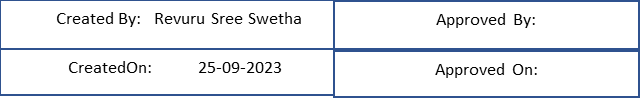


**SOFTWARE DESIGN**

**SPECIFICATION**

DOMAIN NAME:JAVA

PROJECT NAME: IAC ONBOARDING PROCESS



**INDEX**

2.PURPOSE----------------------------------------------------------------3

3.PROJECT SCOPE---------------------------------------------------------3

4.SYSTEM OVERVIEW------------------------------------------------------3

5.DESIGN CONSIDERATIONS----------------------------------------------3

5.1 Requirements-----------------------------------------------------------4

5.2 Assumptions------------------------------------------------------------4

5.3 Dependencies-----------------------------------------------------------4

6.SYSTEM ARCHITECTURE--------------------------------------------------5

6.1 Architectural strategies-------------------------------------------------5

6.2 Structure and Dependencies-------------------------------------------6

7.DETAILED DESCRIPTION OF COMPONENTS----------------------------6

8.INTEGRATIONS-----------------------------------------------------------6

9.APPENDICES--------------------------------------------------------------6

9.1 Appendix-A Detailed Description of Components--------------------7

**1.PURPOSE**

The purpose of the project to automate onboarding workflows and optimize dashboard performances is to enhance operational efficiency, reduce manual intervention, and improve data-driven decision-making within the organization. By automating onboarding processes, we aim to streamline user registration, and approval workflows, reducing processing times and minimizing errors.

**2.PROJECT SCOPE**

The project encompasses the entire lifecycle of onboarding, from user registration to document submission and approval. Dashboard performance enhancements include load testing, data accuracy testing, and real-time updates.

**3.SYSTEM OVERVIEW**

This section will provide an outline of various components and subsystems of IAC Onboarding Process.

The system is designed to address two primary objectives: the automation of user onboarding workflows and the optimization of dashboard performance.

The system overview provides a foundational understanding of how the project addresses the automation of onboarding workflows and dashboard performance, emphasizing its significance in streamlining operations and data-driven decision-making within the organization.

**Benefits:**

* Reduced manual effort and processing time in onboarding workflows.
* Enhanced dashboard responsiveness and data accuracy.
* Improved decision-making through real-time insights.
* Greater operational efficiency and user satisfaction.

**4.DESIGN CONSIDERATIONS**

This section describes the requirements, assumptions and dependencies to be addressed to devise a complete design solution.

4.1 REQURIEMENTS

**1.Hardware Requirements**: RAM 512 or more, Processor: Pentium IV onwards

**2.Software Requirements:** Html, CSS, JavaScript, Spring, Servlets

**3.Functional Requirements:** Allowing new users to register and generate a welcome email which contains the UTM link to track the performance.

**4.Non-Functional Requirements:** Performance

4.2 Assumptions

1.Recieving a correct welcome email after submitting the New Joinee Form.

2. Sufficient development, testing, and training resources are available or can be allocated to the project as needed.

4.3 Dependencies

1. The completion of an ongoing database migration project is a critical dependency. The new database structure must be in place to support the automated workflows and dashboard enhancements.

2. The project depends on the schedule for user training and adoption.

**5.SYSTEM ARCHITECTURE**

The software system architecture refers to the logical organization of a distributed system into software components. It defines how components of a software system are assembled, their relationship and communication between them. It serves as a blueprint for software application and development basis for developer team. An effective architecture serves as the conceptual glue that holds every phase of the project together for all of its stakeholders, enabling agility, time and cost savings, and early identification of design risks.

The Software architecture:

* Defines structure of a system
* Defines behaviour of a system
* Defines component relationship
* Defines communication structure
* Balances stakeholder’s needs
* Influences team structure
* Focuses on significant elements
* Captures early design decisions

Below some important characteristics which are commonly considered are explained.

**Operational Architecture Characteristics:**

* Availability
* Performance
* Reliability
* Low fault tolerance
* Scalability

**Structural Architecture Characteristics:**

* Configurability
* Extensibility
* Supportability
* Portability
* Maintainability

**Cross-Cutting Architecture Characteristics:**

* Accessibility
* Security
* Usability
* Privacy
* Feasibility

6.1 Architectural Strategies

1. User Interface (UI)

2. Database Management

3. Performance Monitoring and Analytics

4. Reporting and Visualization

6.2 Structure and Relationships

**7.Detailed Description of Components**

For detailed description of the components, please refer **Appendix A – Detailed Description of Components**

**8.Integrations**

The project to automate onboarding workflows and optimize dashboard performances necessitates several critical integrations with other applications and tools. Firstly, seamless integration with the organization's existing databases and systems is essential to access user data and enable automated workflow processes.

Furthermore, the project could benefit from integration with monitoring and analytics tools to continuously assess dashboard performance and user interactions.

Finally, integration with reporting and visualization tools is crucial for presenting real-time data insights to stakeholders. These integrations ensure that the project operates cohesively within the existing technology ecosystem, enhancing efficiency and data-driven decision-making.

**9.APPENDICES**

9.1 Appendix-A Detailed Description of Components

|  |  |
| --- | --- |
| **Identification** | Login Screen |
| **Type** | Class/Form/ |
| **Purpose** | Provides user authentication and access control. |
| **Subordinates** | This screen contains links to the following screens:  1.Main Menu Screen  2.New User Account Screen |
| **Dependencies** | The following screen links to this screen  Main Menu Screen |
| **Interfaces** | User login credentials |
| **Resources** | User credentials database, Security module |
| **Processing** | Validates user credentials and grants access. |
| **Data** | The data for Login Screen is the username and password entered by the user. It is validated against a query using database. |

|  |  |
| --- | --- |
| **Identification** | Registration Form |
| **Type** | Class/Form/ |
| **Purpose** | Captures user details for onboarding. |
| **Subordinates** | - |
| **Dependencies** | User Authentication |
| **Interfaces** | User input, API integration |
| **Resources** | User registration database |
| **Processing** | Collects and validates user information. |
| **Data** | User registration data. |

|  |  |
| --- | --- |
| **Identification** | New User Account Screen |
| **Type** | Class/Form/ |
| **Purpose** | Provides user authentication and access control. |
| **Subordinates** | - |
| **Dependencies** | This screen contains links to the following screens:  1.Main Menu Screen  2.Login Screen |
| **Interfaces** | User credentials |
| **Resources** | Database to store the new user credentials. |
| **Processing** | Validates user credentials and grants access. |
| **Data** | User credentials and access permissions. |

|  |  |
| --- | --- |
| **Identification** | Dashboard |
| **Type** | Class/Form/ |
| **Purpose** | Displays real-time data for monitoring. |
| **Subordinates** | - |
| **Dependencies** | Real-time Updates |
| **Interfaces** | User dashboard views |
| **Resources** | Data sources, Analytics tools |
| **Processing** | Presents data and allows user interaction. |
| **Data** | Real-time dashboard data and visualizations |