**Requirement Specification Document**

**Introduction**

1. **Purpose** -This document will outline the project requirements for building a data pipeline that processes, cleans, and analyzes insurance data coming from different sources. This will then feed into the company's strategy in terms of customer segmentation, policy personalization, and the growing of revenues.
2. **Intended Audience and Use** - The intended audience includes data engineers, data scientists, project managers, and QA testers. This document will guide the development, testing, and deployment phases, ensuring the team has a clear understanding of requirements and expected outcomes.
3. **Product Scope** - The project will support the Health Care insurance company’s goal to increase revenue by identifying key customer insights from competitor data. It will enable targeted offers, customer royalties, and accurate reporting on claims and policyholder demographics, aligning with the broader business objectives of customer retention and revenue enhancement.
4. **Definitions and Acronyms** –
   1. SRS: Software Requirements Specification
   2. AWS S3: Amazon Web Services Simple Storage Service
   3. EMR: Elastic MapReduce
   4. Pyspark: Python Apache Spark
   5. ETL: Extract, Transform, Load

**Description**

This project will create a Big Data pipeline to analyze healthcare competitor data to provide insights for an insurance company. The analysis will reveal trends in claims, demographics, and policy preferences, informing the company’s business strategies.

* 1. User Needs - The system will be used by:
     + 1. Data Analysts: To track and analyze competitor data for insights into customer behavior.
       2. Product Managers: To strategize customized offers and loyalty programs based on customer demographics.
       3. Executives: To review summarized insights to inform revenue enhancement strategies.
  2. Assumptions and Dependencies
     1. The project assumes access to competitor data is obtained through scraping and third-party vendors.
     2. Data processing will be performed in an AWS-based environment with Redshift and Databricks.
     3. All datasets (Patients, Subscriber, Claims, Group\_subgroup) are expected to be stored and accessed through AWS S3.

**3. System Features and Requirements**

**a. Functional Requirements**

1. **Data Ingestion**:
   * Upload data from various sources into AWS S3 (folder: input-data).
2. **Data Cleaning**:
   * Detect and handle null values, remove duplicates, and format data to maintain consistency.
   * Clean datasets for Patients, Subscriber, Claims, and Group\_subgroup.
   * Store cleaned datasets in Redshift tables with schema design documentation.
3. **Data Analysis Requirements**:
   * Identify trends and patterns, such as:
     + Maximum number of claims by disease.
     + Subscribers under 30 in subgroups.
     + Hospitals serving the highest patient volumes.
     + Most profitable groups and frequently subscribed subgroups.
     + City-based claim volume and average monthly premium per subscriber.
4. **Result Generation**:
   * For each use case, create separate Redshift tables in the Project-Output schema.
5. **Deployment**:
   * Deploy code on AWS EMR or Databricks, with versioning and collaboration through GitHub.

**b. External Interface Requirements**

1. **User Interface**:
   * Databricks and AWS Redshift consoles for data interaction.
   * JIRA for sprint tracking and project management.
2. **Software Interfaces**:
   * Integration with AWS S3, Redshift, and GitHub.
   * Pyspark scripts for data processing.

**c. System Features**

* Data visualization and report generation using Databricks.
* Automated ETL process on AWS EMR for real-time updates and scalability.

**d. Nonfunctional Requirements**

1. **Performance Requirements**:
   * Data processing must be completed within a maximum of 4 hours per batch.
2. **Safety Requirements**:
   * Ensure data privacy and comply with relevant healthcare data protection standards.
3. **Security Requirements**:
   * Use AWS security protocols to control access and encrypt data in transit and at rest.
4. **Usability Requirements**:
   * Provide clear documentation and data validation checks to ease data analysis.
5. **Scalability Requirements**:
   * System must be scalable to handle increased data volume from new sources.