System Design Document for Validation and Verification System

Team 4

Chadha, Harshita

Karuppaiah. Boomika

Liao, Wenjun

Department of Computer Science

George Washington University

Professor Stephen Kaisler

Department of Computer Science

George Washington University

Version 1.1 Submitted on - **Nov 13, 2023**

TABLE OF CONTENTS

1. Introduction

1.1 Purpose

The purpose of this Software Design Document (SDD) is to describe the architecture and system design of the Validation and Verification (V&V) Tool. This document serves as a comprehensive guide for the development team and stakeholders involved in the project. It outlines the high-level design, functionality, and components of the V&V Tool, aiming to ensure a clear understanding of its structure and operation.

1.2 Scope

The scope of this document is limited to the description of a comprehensive solution for streamlining the Validation and Verification processes in software engineering. The primary goal is to capture the defined system requirements and present a rigorous and quality architecture for the V&V tool. This document discusses in detail, the overall system architecture and the constituent subsystems that interact together to make the system work. Further, the data design, the component design, the initial user interfaces as well as some potential issues to be resolved are also discussed. Section 1.3 below provides a brief overview of these sections within this document.

1.3 Overview

This document provides a detailed description of the architecture and design of the Validation and Verification (V&V) Tool. It is organized into several sections, each addressing specific aspects of the software design and functionality. The document includes an introduction, system overview, system architecture, data design, component design, human interface design, issues to be resolved, and appendices.The document is organized as follows:

- Section 1 introduces the purpose, scope, overview, and definitions and acronyms used in this document.
- Section 2 presents a general description of the V&V Tool, including its functionality, context, and design.

- Section 3 provides details about the architectural design, decomposition of subsystems, and design rationale.
- Section 4 describes how the V&V Tool manages and organizes data.
- Section 5 offers insights into the components and their functionalities.
- Section 6 explains the user interface and interactions.
- Section 7 highlights any unresolved issues.

1.4 Definitions and Acronyms

This section contains definitions of terms, acronyms, and abbreviations that may be used throughout the Software Design Document. These definitions are included to assist the reader in understanding the content of this document more effectively.

Table 1: Abbreviations and Acronyms

Term	Description
V&V	Verification and Validation
SRD	Software Requirement Document
SDD	Software Design Document
STP	System Test Plan
API	Application Programming Interface
Creator User	Can upload SRD, SDD, and STP documents and use it to enter requirements, design and test plans into the V&V tool.
Reviewer User	Can review requirements, design and test plans and provide feedback.
Administrator User	Can add, delete and modify user roles.

2. System Overview

The Validation and Verification (V&V) Tool is designed as a comprehensive solution to ensure the quality and reliability of software through systematic checks. It functions within the context of software development and maintenance, providing a suite of features that support the rigorous evaluation of software artifacts against defined criteria and standards. The tool encompasses various functionalities critical to the V&V process. These are mentioned below -

- **Authentication**: Secure user access through credential verification and OTP validation.
- **Role-Based Interface**: Customized user experiences based on the user's role within the organization, such as document creators, reviewers, or administrators.
- **Document Management**: Facilities for document creation, editing, reviewing, and approval, along with version control and collaborative commenting.
- **Task Management**: Organization and prioritization of pending tasks and administrative requests for effective workload management.
- **Approval Workflow**: Oversight of document approval processes, ensuring all materials meet the necessary standards before publication or further distribution.

The V&V tool helps the client's software engineering team to capture their development workflow systematically and in a centralized manner. The tool is composed of three main parts - the SRD or software requirements document checker, the SDD or the Software Design Document Checker, and the STP or the Software Test Program checker. Each of these works together to ensure the quality and compliance of the client's software development process.

The SRD (Software Requirements Document) checker helps enter the requirements for the given software project against a Concept of Operations or CONOPS as provided by the project's user. This also helps add comments flag down problem areas within the requirements document and make iterative improvements. The SDD (Software Design Document) checker helps review the software design document and check against the previously approved software requirement

document designed and approved using the first component of the V&V tool. Similar to the SRD checker, here there is functionality provided to add comments, flag down problem areas within the document, and make iterative improvements. The STP (Software Test Program) checker helps to track the Software Test Programs for the software project at hand and helps in verifying that they encompass all necessary test cases and scenarios.

A layered architecture is provided to offer these functionalities, promoting separation of concerns and modular development. The design facilitates scalability and ease of maintenance, with each layer focusing on a distinct aspect of the system's operation, from user interface down to data management. The use of an API layer enables the tool to interact seamlessly with external systems and databases, ensuring data consistency and real-time synchronization.

In the backdrop of increasing complexity of software systems and regulatory demands for software reliability, the V&V Tool stands as a crucial asset. It embodies best practices in software engineering, drawing from methodologies that prioritize thorough testing, user-centric design, and secure access control. The tool’s creation was motivated by the need for a robust mechanism to detect and rectify errors early in the development process, thereby reducing the overall cost and improving the quality of the software products.

3. System Architecture

In this section of the design document for the V&V tool, we present a high-level breakdown of the components that make the system architecture. In the sections that follow, we present the architecture diagram, description of each of the components present in this diagram and the rationale behind selecting the chosen architecture for our tool.

3.1 Architectural Design

The diagram illustrated in the figure below represents the proposed system architecture for the V&V tool. This high level representation consists of 4 main layers that interact with each other to achieve the system's capabilities.

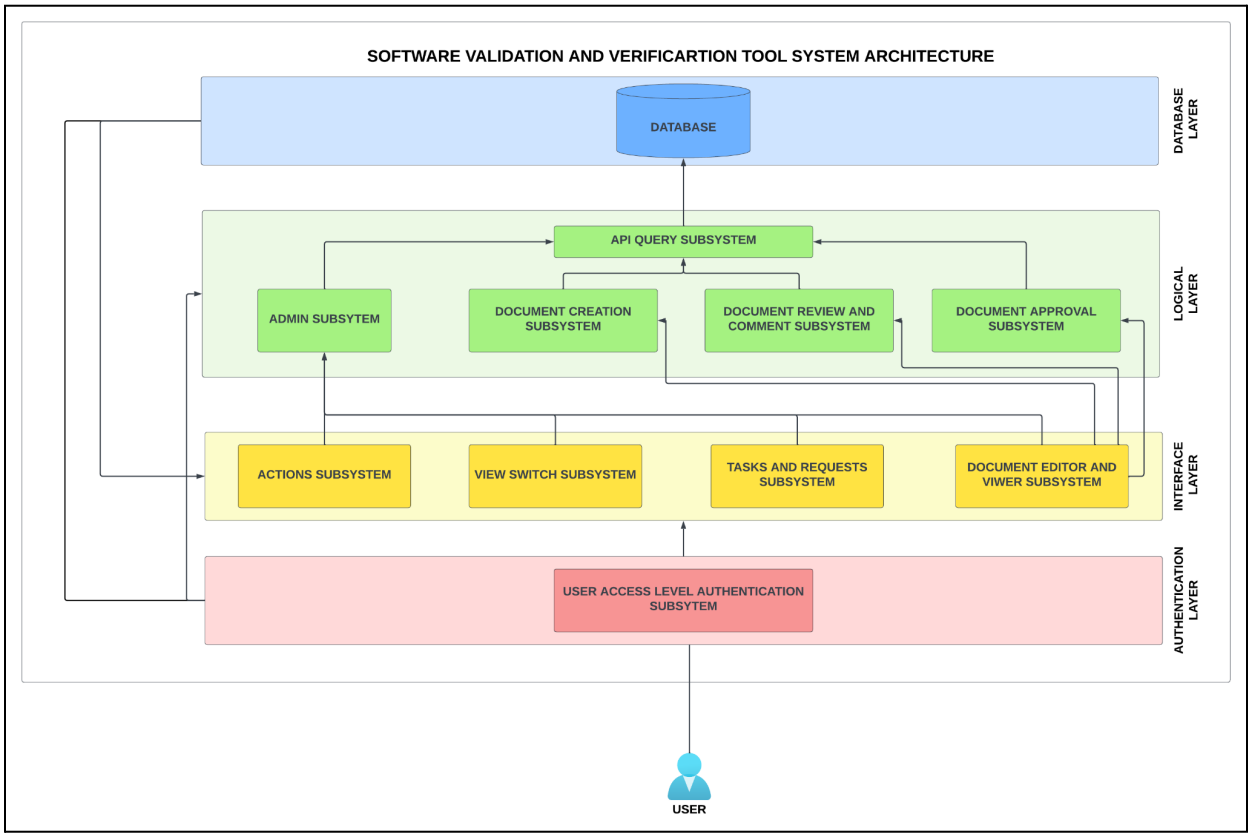

Figure 1: System Architecture of the V&V tool

The user of the V&V tool interacts at the very beginning with the first layer in the architecture which is the Authentication Layer. This is represented in the color red in the figure above. The authentication layer interacts with the database to ensure that the correct access rights are given to the user who's attempting to log in. The next layer in the architecture is the Interface layer.

The user of the system interacts with the interface layer through the access right layer to ensure that only authorized sections are accessed. This is followed by the logical layer where the subsystems work to fetch, modify, and upload the data being manipulated through the interface layer using API connections to the database. The final layer is the database layer where all the information concerned with the V&V tool is suitably stored as explained in detail in section four, Data Design. Each of the constituents of these layers are described in detail in the section below.

3.2 Architecture Description

This section presents the description of the architectural design of the Validation and Verification (V&V) Tool, framed from an object oriented perspective. The architecture is segmented into four primary layers, each meticulously detailed in subsections 3.2.1. To 3.2.4. Additionally, subsection 3.2.5 features a sequence diagram, offering a visual and in-depth depiction of the process flow.

3.2.1 Authentication Layer

The Authentication Layer is the foundation of security within the V&V tool. It ensures that every interaction with the system is conducted by an authenticated and authorized user. This layer upholds the integrity and confidentiality of the system by verifying user credentials before granting access to the system's features and data. This layer constitutes the following subsystem -

User Access Level Authentication Subsystem - This subsystem is responsible for validating user’s credentials. It ensures that only users with valid credentials are granted access and access level is granted accordingly. Two factor authentication mechanism would be used to protect the system from unauthorized access.

3.2.2 Interface Layer

The Interface Layer is the user's entry point into the V&V tool, providing a user-friendly environment for interacting with the system. It is designed to offer a straightforward and responsive experience, allowing users to efficiently navigate through tasks based on their roles and permissions. This layer constitutes the following subsystems -

Actions Subsystem - The Actions Subsystem is tailored to present to the user the actions that they can perform in accordance to the specific roles and responsibilities of users within the V&V

tool. It provides different options and tools, each designed to match the unique workflow requirements of the creator, reviewer, and administrator. A creator will have access to features for drafting and submitting documents, a reviewer will see options for commenting and suggesting revisions, and an administrator will be presented with controls for overseeing the entire document lifecycle and user permissions. This role-based design ensures that each user encounters a personalized interface that aligns with their tasks, streamlining their interactions and enhancing overall system efficiency.

View Switch Subsystem - The View Switch Subsystem in the V&V tool is meticulously designed to support users with multifunctional roles, offering a seamless mechanism for toggling between different role-based interfaces. When a user is assigned dual roles—such as a reviewer and an administrator—they can effortlessly transition from conducting document reviews to performing administrative duties. This subsystem enhances productivity by consolidating access within a single user session, thereby facilitating a fluid shift in roles without the need for multiple logins or disjointed user experiences.

Tasks and Requests Subsystem - The Tasks and Requests Subsystem functions as the central organizational hub for users within the V&V tool, presenting a comprehensive list of pending actions and administrative requests in need of attention. This subsystem organizes tasks in a manner that supports users in managing their workflow efficiently. Administrators benefit from a unified overview of system-wide requests, allowing them to effectively handle user-generated inquiries.

Document Editor and Viewer Subsystem - The Document Editor and Viewer Subsystem is a sophisticated workspace within the V&V tool, designed for creating, editing, reviewing, and viewing documents. It's equipped with a rich set of features that cater to the needs of content creation and document management, including text editing, formatting tools, commenting capabilities, and version control. This subsystem is engineered to support collaborative work, allowing multiple users to work on a document simultaneously while tracking changes and comments in real-time. It aims to provide a comprehensive and integrated environment for document handling, from the initial draft to the final review and approval stages.

3.2.3 Logical Layer

The Logical Layer is the engine of the V&V tool, taking care of all the essential tasks that keep the tool running smoothly. It's where all the main tasks are done, like managing users, working with documents, and getting approvals. This layer ensures that the user's input through the interface is effectively turned into action, and that data is properly stored and retrieved. It's focused on making the tool reliable and easy to use, without users having to worry about the complex processes going on under the hood.

Admin Subsystem - The Admin Subsystem facilitates comprehensive user management capabilities, such as adding new users, configuring roles, updating permissions, and ensuring secure removal of users from the system. This subsystem also handles a variety of administrative requests and tasks, serving as the backbone for governance and control within the V&V environment. It's instrumental in enforcing policy compliance, managing access levels, and ensuring that each user's interaction with the tool is aligned with their organizational role and the security protocols in place.

Document Creation Subsystem - This subsystem provides a robust platform equipped with text editors, formatting tools, and multimedia integration capabilities. Additionally, this subsystem is adept at handling concurrent document edits, maintaining version control, and preventing data loss or conflicts through sophisticated conflict resolution mechanisms. It ensures that multiple users can collaboratively and effectively work on documents in real-time, promoting a seamless creative process.

Document Review and Comment Subsystem - This subsystem provides a structured environment for the review process, allowing reviewer users to scrutinize documents, suggest changes, and engage in discussions through commenting features. This subsystem ensures that software artifacts are meticulously reviewed against the 4C's—completeness, correctness, consistency, and clarity—before they are escalated for higher management approval. This subsystem ensures that documents meet the set standards before moving to the next phase of the approval process.

Document Approval Subsystem - This subsystem helps manage final approval workflow for a document in the V&V tool lifecycle. It provides functionalities that allow a document to be approved by a reviewer and escalated to upper level management for final approvals and implementation.

API Query Subsystem - The API Query Subsystem links the V&V tool and the aforementioned components of the logical layer with the database, playing a key role in the smooth flow of data. It turns the tasks performed by users into API requests, retrieving, modifying, or saving information in the database as required. This subsystem is essential for making sure that data moves swiftly and safely across the system, in line with user needs, thereby supporting the tool's dependable performance.

3.2.4 Database Layer

The Database Layer is an integral part of the V&V tool, where all data is stored and managed. It holds vital information such as user profiles, document archives, edit histories, comments, approval records, etc. The Database, which is the only subsystem in this layer, is optimized for high availability, ensuring that users have continuous access to the data they need.

3.2.5 Sequence Diagram

The sequence diagram shown in Figure 2 is a clear visual guide to how users and subsystems within the V&V tool interact. It lays out the steps and processes in a straightforward manner, detailing each interaction across the tool's architecture. The sequence diagram for the tool's architecture outlines the user interactions that begin with the user entering login details and an OTP into the Interface, triggering the authentication sequence. Following this, the User Access Level Authentication Subsystem checks these credentials with the Database via the API Query Subsystem. The result, successful or not, is relayed back to the Interface and the user. With successful login, the user moves on to the Interface Layer, where the Actions Subsystem provides access to the tool's features appropriate to the user’s role. For users with multiple roles, the View Switch Subsystem allows easy switching between different interfaces to suit the role in use.

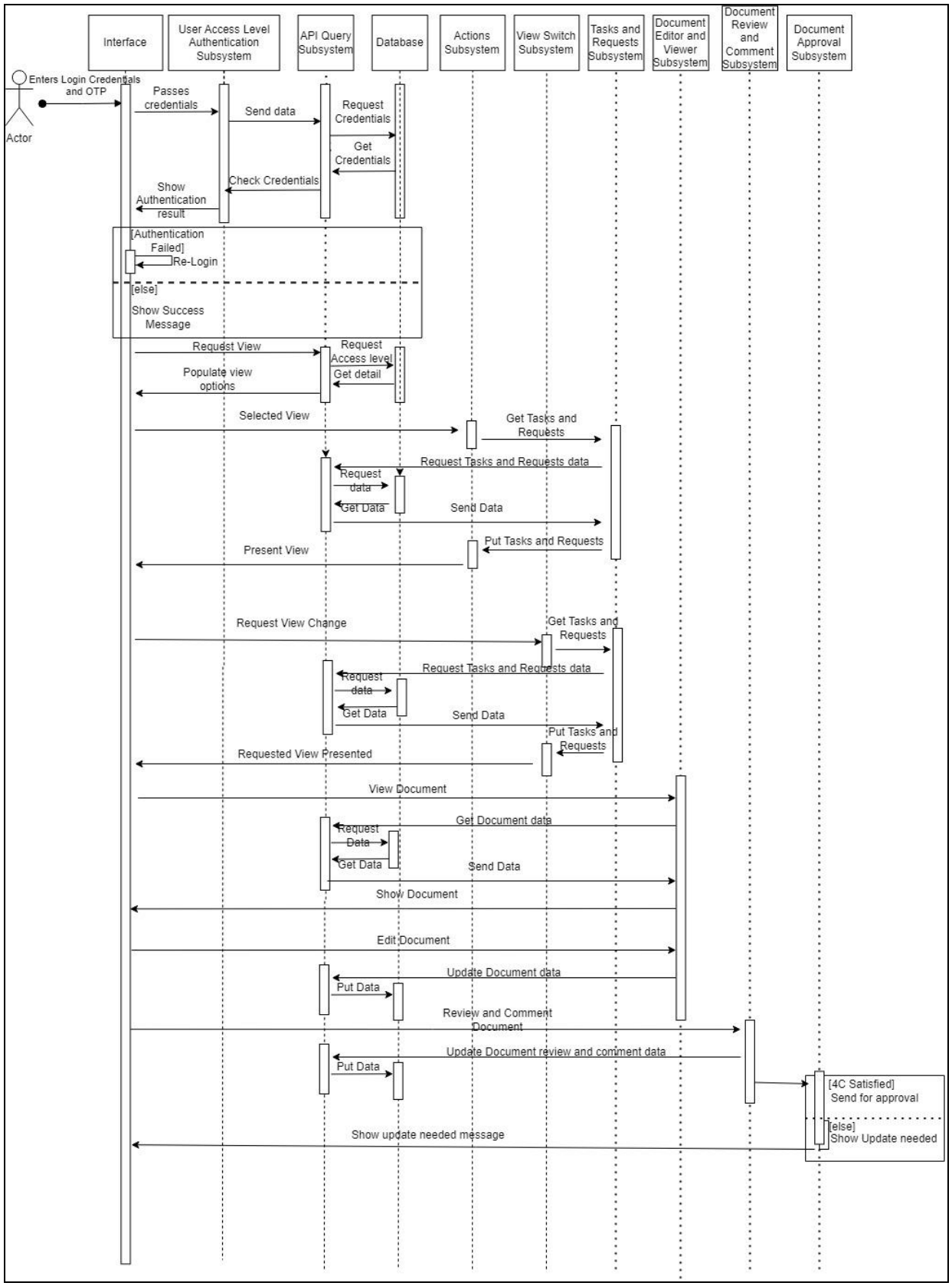

Figure 2. Sequence Diagram for V&V tool

The Tasks and Requests Subsystem in the above diagram, presents the user with their pending tasks for efficient management and workflow tracking. Further, the document workflow component involves the Document Editor and Viewer Subsystem for making changes to documents, the Document Review and Comment Subsystem for feedback, and the Document Approval Subsystem to ensure all documents are properly checked before moving on to the final approval stage. The process culminates with the Document Approval Subsystem, which either approves documents for final release or marks them for further updates.

3.3 Design Rationale

The architecture presented in sections 3.1 through 3.2 above was designed keeping in mind a modular, object oriented approach. It constitutes distinct subsystems for tasks such as authentication and authorization, user interface, document management, API querying etc. The interactions amongst these subsystems make possible the functionality achieved by the tool. The architecture presented was chosen because of its modular nature which can handle scalability. The modularity of the design ensures ease of maintenance and can potentially support extensibility. Furthermore, each subsystem has well-defined, distinct responsibilities that make it easier to modify them without affecting the system's functionality as a whole.

Additionally, security and access control layers that have been added to the architecture, ensures that users can only access the documents and functionalities that they are supposed to, thereby preventing accidental changes and unauthorized access. Furthermore, the logic layer has subsystems that handle concurrency. This introduction of Simultaneous Access Control Module addresses the critical issue of concurrent access to documents. It enables the system to manage conflicts during simultaneous editing, enhancing the overall user experience.

Thus, it can be said that the proposed architecture also addresses the following critical issues -

- **Security -** The architecture promoted security by implementing authentication and access control mechanisms. User access is tightly controlled, reducing the risk of unauthorized access to sensitive documents.

- **Concurrent Access -** Simultaneous Access Control Module ensures that conflicts are appropriately managed during simultaneous editing.
- **Scalability -** New features or subsystems can be added without disrupting existing functionalities, allowing the system to grow as needed.

However, while choosing the present architecture, some trade-offs had to be considered such as the increased complexity of design due to modularity of subsystems as well as the initial development time versus future extensibility. In both the cases, the modularity trade-off is deemed acceptable as the benefits, such as scalability and quick troubleshooting, outweigh the added complexity and additional development time. In summary, the chosen design addresses important concerns while accepting trade-offs by striking a balance between modularity, security, and scalability. It offers a strong framework for a software validation and verification system that can adapt to changing needs.

4. Data Design

4.1 Data Description for V&V Tool

In the V&V tool's data design, the primary structure revolves around a NoSQL database that provides a schema-less approach to manage diverse data types, ensuring flexibility for evolving data requirements.

- **Software Artifacts/ Documents:** They are stored as documents in the database, with each artifact encompassing a unique ID, content, metadata, and version history. The document model allows for the embedding of complex structures, such as revision histories and associated reviews.

- **Users:** Information on users is structured to include credentials, role assignments, and personal settings. The model is designed to efficiently support operations related to authentication and authorization, reflecting each user's access level and role within the system.

- **Reviews and Approvals:** Stored in relation with document, requirement, design, and test ID, reviews contain comments and metadata linked to artifacts, facilitating a comprehensive audit trail for each review cycle. Approval records track decisions made, tagged with timestamps and approver identifiers, critical for maintaining a transparent approval workflow.

- **Tasks:** Represented as individual entries, tasks carry details like status, assigned user, priority, and due dates. The database's ability to handle varying attributes and relationships without predefined schemas aids in managing the dynamic nature of task data.

The utilization of JSON within this framework serves as the standard for data interchange between system components and external interfaces using API.

4.2 Data Flow Diagram

The image in figure 3 below shows the data flow diagram (DFD) of the V&V tool.

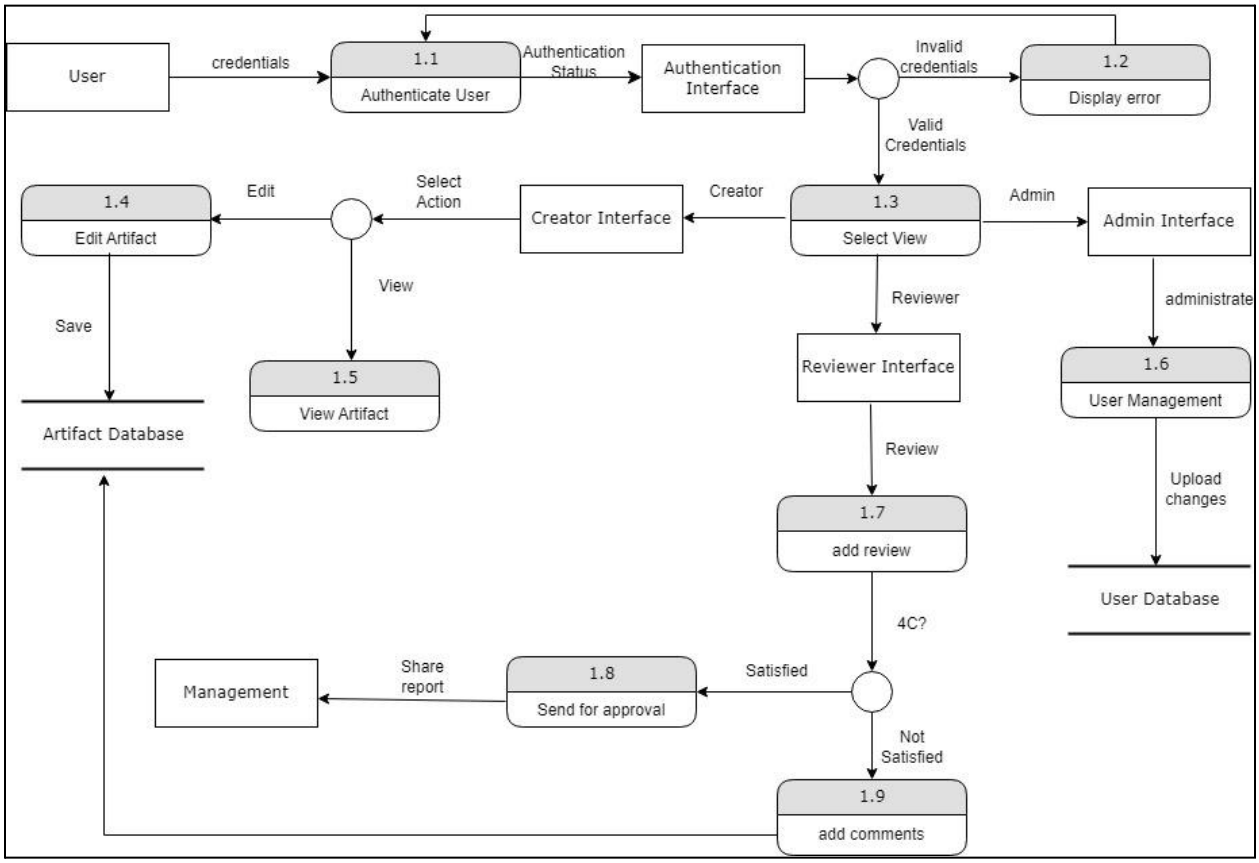

Figure 3: Data Flow Diagram for V&V tool

Based on the DFD provided in Figure 3, the structured explanation of the data flow within the V&V tool is provided below:

User Authentication:

- Users provide their credentials to the Authenticate User process (1.1).
- If credentials are valid, the Authentication Status is confirmed, and users proceed to the Creator Interface.
- Invalid credentials trigger an error message displayed through the Authentication Interface (1.2).

Interface Selection and Artifact Management:

- The Creator Interface allows users to Select Action for artifact management (1.3).

- Users can Edit Artifact (1.4), which involves making changes and saving updates to the Artifact Database.
- Users can also View Artifact (1.5), retrieving information from the Artifact Database.

Review and Approval Process:

- The Reviewer Interface presents options to Review (1.7), where users can add a review to the artifact.
- If the review satisfies the 4C's criteria, the artifact is sent for approval (1.8); otherwise, additional comments are added (1.9).

Administration and User Management:

- Admins select the Admin Interface to manage user roles and permissions (1.6).
- Changes in user data are uploaded to the User Database.

Management and Reporting:

- The Management process allows for broader actions such as sharing reports, which are generated based on artifact data and reviews.

5. Component Design

In this section, we delve into the detailed design of each component of the Validation and Verification (V&V) Tool. We provide a summary of the functions and methods for each subsystem within the architectural design described in section 3.2.

5.1 Authentication Component

This subsystem is necessary for security, initiating the process by verifying user credentials and OTP for two factor authentication purposes. Upon successful authentication, it grants the user access to the system. If authentication fails, access is denied, effectively preventing unauthorized entry.

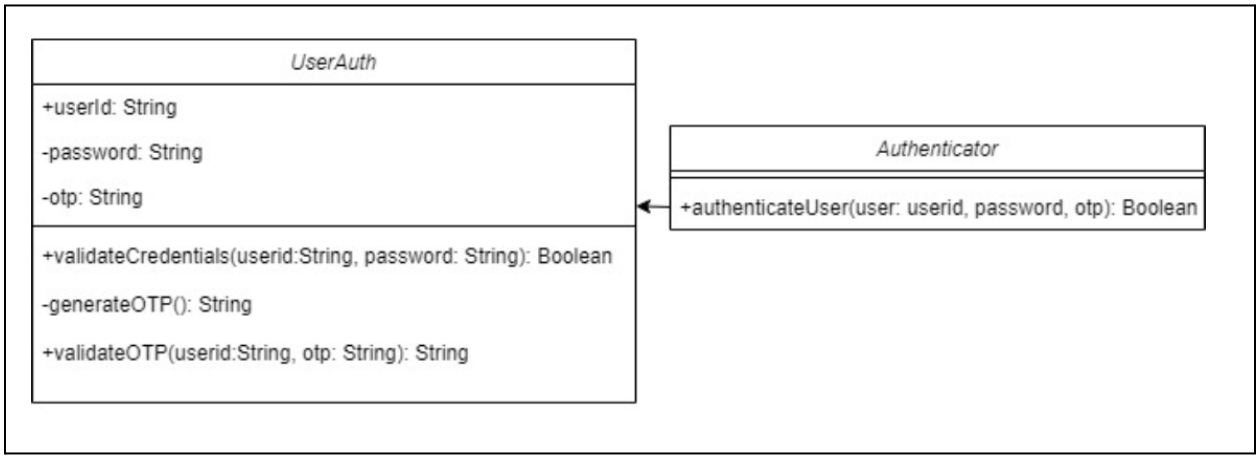

Figure 4: Object Diagram for Authentication Component

UserAuth Class

Attributes:

- **userId:** A unique identifier for the user. It's used to retrieve the user's data during the authentication process.
- **password:** The user's password. Since this is sensitive information it has been kept private.
- **otp:** The one-time password used for two-factor authentication. Similar to the password, this is sensitive information and is kept private.

Functions:

- **validateCredentials:** Validates the provided password against the stored values.

- **generateOTP:** Generates a time-sensitive OTP. This method is typically used internally and might be better set as private unless there's a specific design decision to expose it.

- **validateOTP:** Checks the provided OTP against the generated OTP to confirm the user's identity. This is part of the two-factor authentication process to add an extra layer of security.

Authenticator Class

Attributes:

- No persistent data variables should be stored in the Authenticator class as it's meant to be stateless and only process the current authentication request.

Functions:

- **authenticateUser:** This is the main method used for authenticating a user. It takes the user object along with the password and OTP provided by the user during the login attempt. It then calls the User class's validateCredentials and validateOTP methods to perform authentication. If both checks pass, it returns true, indicating the user is authenticated; otherwise, it returns false.

5.2 Interface Component

The Interface Component of the V&V tool customizes the user interface to fit each user's role, showing them only the tools and features they need. It lets users with more than one role switch views easily. The component helps users keep track of their tasks and manage documents effectively, providing straightforward functions for editing and viewing documents as needed.

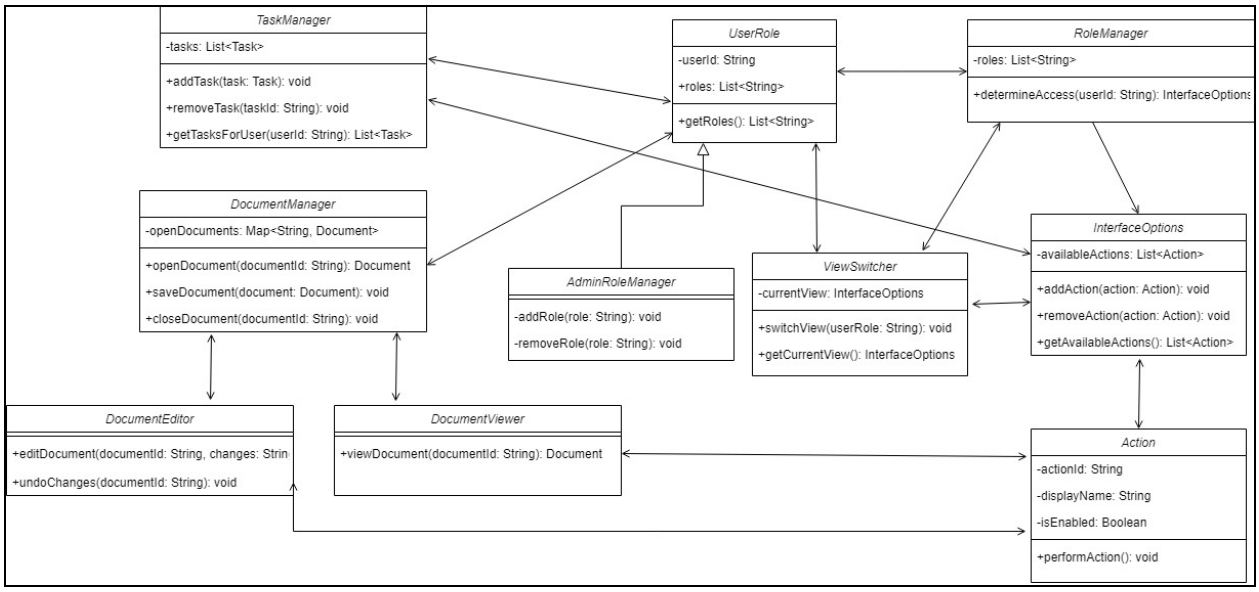

Figure 5: Class Diagram for Interface Component

UserRole Class

Attributes:

- **userId**: Uniquely identifies a user within the system, typically used to associate a user with their role(s).

Methods:

- **getRoles**: Retrieves the list of roles assigned to the user.

AdminRoleManager Class

Methods:

- **addRole**: Adds a new role to the user's roles list. This could be used by administrators to grant additional permissions to a user.
- **removeRole**: Removes an existing role from the user's roles list. This could be used to revoke permissions from a user.

RoleManager Class

Attributes:

- **roles**: Holds a collection of all possible roles that can be assigned to users within the system.

Methods:

- **determineAccess**: Determines the set of InterfaceOptions a user has based on their roles. It uses the UserRole.getRoles() method to get the user's roles and then generates an InterfaceOptions object that contains only the actions the user is allowed to perform.

InterfaceOptions Class

Attributes:

- **availableActions**: A collection of Action objects that represents the actions available in the user interface.

Methods:

- **addAction**: Adds an Action to the availableActions list, making it available in the user interface.
- **removeAction**: Removes an Action from the availableActions list, making it unavailable in the user interface.
- **getAvailableActions**: Returns the list of currently available actions that the user can perform in the interface.

Action Class
Attributes:
- **actionId**: A unique identifier for an action, used to reference and manage actions programmatically.
- **displayName**: The human-readable name of the action, which could be displayed in the user interface.
- **isEnabled**: A flag indicating whether the action is currently enabled and can be performed by the user.

Methods:
- **performAction**: Executes the action's behavior. The specifics of what this method does would depend on the particular action it represents.

ViewSwitcher Class
Attributes:
- **currentView**: Stores the current interface options that the user is interacting with.

Methods:
- **switchView**: Updates currentView to reflect the interface options associated with a different role.
- **getCurrentView**: Returns the interface options currently being displayed to the user.

TaskManager Class
Attributes:
- **tasks**: A list that holds all the tasks and requests relevant to the user.

Methods:
- **addTask**: Includes a new task in the tasks list.
- **removeTask**: Deletes an existing task from the tasks list.
- **getTasksForUser**: Retrieves a list of tasks that are assigned to or created by a specific user.

DocumentManager Class
Attributes:
- **openDocuments**: A collection that manages documents that are currently open in the system.

Methods:
- **openDocument**: Retrieves a document for viewing or editing and adds it to the openDocuments.
- **saveDocument**: Saves changes made to a document back to the storage system.
- **closeDocument**: Closes a document and removes it from openDocuments.

DocumentEditor Class
Methods:

- **editDocument**: Applies changes to a document's content.
- **undoChanges**: Reverts the document to a previous state before the last set of changes.

DocumentViewer Class
Methods:

- **viewDocument**: Retrieves a document's content for the purpose of viewing.

5.3 Logical Layer Component

The Logical Layer Component handles the creation, updating, and reviewing of documents. It assigns IDs to new documents and updates existing ones. Users can add comments to documents for review. After review, documents are checked against the 4C's criteria before being approved or rejected, updating their status in the system.

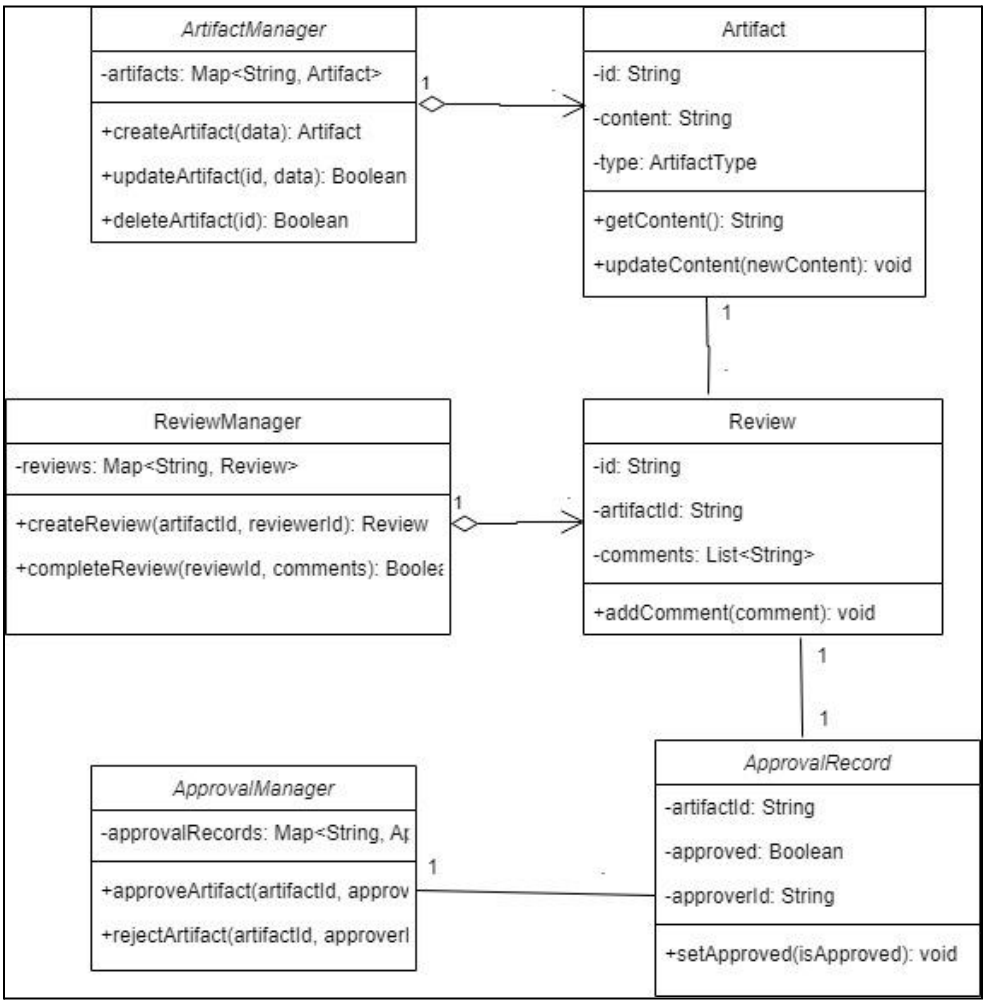

Figure 6: Class Diagram for Logical Component

ArtifactManager Class

Attributes:

- **artifacts:** A collection that maintains a mapping of unique identifiers to artifacts within the system.

Methods:

- **createArtifact:** Generates a new artifact based on provided data and adds it to the collection.
- **updateArtifact**: Applies updates to a selected artifact identified by its ID.
- **deleteArtifact**: Removes the specified artifact from the collection.

Artifact Class

Attributes:

- **id**: A unique identifier assigned to each artifact.
- **content**: The main data or body of the artifact.
- **type**: Categorizes the artifact, such as a document or code.

Methods:

- **getContent**: Retrieves the content of the artifact.
- **updateContent**: Replaces the artifact's existing content with new content.

ReviewManager Class

Attributes:

- **reviews**: A repository that stores reviews, linked to artifacts by their IDs.

Methods:

- **createReview**: Starts the review process for a given artifact.
- **completeReview**: Finalizes the review, attaching comments to the artifact.

Review Class

Attributes:

- **id**: Distinguishes each review with a unique identifier.
- **artifactId**: Connects the review to a specific artifact.
- **comments**: Accumulates feedback from reviewers.

Methods:

- **addComment**: Appends a new comment to the review.

ApprovalManager Class

Attributes:

- **approvalRecords**: Tracks the approval status of artifacts across the system.

Methods:

- **approveArtifact**: Marks an artifact as approved in the records.
- **rejectArtifact**: Marks an artifact as not approved, indicating rejection.

ApprovalRecord Class

Attributes:

- **artifactId**: Associates the record with an artifact.
- **approved**: Indicates the current approval status of the artifact.
- **approverId**: Identifies the approver responsible for the artifact's status.

Methods:

- **setApproved**: Updates the approval status of the linked artifact.

6. Human Interface Design

In this section , we present the user interface for the V&V tool which will be powered by the modules in the interfacelayer as presented in section 3 of this design document. The sections that follow present the details of the functionalitiesof different interfaces that are to be present int the tool.

6.1 User Login and Access Retrieval

The user interface for the V&V verification tool will begin with a standard logion screen. The design and attributes of this screen will remain uniform across all users. A mock-up illustrating an example of a potential design is shown in the figure below.

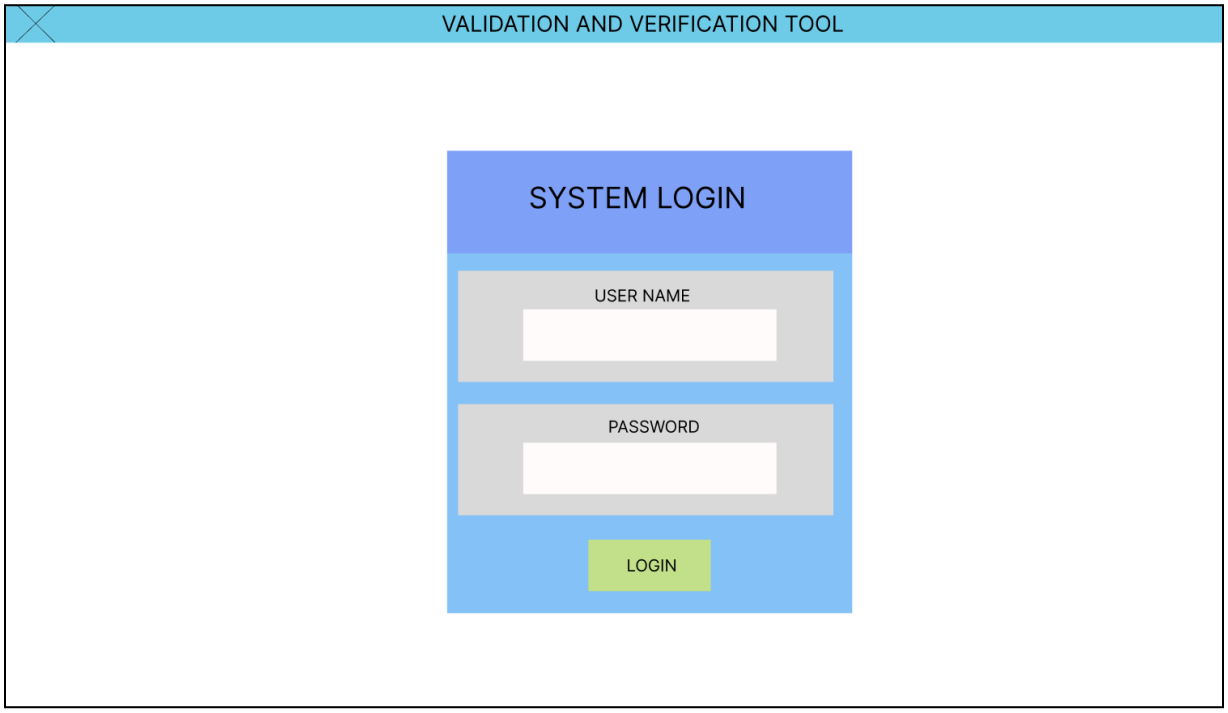

Figure 7: Initial Login Screen for the V&V Tool

The user will be prompted to enter their user name and password. Pressing the login button grants the user access if their credentials are verified. In case, the entered credentials do not match the database contents, a second screen shall be displayed. A mockup of this is presented in the figure below.

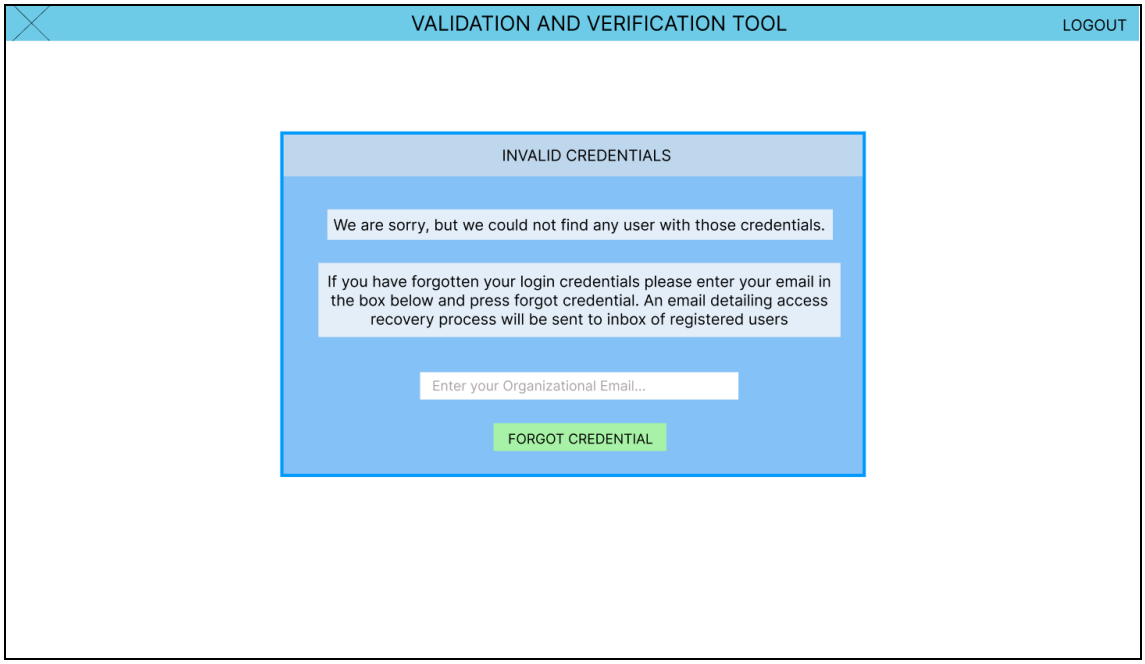

Figure 8: Error While Logging In Screen for the V&V Tool

Once the user has access to the tool, a screen similar to the one shown in the figure below will be displayed. The select view menu will allow the user to switch between types of views and the corresponding information displayed under them on the homepage. This feature will allow a user with multiple access permissions to quickly switch between views.

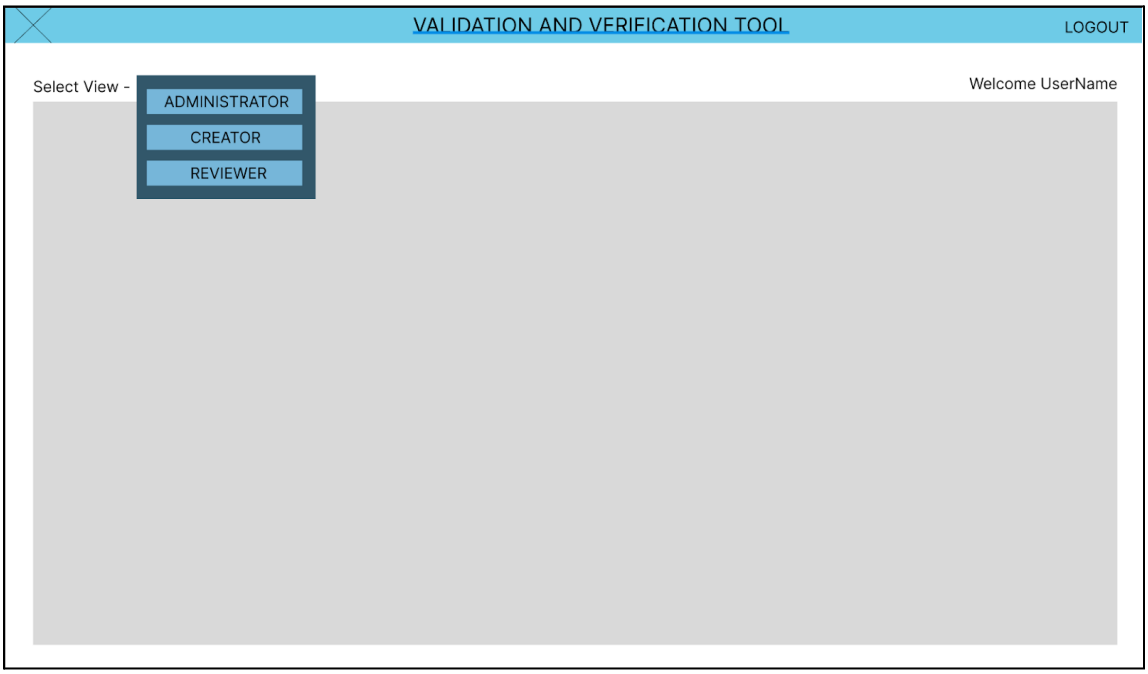

Figure 9: Standard View After Login for the V&V Tool

6.2 User Specific Views

Once a user has logged into the system, based on the access rights they have or the selected view in case of multiple access rights, different kinds of information will be visible on the home screen. For a user with administrator access right, the view in the figure below is present on the home screen.

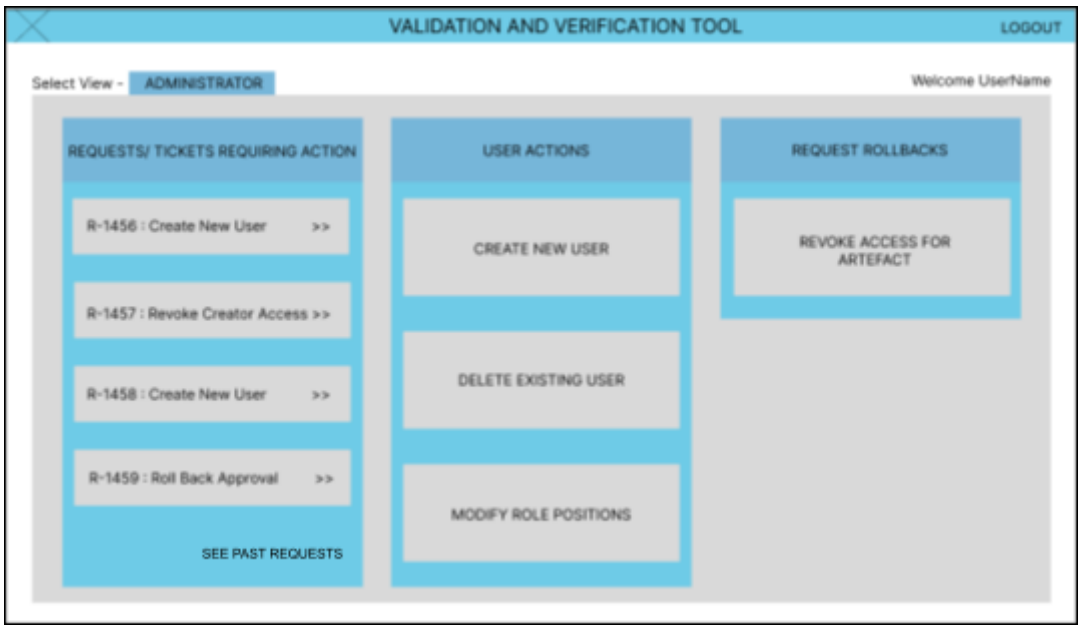

Figure 10:Administrator View After Login for the V&V Tool

The Administrator user can use the Request Rollbacks panel to select the “Revoke Access for Artefact” option that will allow them to undo the approval a reviewer user might have granted to a document, given that a request has been escalated for the same. The “User Actions” panel will be used to perform tasks such as create a new user and generate access credentials, delete an existing user, and modify access rights. Buttons for availing these options will be provided on the panel. The third panel is the “Requests/ Tickets Requiring Action” panel where the active tickets that need to be resolved by the administrator show up. Clicking on the “See Past Requests” will let the user see previously approved requests by administrator.

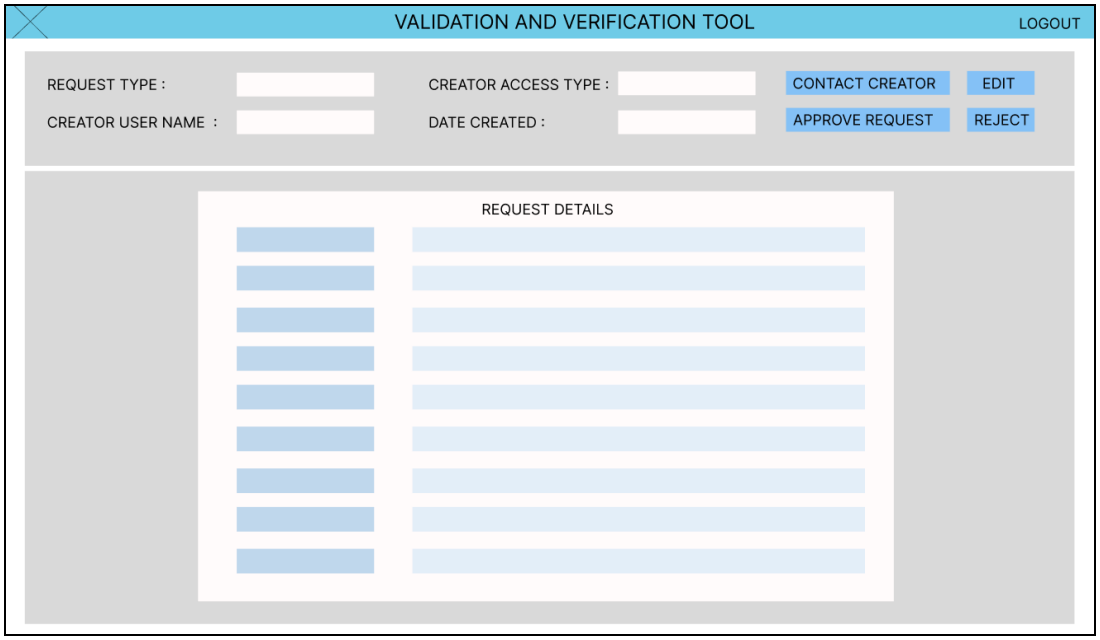

Figure 11: Request Dialog in the Administrator View

The figure above shows a sample mockup of a request when opened by the administrator user. Information about the type of request - user addition, access revocation, document rollback, etc.- will be presented along with the name of the request creator, the date of request creation etc. In addition to this, buttons will be provided so that the administrator can take suitable actions given a request.

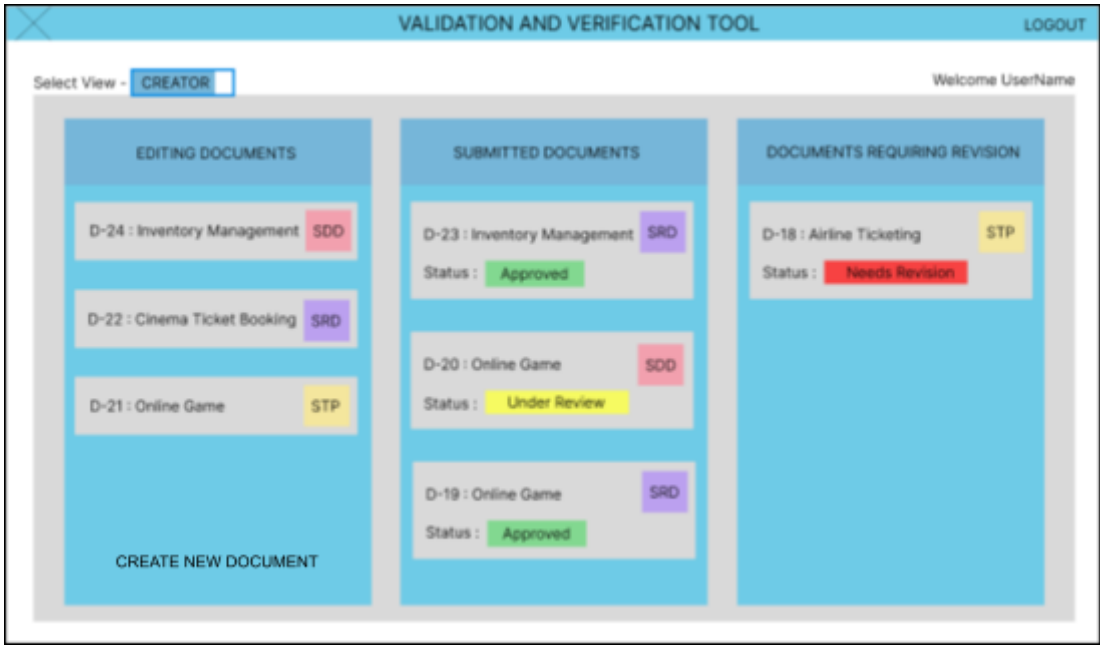

Figure 12: Standard View After Login for the V&V Tool

The figure above shows the creator view of the application after login to the V&V tool. There will be three panel in this view - Editing Documents, Submitted Documents, and Documents Requiring Revision. The Editing Documents Panel will show the documents created by the user along with colored tags to signify the type of document. There can be three main type of documents in our V&V tool according to system requirements. These are the - SRD, SDD, and STP type of documents. Further, the "Create New Document" option at the bottom will allow the user to create a new document.

The next panel is the Submitted Documents panels that will the submitted documents along with their status of approval. The user may click on these documents to open a view only mode. The third panel is the "Documents Requiring Revision" panel which will show the documents retuirned back to the dreator for review. These documents can be clicked on and edited to make suggested changes.

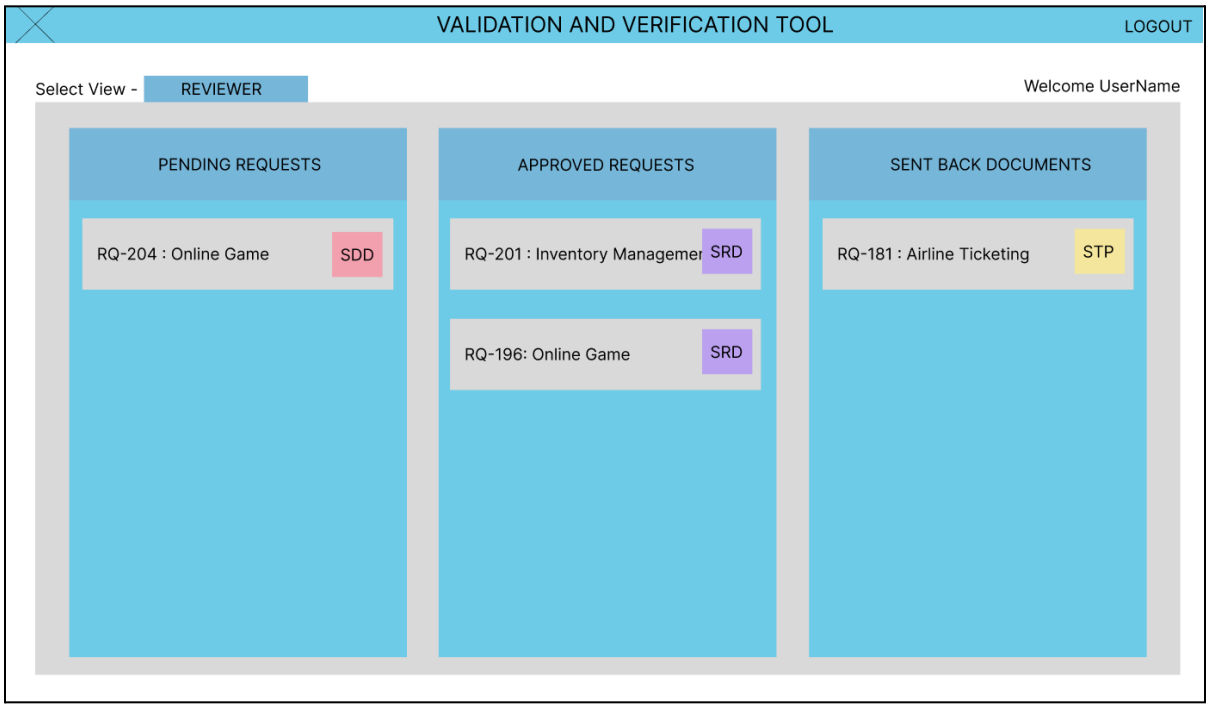

Figure 13: Standard View After Login for the V&V Tool

The figure above shows the Reviewer user view if the V&V tool homescreen after login. This view also constitutes three panels. The first "Pending Request" panel will show all the request

ready for the user's review and comment. The "Approved Requests" panel will show all the requests that have been approved by the user and can be clicked on to open a view only mode version of the document. The "Sent Back" panel will show all the documents that have been sent back to review to the realtor. The user can open these to make additional comments as well.

In case of all the users, the top right corner has a logout option using which the users can exit the V&V tool. The section that follows shows the user interface for the three main types of documents for the V&V tool.

6.2 Document Type Views

The first main type of document for the V&V tool is the SRD or the Siftwarte Requirements Document. The figure below shows the user view of this type of document when opened in our tool. The panel at the top of the screen will show details of the opened document such as the name of the document, the project manager details, the document ID, and the type, etc. Further, the panel will also contain buttons to perform actions like submit the document, save and close, message assigned reviewer (in case of revisions), add comments, etc.

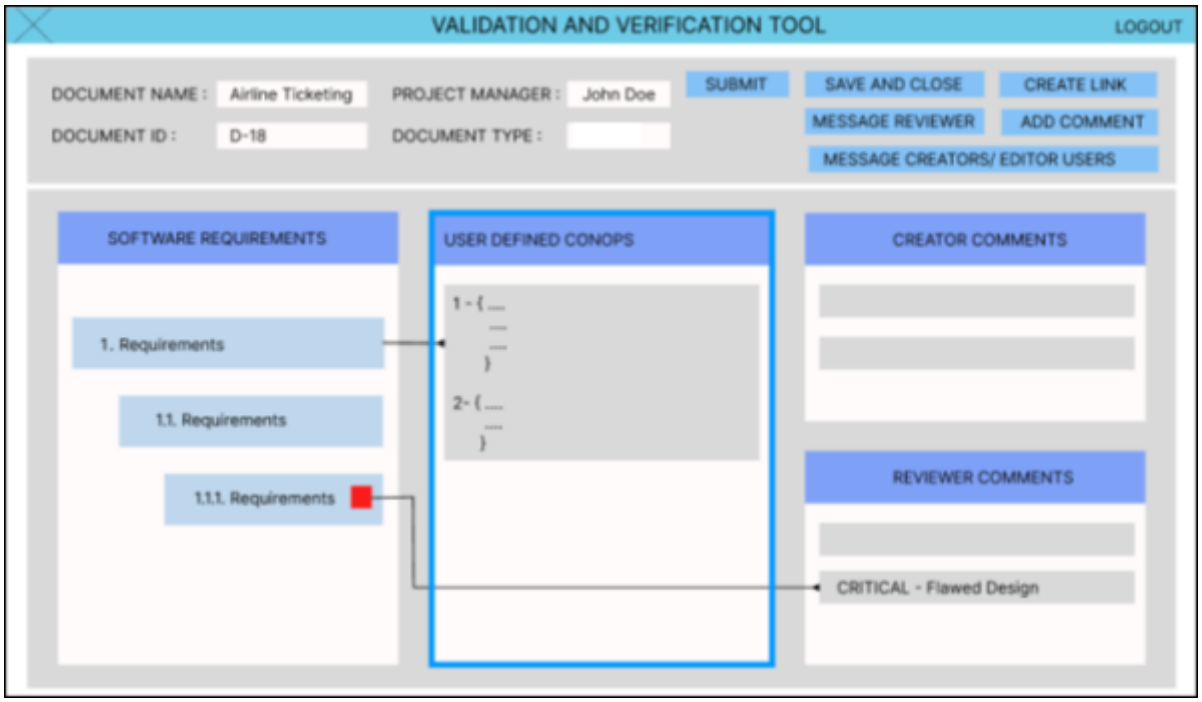

Figure 14: Software Requirements Document View for the Creator User

The create link option will allow the users to establish a link between user defined CONOPS and the entered requirement built on top of the requirement. Additionally, messages can also be sent to other creators for communication purpose. The “Creator Comments” panel will show all the comments made by creator users for documentation or note taking purposes. Further, the “Reviewer Comments” panel will show the comments made by the reviewer for submitted or sent back documents. The color of the comment pointer shall indicate the severity of the type of issue with the particular section. Table 2 below tabulates the different levels of findings categories and the associated suggested colors of their pointers in the user interface.

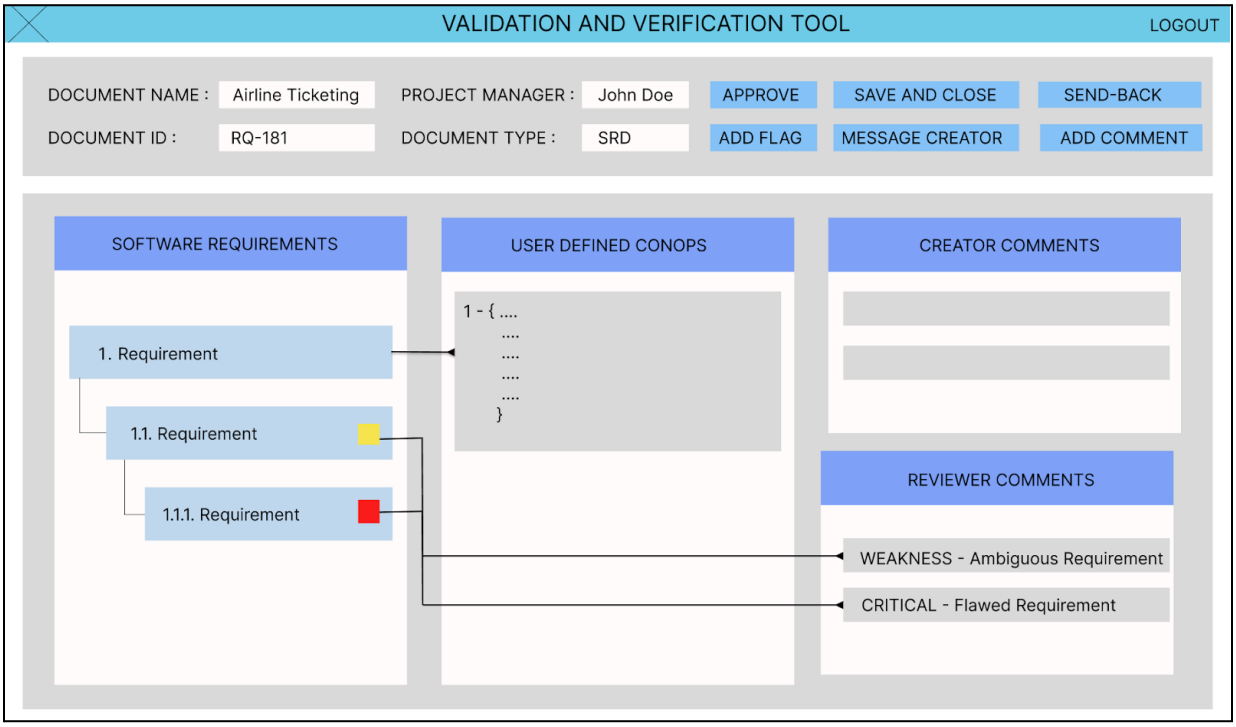

Figure 15: Software Requirements Document View for the Editor User

The figure above shows the Editors view of the document. Most items are the same as the creator viewer with the exception of a few additional buttons added onto the top panel to give user specific functionality. The Approve button will allow the Reviewer to submit the approval for this document if no findings are identified. The send-back option will allow the Reviewer to send the document back to the creator after identifying and flagging the findings. Creators can also be messaged for communication via the “Message Creator” option and comments can be added for clarity on findings.

Table 2 : Finding Categories

CATEGORY	FLAG	DESCRIPTION
Critical		The requirement is flawed and needs major restatement and correction before design can proceed.
Non-critical		The requirement is flawed but can be corrected with minor effort.
Weakness		The requirement is ambiguous, incomplete, or incoherent.
Missing		A requirement implied by other requirements is missing.
Clarification		Additional explanation is required.

The figure below is a mockup of the software design document view for the V&V tool for creator users. The functionality, panel and buttons will all remain the same as in the case of the software requirements document view described in detail above. However, instead of establishing linkages between a user defined CONOPS and entered SRD details, the creator user will be able to establish links between the Software Requirements Document or the approved SRD and the SDD items entered.

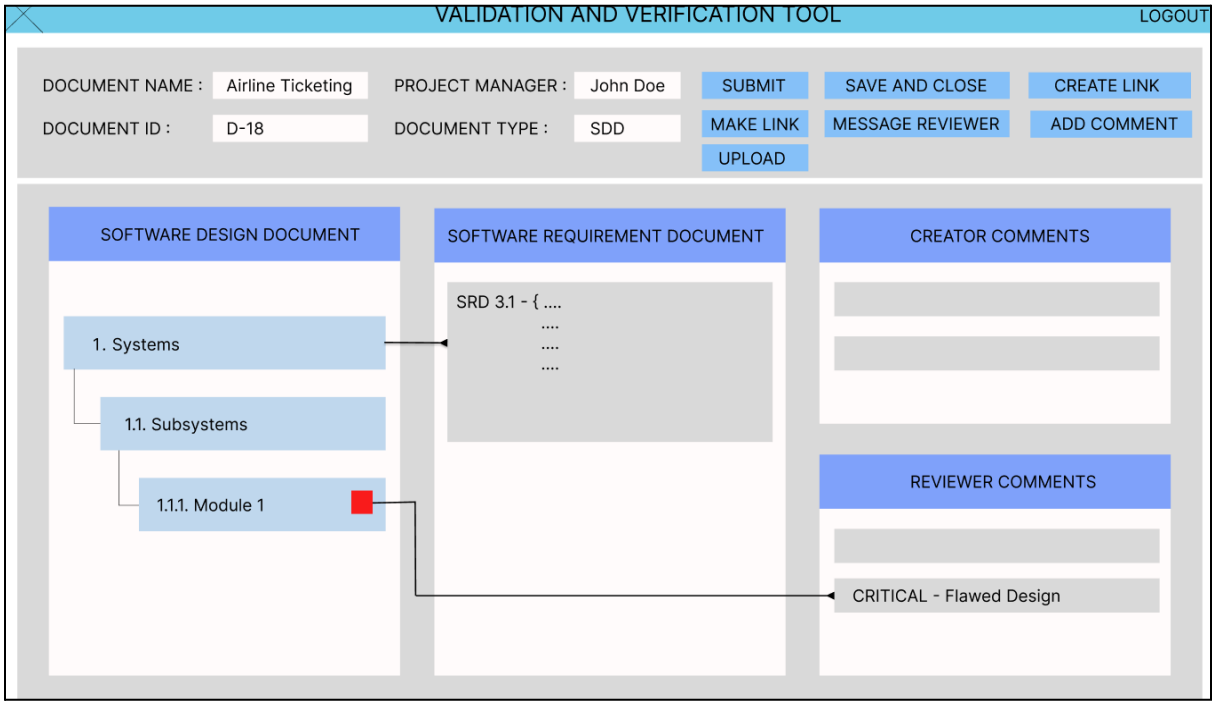

Figure 16: Software Design Document View for the Creator User

The figure below shows the editor's view of the SDD editor for the editor user and just like the case of the SRD view, all the functionalities will remain the same but the linkages will be shown between the Software Requirements Document or the approved SRD and the SDD items entered instead of between CONOPS and requirements entered.

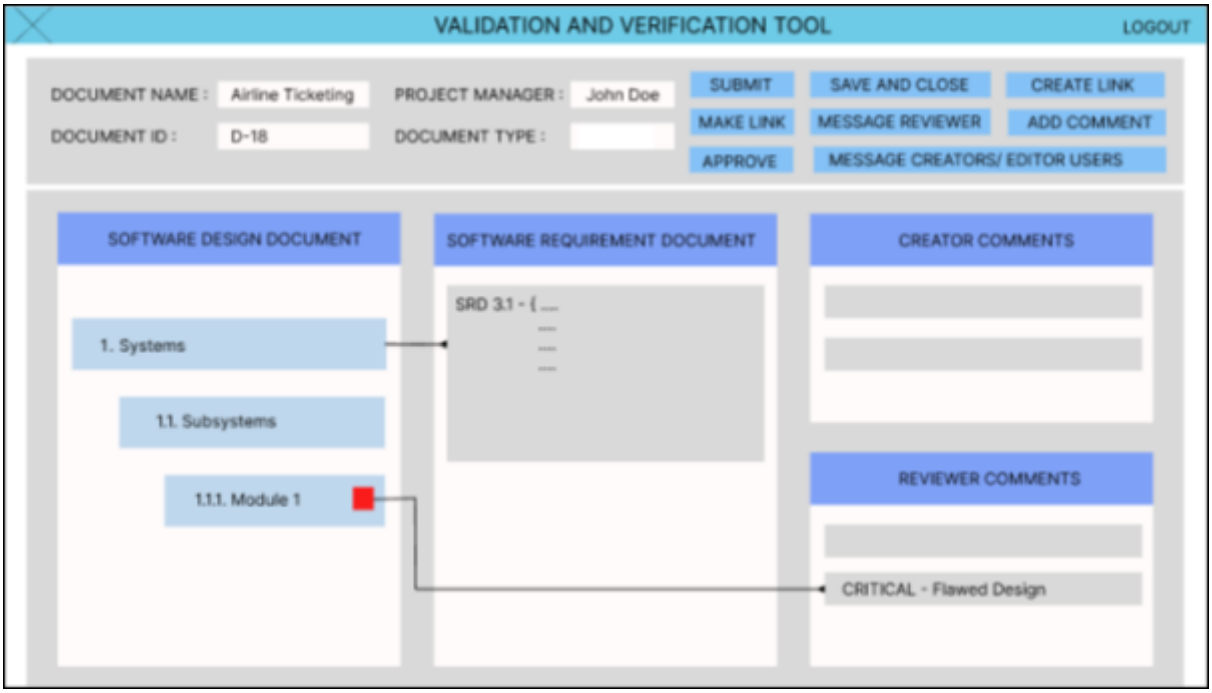

Figure 17: Software Design Document View for the Editor User

The figure below is a mockup of the software test document view for the V&V tool for creator users. Just like in the case of the SDD above, the functionality, panel and buttons will all remain the same as in the case of the software requirements document view described in detail above. However, instead of establishing linkages between a user defined CONOPS and entered SRD details, the creator user will be able to establish links between the Software Design Document or the approved SDD and the STP items entered.

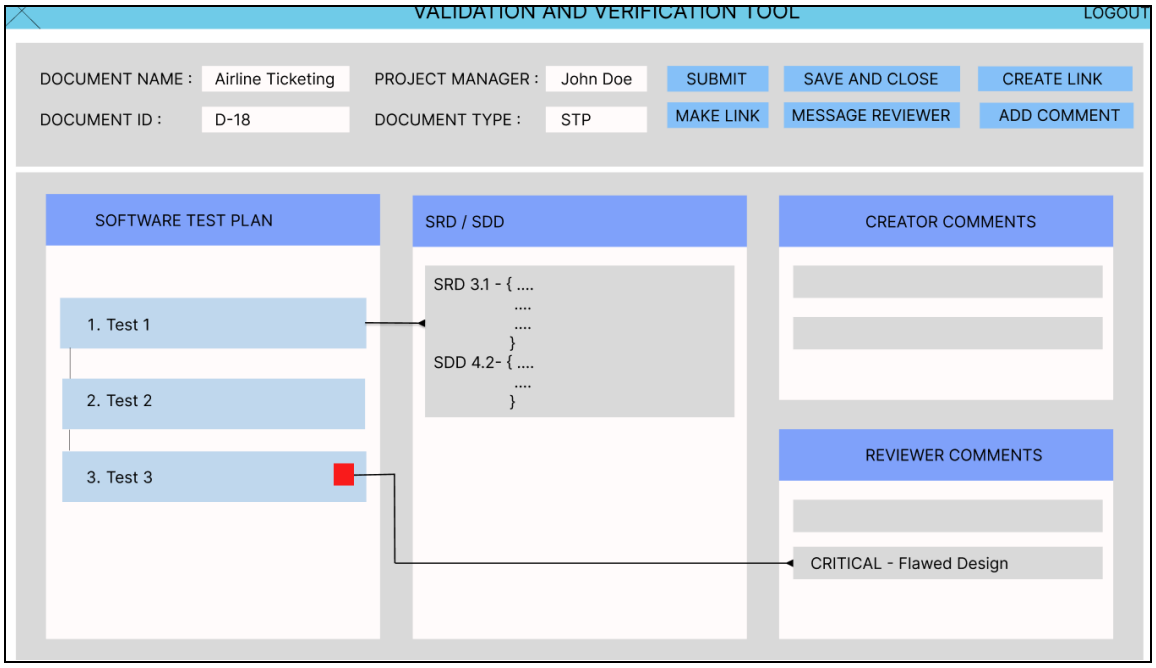

Figure 18: Software Test Document View for the Creator User

The figure below shows the editor's view of the STP editor for the editor user and just like the case of the SRD view, all the functionalities will remain the same but the linkages will be shown between the Software Design Document or the approved SDD and the STP items entered instead of between CONOPS and requirements entered.

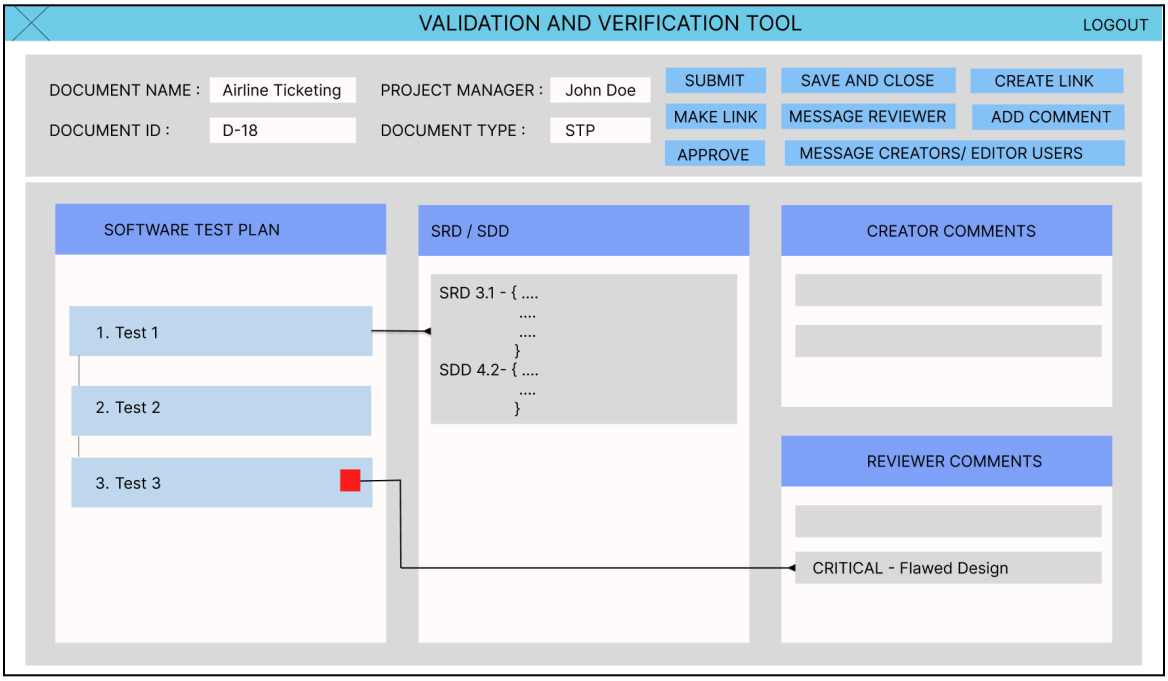

Figure 19: Software Test Document View for the Editor User

7. Issues to be Resolved

The list below identifies some possible areas of improvement that need to be addressed in further detail -

- **Integration with External Systems:** Determine the compatibility of the V&V tool with existing company databases and third-party services and establish secure and reliable API connections for data exchange.
- **User Authentication Mechanism:** Clarify the strategy for encrypted storage and retrieval of user credentials.
- **Scalability and Performance:** Assess the current architecture for scalability to handle an increasing number of users and artifacts. Identify potential performance bottlenecks in artifact processing and database access.
- **Review and Approval Workflow:** Finalize the steps involved in the review and approval process and how they are to be automated and determine the protocol for notification and escalation in the review workflow.