

# Architecture

Random Jokes App

Written By	Vaishnavi Jaju
Document Version	1.0
Last Revised Date	26/03/2023

## Document Control

### Change Record:

Version	18/03/2023	Author	Comments
1.0	18/03/2023	Vaishnavi Jaju	Introduction & Architecture defined
1.1	23/03/2023	Vaishnavi Jaju	Architecture & Architecture Description appended
1.2	25/03/2023	Vaishnavi Jaju	Unit Test Cases defined and appended

### Reviews:

Version	Date	Reviewer	Comments

### Approval Status:

Version	Review	Reviewed By	Approved By	Comments

## Contents

<b>1. Introduction .....</b>	<b>1</b>
<b>1.1. What is Architecture Design document?.....</b>	<b>1...</b>
<b>1.2. Scope.....</b>	<b>1</b>
<b>2. Architecture .....</b>	<b>2</b>
<b>3. Architecture Description .....</b>	<b>3</b>
<b>3.1. Data Description.....</b>	<b>3</b>
<b>3.2. Web Scrapping .....</b>	<b>3</b>
<b>3.3. Data Transformation .....</b>	<b>3</b>
<b>3.4. Data Insertion into Database.....</b>	<b>3</b>
<b>3.5. Export Data from Database .....</b>	<b>3</b>
<b>3.6. Data Pre-processing.....</b>	<b>3</b>
<b>3.7. Data Clustering.....</b>	<b>3</b>
<b>3.10. Model Building.....</b>	<b>4</b>
<b>3.11. Data from User.....</b>	<b>4</b>
<b>3.12. Data Validation .....</b>	<b>4</b>
<b>3.13. User Data Inserting into Database.....</b>	<b>4</b>
<b>3.14. Data Clustering.....</b>	<b>4</b>
<b>3.15. Model Call for Specific Cluster.....</b>	<b>4</b>
<b>3.16. Jokes Recommendation &amp; Saving Output in Database.....</b>	<b>4</b>
<b>3.17. Deployment .....</b>	<b>4</b>
<b>4. Unit Test Cases .....</b>	<b>5</b>

## 1. Introduction

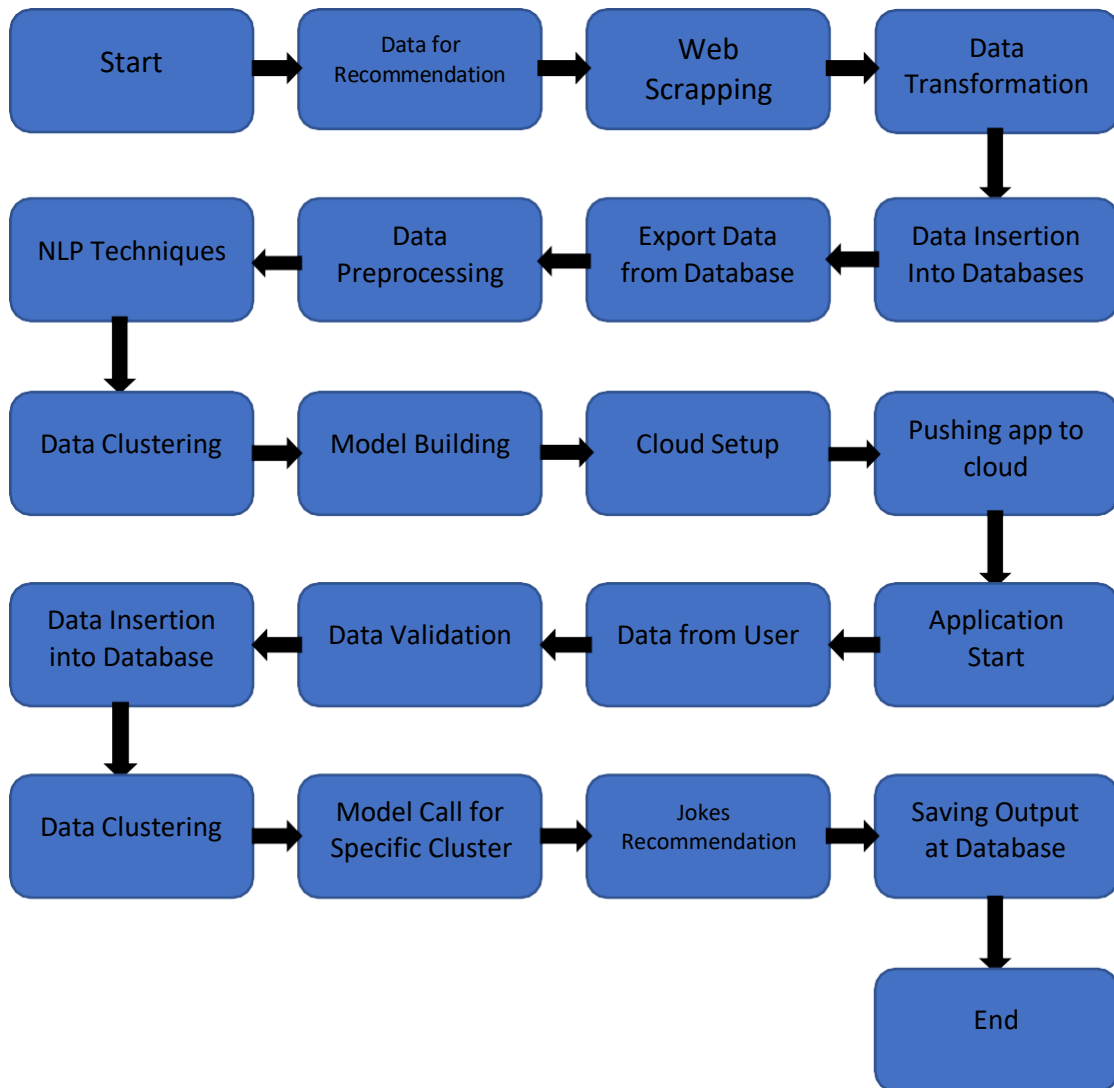
### 1.1. What is Architecture design document?

Any software needs the architectural design to represent the design of software. IEEE defines architectural design as “the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system.” The software that is built for computer-based systems can exhibit one of these many architectures.

### 1.2. Scope

Architecture Design Document (ADD) is an architecture design process that follows a step-by-step refinement process. The process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the design principles may be defined during requirement analysis and then refined during architectural design work.

## 2. Architecture



### 3. Architecture Description

#### 3.1. Data Description

We can get different types of jokes at a one place by just clicking on button.

#### 3.2. Web Scrapping

In order to get a more new jokes we will need some more datasets which will contain different defferent types of jokes.

#### 3.3. Data Transformation

In the Transformation Process, we will convert our original dataset which is in JSON format to CSV Format. And will merge it with the Scrapped dataset.

#### 3.4. Data Insertion into Database

- a. Database Creation and connection - Create a database with name passed. If the database is already created, open the connection to the database.
- b. Table creation in the database.
- c. Insertion of files 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 Data Pre-processing and Model Training.

#### 3.6. Data Pre-processing

Data Pre-processing steps we could use are Null value handling, stop words removal, punctuation removal, Tokenization, Lemmatization, TFIDF, Imbalanced data set handling, Handling columns with standard deviation zero or below a threshold, etc.

#### 3.7. Data Clustering

K-Means algorithm will be used to create clusters in the pre-processed data. The optimum number of clusters is selected by plotting the elbow plot. The idea behind clustering is to implement different algorithms to train data in different clusters. The K-means model is trained over preprocessed data and the model is saved for further use in prediction

### 3.10. Model Building

After clusters are created, we will find the best model for each cluster. For each cluster, 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 cluster. All the models for every cluster will be saved for use in Recommendation.

### 3.11. Data from User

Here we don't need any kind of data from user, user just need to click on generate joke button and they will get new joke everytime.

### 3.12. Data Validation

Here No Data Validation will be done, cause no need of data from user.

### 3.13. 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. (Here we didn't want to )

### 3.14. Data Clustering

The model created during training will be loaded, and clusters for the user data will be predicted.

### 3.15. Model Call for Specific Cluster

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

### 3.16. Jokes Recommendation & Saving Output in Database

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

### 3.17. Deployment

We will be deploying the model to AWS.  
This is a workflow diagram for the Random jokes app.

## 4. 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 health	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 health
Verify whether the KPIs indicate details of the suggested recipe	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 suggested recipe