# Low Level Design (LLD)

# Parkinson’s

# Disease prediction

**Version 1.0**

**Date of Revision 08 May 2022**

# Contents

Document Version Control Abstract

1. Introduction
   1. What is a Low-Level design document?
   2. Scope
2. Architecture
3. Architecture Description
   1. Data Description
   2. Data Insertion into Database
   3. Data Preprocessing
   4. Model Building
   5. Recommendation
   6. Deployment
4. Unit Test Cases

**Document Version Control**

**Date Issued**

**Versio n**

**Description Author**

0 8 / 0 5 / 2022

1 Initial LLD – V1.0 Sachchit S kolekar

## Introduction

### What is a 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 Food Recommendation System. LLD describes the class

diagrams with the

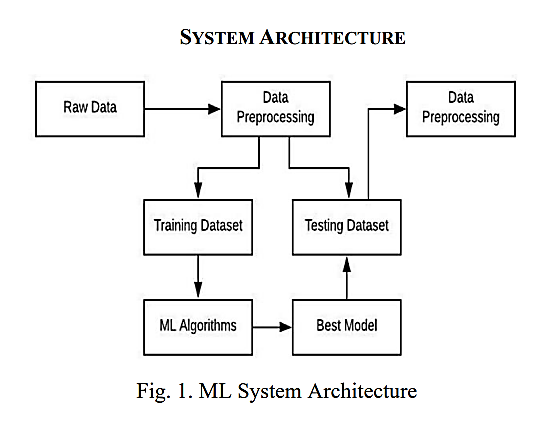
methods and relations between classes and program specs. It describes the modules so that the

programmers 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 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



1. **Architecture Description**

### Model Building

Model will be built using a text vectorization approach and it will create 2 models.

Tags : Will use for Taking inputs.

Prediction R e s : Will use for Obtaining Prediction.

### Prediction

Users can provide input data and the application will Predict whether the persons have disease or not.

### Deployment

We will be deploying the model to Streamlit.

## Unit Test Cases

|  |  |  |
| --- | --- | --- |
| **Test Case Description** | **Prerequisite** | **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. Applicati on URL is accessible 2. Applic ation is   deployed | The Application should load completely for the user when the URL is accessed |
| Verify whether user is able to successfully login to the application | 1. Applic ation 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. Applic ation is accessible | User should be able to see input fields on logging in |
| Verify whether user is able to edit all input fields | 1. Applic ation is accessible | User should be able to edit all input fields |
| Verify whether user gets Submit button to submit the inputs | 1. Applic ation is accessible | User should get Submit button to submit the inputs |
| Verify whether user is presented with recommended results on clicking  submit | 1. Applic ation is accessible | User should be presented with recommended results on clicking  submit |

|  |  |  |
| --- | --- | --- |
|  | 1. Applic |  |
|  | ation is |  |
|  | accessible |  |
| Verify whether the | 2. User is | The recommended results |
| recommended results are in | signed up to the | should be in accordance to the |
| accordance to the selections | application | selections user made |
| user made | 3. User is |  |
|  | logged in to the |  |
|  | application |  |