

Low Level Design (LLD)

Book Recommender System

Version 1.0

Date of Revision 15 April 2022

Contents

Document Version Control

Abstract

1. Introduction

1.1 What is a Low-Level design document?

1.2 Scope

2. Architecture

3. Architecture Description

3.1 Data Description

3.2 Data Insertion into Database

3.3 Data Preprocessing

3.4 Model Building

3.5 Recommendation

3.6 Deployment

4. Unit Test Cases

Document Version Control

Date Issued	Version	Description	Author
27th February 2022	1	Initial LLD – V1.0	Savan Javia

1. Introduction

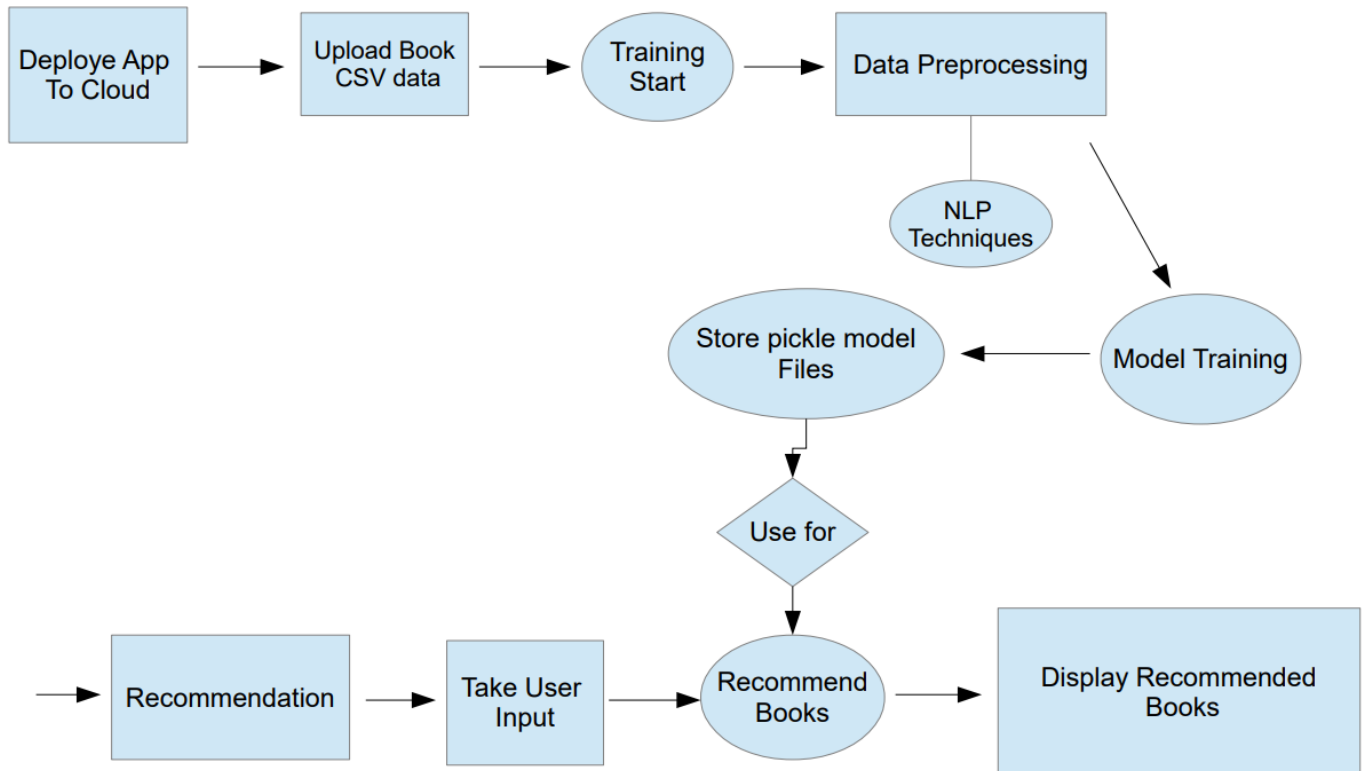
1.1 What is a Low-Level design document?

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.

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, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work.

2. Architecture



3. Architecture Description

3.1 Data Description

Book CSV dataset contains information of Book Title, Genre, Subgenre, Author, Publisher will be used in model training.

3.2 Data Insertion into Database

We can upload book csv data and it will be stored in the database, and later on it will be used for model training and recommendation.

3.3 Data Preprocessing

Use NLP Techniques like removing stop words, apply stemming on dataset.

3.4 Model Building

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

Tags : Will use for search for books at the time of recommendation

Similarity : Will use for finding nearest vector distance books

3.5 Recommendation

Users can select interested books and the application will recommend similar books.

3.6 Deployment

We will be deploying the model to Heroku.

4. 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. Application URL is accessible 2. Application 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. 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	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	User should be able to edit all input fields
Verify whether user gets Submit button to submit the inputs	1. Application is accessible	User should get Submit button to submit the inputs
Verify whether user is presented with recommended results on clicking submit	1. Application is accessible	User should be presented with recommended results on clicking submit

Verify whether the recommended results are in accordance to the selections user made	<ol style="list-style-type: none">1. Application is accessible2. User is signed up to the application3. User is logged in to the application	The recommended results should be in accordance to the selections user made
--	--	---