Low Level Design

*Spam Detection System*

|  |  |
| --- | --- |
| Written by | Sarthak Bathla, Srishti Gupta |
| Last revised date | 28-2-24 |

Document Control

|  |  |  |
| --- | --- | --- |
| Date | Author | Comments |
| 25-2-24 |  | Introduction and architecture defined |
| 26-2-24 |  | Architecture and architecture description appended and updated |
| 27-2-24 |  | Unit test cases defined and appended |

Reviews

|  |  |  |
| --- | --- | --- |
| Date | Reviewer | Comment |
| 27-2-24 |  | Unit test cases have to be added |

Approval status

|  |  |  |  |
| --- | --- | --- | --- |
| Reviewed date | Reviewed by | Approved by | Comments |
|  |  |  |  |

**Contents**

1. Introduction.......................................................................................................................................1

1.1. What is Low-Level design document? ......................................................................................1

1.2. Scope.........................................................................................................................................1

2. Architecture........................................................................................................................................2

3. Architecture Description.....................................................................................................................3

3.1. Data Description........................................................................................................................3

3.2. Working with CQL......................................................................................................................3

3.3. Data Preprocessing....................................................................................................................3

3.4. Model Building..........................................................................................................................3

3.5. Data from user...........................................................................................................................3

3.6. Data Validation .........................................................................................................................3

3.7. Model call .................................................................................................................................3

3.8 . Deployment .............................................................................................................................3

4. Unit Test Cases ...................................................................................................................................4

**1. Introduction**

1.1. 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 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

programmer can directly code the program from the document.

1.2. Scope

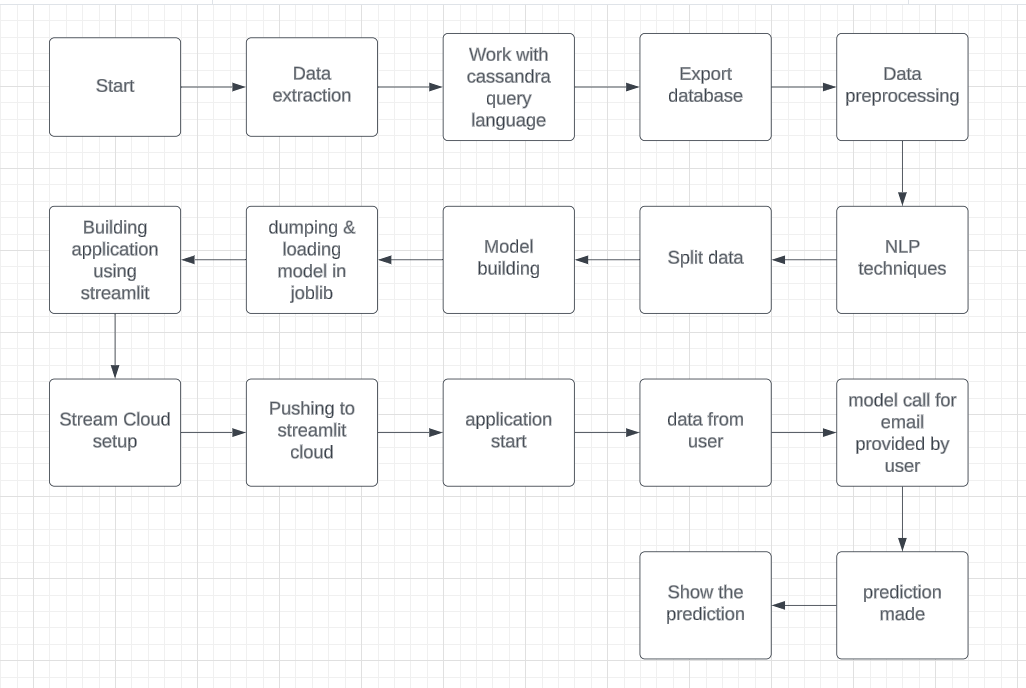
Low-level design (LLD) is a component-level design process that follows a step-by-

step refinement process. This process can be used for designing data structures, required software

architecture, source code and ultimately, 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

Spam detection dataset is the biggest publicly available spam mails dataset. Each mail has been categorized into spam and not spam. There are total 5573 rows which contains all possible type of spam mails.

3.2 Working with Cassandra Query Language

Cassandra Query Language," which is a query language for the Apache Cassandra database. Apache Cassandra is a distributed NoSQL database management system designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure. Here we uploaded and manipulated the data using CQL.

3.3 Data preprocessing

Data preprocessing techniques have been used. Data is being vectorized i.e converted into an array where data is being represented in binary format.

3.4 Model Building

We will choose the best model for training and prediction. Best parameters will be passed . Model with the highest accuracy will be chosen.

3.5 Data from user

We will collect mail body in text format which is supposed to be categorized

3.5 Data validation

Here, data validation will be done.

3.6 Model call

Model will be called and the data provided by user will be loaded. Categorization done by model is represented.

3.7 Deployment

The deployment is done on streamlit cloud.

**4. Unit Test Cases**

|  |  |  |
| --- | --- | --- |
| **Test Case Description** | Pre-requisite | Expected Results |
| Empty Email | None | Classify as not spam |
| Email with Common Spam Keywords | Email content contains common spam keywords | Classify as spam |
| Email with Uncommon Spam Keywords | Email content contains uncommon spam keywords | Classify as spam |
| Email with HTML Tags | Email contains HTML tags | Classify as not spam |
| Email with Attachments | Email contains attachments | Classify as not spam |
| Email with Known Sender | Sender's email address is whitelisted | Classify as not spam |
| Email with Unknown Sender | Sender's email address is not recognized | Classify as spam |
| Email with Misspelled Words | Email contains misspelled words | Classify as spam |
| Email with Excessive Punctuation | Email contains excessive punctuation marks | Classify as spam |