1. About this document

a) Purpose & Scope of the document

The purpose of the functional requirements document is to systematically capture requirements for the project and the case study to be developed. Both functional and non-functional requirements of this project are captured in this document. It also serves as the input for the project scope. The scope of this document is limited to addressing the requirements from a user, quality system requirement and non-functional perspective.

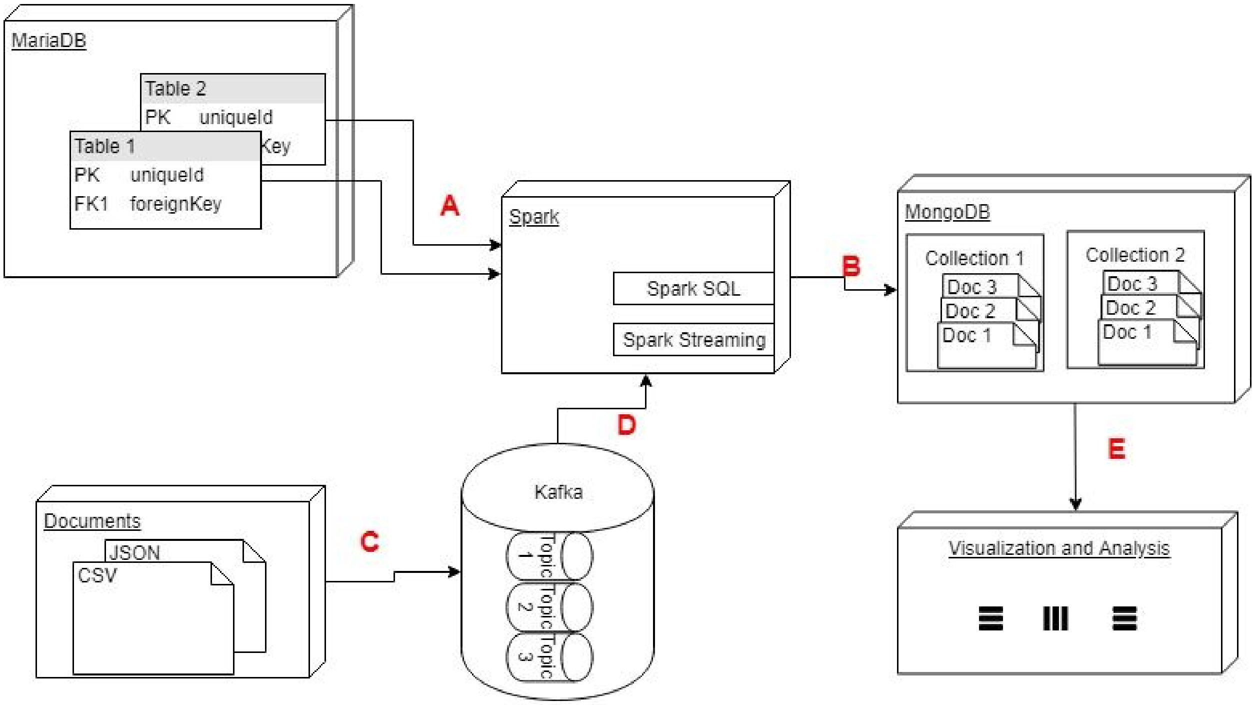
b) Impact of the System

This is a new product, which has been developed for internal use. The expected impact of the project is to automate existing manual processes in order to make them more efficient and cost-effective.

c) Dataset and Database

Use “Credit Card Database” from RDBMS and “Health Insurance Marketplace Data,” which you have to read/extract by streaming tools like Spark SQL.

2. System Requirements Flow Diagram



Overview of System Requirements

This case study requires students to work with the following technologies in order to manage an ETL process for a Health Insurance Marketplace and Credit Card dataset: Python (PySpark, Pandas, Matplotlib), MariaDB, Apache Spark (Spark Core, Spark SQL, Spark Structured Streaming), MongoDB, Apache Kafka. Students will be expected to set up their environments and perform installations on their local machines.

The case study also will explore the following concepts: data modeling, data warehousing, RDBMS, NoSQL, ETL, SDLC/Agile Methodology.

Part One (A + B) : MariaDB → Spark SQL → MongoDB

1) Students will use Python to extract data from tables stored in MariaDB.
2) Students will transform the data extracted from MariaDB using Spark (SparkSQL).
3) Students will load the transformed data using Spark into MongoDB.

Part Two (C + D + B) : Documents → Kafka → Spark Streaming → MongoDB

1) Students will extract JSON, CSV and other types of data from web URLs using the Spark Streaming integration for Kafka, a stream-processing software.
2) Specifically, students will use the Kafka connector in the Spark Structured Streaming in order to manage the transformation of the data before loading into MongoDB collections.

Part Three (B+E) : MongoDB → Visualization and Analytics

Students will extract/query from MongoDB to visualize the data for analytical functions

3. Functional Requirements

Application Front-end	
Functional Requirement 0.1	Create a Python program to display the following options. a) Part One: Read credit card data from MariaDB b) Part Two: Read data from “Health Insurance Marketplace”.
Phase one	
	Data Extraction and Transformation with Python and Spark SQL
Functional Requirement 1.1	Utilize Python and SparkSQL to read/extract the following data from “Credit Card Database (MariaDB)” and **according to the specifications found in the mapping document.** 1. Table One: CDW_SAPP_BRANCH 2. Table Two: CDW_SAPP_CREDITCARD 3. Table Three: CDW_SAPP_CUSTOMER **Note**: **Data Engineers will be required to transform the data based on requirements found in the Mapping Document.**

	Data loading with MongoDB and Python
Function Requirement 1.2	Once Spark SQL reads data from RDBMS (MariaDB) then utilize Python and Python modules to load data into MongoDB. a) Create a Python Program (pymongo) to load/store “Credit Card Data” into MongoDB (Hint: make new collections by any “title name” in MongoDB.)
Part Two	
	Data Extraction and Transportation with Python and Kafka
Functional Requirement 2.1	Create a Python program to read/extract the following dataset’s files from “Health Insurance Marketplace Data.” Then produce/write into Kafka multiple topics. 1. BenefitsCostSharing.txt 2. ServiceArea.csv 3. Insurance.csv 4. PlanAttributes.csv 5. Network.csv Hint: (Utilize python Kafka modules and other python modules. Create topics by the name of above CSV files e.g: BenefitsCostSharing, ServiceArea, Insurance, PlanAttributes, Network.)
	Data transformation with Python, Pyspark and Spark streaming
Functional Requirement 2.2	Create a Spark streaming program to read/consume “Health Insurance Marketplace Data” from Kafka topics and perform a transformation, if needed. (Hint: Spark will act as a consumer. Spark must make a connection with Kafka and make individual Python file for Kafka topics)

	Data loading with MongoDB and Python
Functional Requirement 2.3	Once Spark has consumed the data from Kafka topics, then load data into MongoDB. a) Create a Python program (pymongo) to load/store “Health Insurance Marketplace Data” into MongoDB (Hint: make a new database in MongoDB and collections by the name of topics.)
	Analysis and Visualization
Functional Requirement 2.4	Work towards the following requirements: a) Use “Service Area Dataset” from MongoDB. Find and plot the count of **ServiceAreaName, SourceName , and BusinessYear** across the country each state? b) Use “Service Area Dataset” from MongoDB. Find and plot the count of “**sources**” across the country. c) Use the “Benefit-Cost Sharing” dataset from MongoDB. Display a table of the names of the plans with the most customers by state, the number of customers for that plan and the total number of customers. (Hint: use Statecode, benefitName) d) Use the “Benefit Cost Sharing” dataset from MongoDB. Find and plot the number of benefit plans in each state. e) Use the “Insurance” dataset from MongoDB and find the number of mothers who smoke and also have children. f) Use the “Insurance” dataset from MongoDB. Find out which region has the highest rate of smokers. Plot the results for each region.