Low Level Design

Campus Placement Prediction System

|  |  |
| --- | --- |
| Written By | Jatin Aggarwal |
| Document Version | 0.3 |
| Last Revised Date | 05– March-2024 |

**Document Control**

### Change Record:

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comments** |
| 0.1 | 04 – March -  2024 | Jatin Aggarwal | Introduction & Architecture defined |
| 0.2 | 05 – March -  2024 | Jatin Aggarwal | Architecture & Architecture Description appended and updated |
| 0.3 | 05 – March-  2024 | Jatin Aggarwal | Unit Test Cases defined and appended |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### Reviews:

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Reviewer** | **Comments** |
| 0.2 | 05 – March -  2024 | Jatin Aggarwal | Document Content , Version Control and Unit Test Cases to be added |

### Approval Status:

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

Contents

1. [**Introduction**](#_heading=h.gjdgxs) 1
   1. [**What is Low-Level design document?**](#_heading=h.30j0zll) 1
   2. [**Scope**](#_heading=h.1fob9te) 1
2. [**Architecture**](#_heading=h.3znysh7) 2
3. [**Architecture Description**](#_heading=h.2et92p0) 3
   1. [**Data Description**](#_heading=h.tyjcwt) 3
   2. [**Web Scrapping**](#_heading=h.3dy6vkm) 3
   3. [**Data Transformation**](#_heading=h.1t3h5sf) 3
   4. [**Data Insertion into Database**](#_heading=h.4d34og8) 3
   5. [**Export Data from Database**](#_heading=h.2s8eyo1) 3
   6. [**Data Pre-processing**](#_heading=h.17dp8vu) 3
   7. [**Data Clustering**](#_heading=h.3rdcrjn) 3
   8. [**Model Building**](#_heading=h.26in1rg) 4
   9. [**Data from User**](#_heading=h.lnxbz9) 4
   10. [**Data Validation**](#_heading=h.35nkun2) 4
   11. [**User Data Inserting into Database**](#_heading=h.1ksv4uv) 4
   12. [**Data Clustering**](#_heading=h.44sinio) 4
   13. [**Model Call for Specific Cluster**](#_heading=h.2jxsxqh) 4
   14. [**Recipe Recommendation & Saving Output in Database**](#_heading=h.z337ya) 4
   15. [**Deployment**](#_heading=h.3j2qqm3) 4
4. [**Unit Test Cases**](#_heading=h.1y810tw) 5

# Introduction

## What is Low-Level design document?

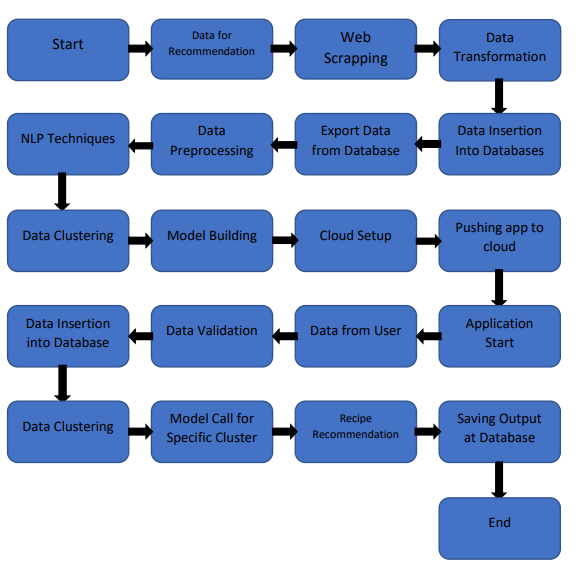
The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Campus Placement Prediction System. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

## Scope

Low-level design (LLD) is a component-level design process that follows a step-by-

step [refinement](https://en.wikipedia.org/wiki/Refinement_(computing)) process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work

# Architecture



# Architecture Description

## Data Description

The data required for the Campus Placement Prediction System includes various attributes related to students' academic performance, skills, internship experiences, and placement outcomes. Here's a description of the key data attributes:

Student Information:

Gender: The gender of the student (male, female, other).

Age: The age of the student.

Specialization: The specialization or major of the student (e.g., Computer Science, Finance).

Undergraduate GPA: The Grade Point Average (GPA) obtained by the student during their undergraduate studies.

Work Experience: The number of years of work experience, if any, the student has.

Placement Status: The placement status of the student (placed, not placed).

This comprehensive dataset provides the necessary information to train machine learning models for predicting students' placement probabilities accurately. By analyzing patterns and trends in historical data, the system can generate personalized recommendations and guidance to optimize the campus placement process for students, educational institutions, and recruiters.

## Web Scrapping

In order to create a more complete Prediction we will need some more datasets which will contain more information.

## Data Transformation

In the Transformation Process, we will convert our original dataset which is in JSON format to CSV Format if it is not in csv format. And will merge it with the Scrapped dataset.

## Data Insertion into Database

1. Database Creation and connection - Create a database with name passed. If the database is already created, open the connection to the database.
2. Table creation in the database.
3. Insertion of files in the table

## Export Data from Database

Data Export from Database - The data in a stored database is exported as a CSV file to be used for Data Pre-processing and Model Training.

## Data Pre-processing

Data Pre-processing steps we could use are Null value handling, label encoding , categorical to numerical features , Imbalanced data set handling, Handling columns with standard deviation zero or below a threshold, etc.

## Data Clustering

It is according to the problem statement

Data clustering is not use in this specific problem

## Model Building

After Preprocessing id done, we will find the best model for prediction. For each model, algorithms will be passed with the best parameters derived from Grid-Search. We will calculate the AUC scores for models and select the model with the best score. Similarly, the models will be selected for each model. All the models for every algorithm will be saved for use in Prediction.

## Data from User

Here we will collect physiological data from user such as user Internship, CGPA, Backlogs, etc.

## Data Validation

Here Data Validation will be done, given by the user

## User Data Inserting into Database

Collecting the data from the user and storing it into the database. The database can be either MySQL or Mongo DB.

## Data Clustering

Not used.

## Model Call for Specific Algorithm

Based on the model number, the respective model will be loaded and will be used to predict/Recommend the data for that Dataset.

## Prediction & Saving Output in Database

After calling model Predict/Output will be Predicted, this output will be saved in Database and it will be used to show the same Output if other users provide the same data.

## Deployment

We will be deploying the model to AWS.

This is a workflow diagram for the Campus Placement Project.

# Unit Test Cases

|  |  |  |
| --- | --- | --- |
| **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 whether user is able to edit all input fields | 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 all input fields |
| 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 | User should have options to filter the recommended results as well |

|  |  |  |
| --- | --- | --- |
|  | to the application  3. User is logged in to the application |  |
| Verify whether KPIs modify as per the user inputs for the user's Placement | 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 for the user's Placement |
| Verify whether the KPIs indicate details of the Prediction | 1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application | The KPIs should indicate details of the Prediction |