

Diagram

Description automatically generated

Figure 4 Login sequence diagram

## Search event

Diagram

Description automatically generated

Figure 5 Search event sequence diagram

## Create event

Diagram

Description automatically generated

Figure 6 Create event sequence diagram

# API Format



## Get user token

|  |  |
| --- | --- |
| **Method** | **URL** |
| POST | {{authHost}}/connect/token |

### Headers

|  |  |
| --- | --- |
| **Key** | **Value** |
| Content-Type | application/x-www-form-urlencoded |

### Request Body

|  |  |
| --- | --- |
| **Key** | **Value** |
| grant\_type | password |
| scope | scope1 |
| client\_id | ro.client |
| client\_secret | secret |
| username | {username} |
| password | {password} |

### Response

|  |  |  |
| --- | --- | --- |
| **Status** | **Code** | **Body** |
| Success | 200 | {  "access\_token": {token}  "expires\_in": 3600,  "token\_type": "Bearer",  "scope": "scope1"  } |
| Fail | 400 | {  "error": "invalid\_grant"  } |

## Search event

|  |  |
| --- | --- |
| **Method** | **URL** |
| GET | {{host}}/api/v1/events |

### Headers

|  |  |
| --- | --- |
| **Key** | **Value** |
| Authorization | Bearer {token} |

### Parameters

|  |  |  |
| --- | --- | --- |
| **Name** | **Data Type** | **Description** |
| fromDate | date time | search from date |
| toDate | date time | search to date |
| responder | string | responder code |
| offset | number | records’ offset |
| limit | number | limit of records per one page |

### Response

|  |  |  |
| --- | --- | --- |
| **Status** | **Code** | **Body** |
| Success | 200 | {  "events": [  {  "agency\_id": "4b63867c-e27c-41da-9c06-5e10817c1266",  "event\_id": "c669bcd7-511b-432d-904c-9e2d01b16ae1",  "event\_number": "987654",  "event\_type\_code": "TEST",  "event\_time": "2021-03-18T18:09:55.265",  "dispatch\_time": "2021-03-19T18:09:55.265",  "responder": "Responder\_03"  }  ],  "total": 1  } |

## Create event

|  |  |
| --- | --- |
| **Method** | **URL** |
| POST | {{host}}/api/v1/events |

### Headers

|  |  |
| --- | --- |
| **Key** | **Value** |
| Authorization | Bearer {token} |

### Request body

|  |  |  |
| --- | --- | --- |
| **Name** | **Data Type** | **Description** |
| event\_number | string | event number |
| event\_code | string | event type code |
| event\_time | date time | event time |
| dispatch\_time | date time | event dispatch time |
| responder | string | responder code |

### Request sample

{

"event\_number": "45456",

"event\_code": "TEST",

"event\_time": "2021-03-25T18:09:55.265Z",

"dispatch\_time": "2021-03-26T18:09:55.265Z",

"responder": "Responder\_03"

}

### Response

|  |  |  |
| --- | --- | --- |
| **Status** | **Code** | **Body** |
| Success | 200 | {  "agency\_id": "4b63867c-e27c-41da-9c06-5e10817c1266",  "event\_id": "133271be-05cb-490d-8246-c5b3984089b5",  "event\_number": "45456",  "event\_type\_code": "TEST",  "event\_time": "2021-03-25T18:09:55.265Z",  "dispatch\_time": "2021-03-26T18:09:55.265Z",  "responder": "Responder\_03"  } |