##### High Level Design (HLD)

##### Online EDA Automation (Web Application)

Revision Number: 0.1

Date of revision:

## **Document Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date Issued** | **Version** | **Author** | **Comments** |
| 19/11/2021 | 0.1 | Sunny Tamang | Introduction & Architecture defined |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Comments** |
|  |  |  |  |

**Approval Status:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Version** | **Reviewed By** | **Approved By** | **Comments** |
|  |  |  |  |  |

**Table of contents**

[High Level Design (HLD) 1](#_Toc1)

[Online EDA Automation (Web Application) 1](#_Toc2)

[**Document Version Control** 2](#_Toc3)

[1 Introduction 4](#_Toc4)

[1.1 Why this Low-Level Design Document? 4](#_Toc5)

[The goal of LLD or a low-level design document (LLDD) is to give the internal logic design of the actual program code for The Online EDA Automation. 4](#_Toc6)

[1.2 Scope 4](#_Toc7)

[2. Architecture 5](#_Toc8)

[3. Architecture Description 6](#_Toc9)

[3.1 Log in 6](#_Toc10)

[3.3 Data Transformation 6](#_Toc11)

[3.4 Data Insertion into Database 6](#_Toc12)

[3.5 Export Data from Database 6](#_Toc13)

[3.6 Customization by User 6](#_Toc14)

[3.7 Save the customization 7](#_Toc15)

[3.8 Data Insertion into Database 7](#_Toc16)

[3.9 Dashboard Maintenance 7](#_Toc17)

[4.0 Deployment 7](#_Toc18)

[4 Unit Test Case 8](#_Toc19)

##### 1 Introduction

##### 1.1 Why this Low-Level Design Document?

##### The goal of LLD or a low-level design document (LLDD) is to give the internal logic design of the actual program code for The Online EDA Automation.

LLD describes the class diagrams with the methods and relations between classed and program specs. It describes the modules so that the programmer can directly code the program from the document.

##### 1.2 Scope

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture source code and ultimately, performances algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work.

##### 2. Architecture

Data Upload

Log in

start

Export Data from Database

Data Insertion into Database

Data Transformation

Save the customization

Customization by user

Pre built EDA steps

Cloud setup

Dashboard maintenance

Data Insertion into Database

Log in

Application Start

Deployment

Data Insertion into Database

Data Validation

Data Upload

Save the customization

Customization by user

Pre built EDA steps

End

Dashboard maintenance

##### 3. Architecture Description

##### 3.1 Log in

The user needs to sign up to access the application as it will help maintain the user data for the dashboard and once sign up user can log in and access the application.

3.2 Data Upload

Once the user the logged in he/she has to load the dataset which can be of different format like csv, json, xml etc.

##### 3.3 Data Transformation

Data will be converted into rows and columns with headers in order to carry out further activities.

##### 3.4 Data Insertion into Database

* Database creation and connection
  + Create a database with the file name but without any space
  + If the database already created by the user, open the connection to the database
* Table creation in the database
* Insertion of the data in the file in the table

##### 3.5 Export Data from Database

Data Export from Database - The data in a stored database is exported as a CSV file to be used for further EDA steps.

##### 3.6 Customization by User

Once the pre built EDA steps executes, user can customize the steps as per need.

##### 3.7 Save the customization

User can save the customized steps as per need and user it in future as well

##### 3.8 Data Insertion into Database

Once is happy and wish to save it then the template is saved in Database for further activities

##### 3.9 Dashboard Maintenance

Dashboard can be created on the basis of user EDA activities

##### 4.0 Deployment

We will deploy this in AWS, GCP, Azure and Heroku cloud

##### 4 Unit Test Case

|  |  |  |
| --- | --- | --- |
| **Test Case Description** | **Pre-Requisite** | **Expected Result** |
| Verify whether the Application URL is  accessible to the user | 1. Application URL  should be defined | Application URL should be  accessible to the user |
| Verify whether the Application loads  completely for the user when the URL  is accessed | 1. Application URL  is accessible  2. Application is  deployed | The Application should load  completely for the user when the  URL is accessed |
| Verify whether the User is able to sign  up in the application | 1. Application is  accessible | The User should be able to sign up  in the application |
| Verify whether user is able to  successfully login to the application | 1. Application is  accessible  2. User is signed up  to the application | User should be able to successfully  login to the application |
| Verify whether user is able to see input  fields on logging in | 1. Application is  accessible  2. User is signed up  to the application  3. User is logged in  to the application | User should be able to see input  fields on logging in |
| Verify if the user is upload the dataset | 1. Application is  accessible  2. User is signed up  to the application  3. User is logged in  to the application | User should be able to upload the dataset |
| Verify whether user is able to edit the fields/drop down | 1. Application is  accessible  2. User is signed up  to the application  3. User is logged in  to the application | User should be able to edit the fields/drop down |
| Verify whether user gets Submit  button to submit the inputs | 1. Application is  accessible  2. User is signed up  to the application  3. User is logged in  to the application | User should get Submit button to  submit the inputs |
| Verify whether user is presented with  recommended results on clicking  submit | 1. Application is  accessible  2. User is signed up  to the application  3. User is logged in  to the application | User should be presented with  recommended results on clicking  submit |

|  |  |  |
| --- | --- | --- |
| Verify whether the recommended  results are in accordance to the  selections user made | 1. Application is  accessible  2. User is signed up  to the application  3. User is logged in  to the application | The recommended results should  be in accordance to the selections  user made |
| Verify whether user has options to  filter the recommended results as well | 1. Application is  accessible  2. User is signed up to the application  3. User is logged in  to the application | User should have options to filter  the recommended results as well |
| Verify whether KPIs modify as per the  user inputs | 1. Application is  accessible  2. User is signed up  to the application  3. User is logged in  to the application | KPIs should modify as per the user  inputs |
| Verify whether the KPIs shows details of the user on the basis of EDA steps | 1. Application is  accessible  2. User is signed up  to the application  3. User is logged in  to the application | The KPI should show details of the user on the basis of EDA steps |