# Introduction to UML Diagrams

This report provides an overview of the UML diagrams designed for the system requirements of the application. The UML (Unified Modelling Language) diagrams are visual representations that depict the structure, behaviour, and interactions of the system components. Three types of UML diagrams have been included: Class Diagram, Sequence Diagram, and Activity Diagram.

These UML diagrams serve as powerful tools for understanding and documenting the system's functionality and structure. They help in visualizing the relationships between different components, the flow of information, and the sequence of activities. The diagrams provided in this report offer a detailed and comprehensive view of the system's behaviour, facilitating effective communication and analysis during the development and maintenance phases.

# Class Diagrams

The Class Diagram presents a high-level view of the entities and their relationships in the system. It showcases the key classes such as User, Case, and CVDForm, along with their attributes and associations. The diagram also represents the different user roles, including General Public, NSCS Officer, Admin, and OfficerUser. The Class Diagram provides an understanding of the basic structure of the system and the relationships between its entities.

A picture containing text, screenshot, font, number

Description automatically generated

Class Diagram

Figure Class Diagram

The **User** class represents a user in the system. It has the attributes:

·       **username**: A string that stores the username of the user.

·       **password**: A string that stores the password of the user.

·       **role**: A string that represents the role of the user in the system (e.g., General Public, NSCS Officer, Admin).

·       The **Case** class represents a case in the system. It has the attributes:

·       **caseId**: A string that uniquely identifies the case.

·       **title**: A string that represents the title of the case.

·       **description**: A string that describes the details of the case.

·       **userId**: A string that identifies the user associated with the case.

·       **status**: An instance of the **CaseStatus** enumeration that represents the status of the case (e.g., open, closed, in progress).

·       The **CVDForm** class represents a CVD form in the system. It has the attributes:

·       **formId**: A string that uniquely identifies the CVD form.

·       **formData**: A string that stores the data of the CVD form.

·       **userId**: A string that identifies the user who created the CVD form.

·       **officerId**: A string that identifies the officer who views the CVD form.

·       The **GeneralPublic** class represents a user with the role of General Public. It inherits from the **User** class and has the method **createCVDForm()** to allow the creation of CVD forms by the general public.

·       The **NSCSOfficer** class represents a user with the role of NSCS Officer. It inherits from the **User** class and has the method **createCase()** to enable NSCS officers to create cases.

·       The **AdminUser** class represents a user with the role of Admin. It inherits from the **User** class and has the methods **viewAllUsers()** and **deleteAccount()** to provide administrative functionality for viewing all users and deleting user accounts.

·       This detailed explanation of the class diagram provides an understanding of each class, their attributes, and methods, and how they relate to each other in the system.

# Sequence Diagrams

The Sequence Diagram dives deeper into the user login process and highlights the interactions between the User, User Service, and Database components. It details the steps involved in the authentication and authorization process, including the verification of user credentials, retrieval of user data, token generation, and session creation. The Sequence Diagram captures the dynamic flow of messages between the components during the login process.

A close-up of a document

Description automatically generated with low confidence

Sequence Diagram

Figure Sequence Diagram

·       The user initiates the login process by sending their login credentials (username and password) to the User Service.

·       The User Service receives the login credentials and proceeds to verify them.

·       The User Service accesses the Database to retrieve the user data associated with the provided username.

·       The User Service verifies the credentials by comparing the password provided by the user with the password stored in the user data fetched from the Database.

·       If the credentials are valid, the User Service generates an authentication token.

·       The User Service sends the authentication token back to the user.

·       The user receives the authentication token.

·       The user initiates a request to fetch their user data from the User Service.

·       The User Service retrieves the user data from the Database based on the user's request.

·       The User Service validates the user data to ensure its integrity and consistency.

·       If the user data is valid, the User Service creates session data for the user.

·       The User Service sends the session ID, associated with the created session data, back to the user.

·       The user receives the session ID.

·       The user can use the received session ID for subsequent authenticated requests to the User Service.

·       The User Service responds to the user with a login success message, indicating that the login process was successful.

In summary, this detailed sequence diagram represents the flow of events during the user login process, including the steps of verifying credentials, retrieving, and validating user data, generating an authentication token, and creating a session for the user.

# Activity Diagrams

The Activity Diagram focuses on the process of creating a CVD (Contact, Vehicle, Description) form. It outlines the steps involved in this process, starting from the user's initiation of the form creation to the CVD Form Service's response with a success message. The Activity Diagram highlights the activities of validating the form data and storing it in the database. It provides a visual representation of the sequential flow of activities in creating a CVD form.

·       The user initiates the creation of a CVD form by providing the necessary form data.

·       The CVD Form Service validates the provided CVD form data to ensure its integrity and adherence to any defined rules or constraints.

·       Once the CVD form data is validated, the CVD Form Service stores the form data in the Database.

·       The CVD Form Service assigns a unique identifier ("id") to the CVD form data in the Database.

·       The CVD Form Service responds to the user with a success message, indicating that the CVD form creation process was successful.

# Conclusion

The UML diagrams presented in this report, including the Class Diagram, Sequence Diagram, and Activity Diagram, offer a comprehensive and visual representation of the system's structure, behaviour, and interactions. These diagrams provide insights into the system's functionality, user interactions, and data flow. They facilitate effective communication, analysis, and documentation of the system, enabling stakeholders to understand the requirements and processes. By employing these UML diagrams, the development team can design, implement, and maintain the system effectively, ensuring its successful realization while meeting the specified requirements.