### **Requirements Specifications Document**

#### **1. Introduction**

### This document details the requirements for a Big Data solution aimed at enhancing revenue and improving customer insights for a healthcare insurance company. By leveraging data about competitors from varieties of sources, namely through scrapping and third-party sources, this system will create data pipelines for the Health Care insurance company which will make the company make appropriate business strategies to enhance their revenue by analyzing customers behaviors and send offers and royalties to customers respectively.

1. **Purpose:**

### The purpose of this requirements specifications document is to define the scope and requirements for developing a data analytics pipeline that will process and analyze healthcare insurance data. This pipeline will enable the company to make informed decisions to boost revenue and better serve customers by understanding their needs and behaviors so that they can send offers and royalties to customers respectively.

1. **Intended Audience and Use:**

### **Developers**: To understand the system's technical and functional requirements, including data ingestion, processing, and transformation.

### **Testers**: To create test cases and validate that the system meets the outlined requirements.

### **Project Managers**: To track the progress, manage the project timeline, and ensure that all requirements are met.

### **Data Analysts**: To use the output of the system for further analysis and decision-making.

* **Data Scientists**: To use the output of the system for further analysis to make future predictions and create machine learning models.

1. **Product Scope:**

This platform aims to provide insights from healthcare insurance data by identifying trends in claims, customer demographics, policy preferences, and profitability. It aligns with the company’s business goals to increase revenue and customer retention through customer targeted offerings. The platform will support decision-making processes by delivering actionable insights about customer behavior and operational performance.

#### **d. Definitions and Acronyms**

### **AWS S3**: Amazon Web Services Simple Storage Service, used for storing data files.

### **AWS Redshift**: AWS Redshift, a cloud data warehouse for storing processed data.

### **AWS EMR Studio**: AWS Elastic MapReduce, used for big data processing and transformations. It can also be used for ETL jobs.

### **Databricks**: A cloud-based platform for big data which will be used here for data processing and visualization with the help of notebooks.

### **Jira**: A tool for project management and sprint tracking.

### **PySpark**: Python API for Spark, used for data processing and analytics.

* **GitHub:** Github integrates with many tools like Databricks, AWS, etc. which helps to create seamless workflows from code development to deployment.

**2. Overall Description**

This project involves creating a Big Data analytics platform for a healthcare insurance company. It will be a new data-driven solution to analyze customer and claims data, sourced from competitors and third parties. The insights generated will help the company target customers with tailored offers, track claims, and identify profitable policies, enhancing revenue. This platform will be deployed on AWS, Pyspark and Databricks with data uploaded on S3 first, followed by ingestion into Databricks from S3 where it will be transformed, then stored in Redshift and lastly visualized using Databricks.

1. **User Needs:** This project will be used by multiple team members and stake-holders for various reasons:

* **Insurance Data Analysts**: Need insights into customer behavior, claims patterns, and profitable policies.

### **Marketing Team**: Requires data on customer demographics and behaviors to design personalized offers.

### **Financial Team**: Needs to track and calculate royalties and claims to manage company revenue.

### **Data Engineers**: Need tools and infrastructure to manage and process large datasets securely and efficiently.

1. **Assumptions and Dependencies:** We will be making some informed assumptions and dependencies to save us time in the future, some of which are outlines as follows:

* **Assumptions**:

### Data sources will include competitive data scraped from public sources and third-party datasets.

### Existing infrastructure includes AWS services such as S3, EMR, Redshift, and also Databricks.

### **Dependencies**:

### AWS for storage, data processing, and analytics.

### Databricks for data transformation and visualization.

### GitHub for version control.

### Jira for project and sprint management.

* + PySpark for ETL jobs.

**3. System Features and Requirements:**

### **Functional Requirements**:

### **Data Ingestion**:

### Collect and upload sample data to an S3 bucket in the folder input-data.

### **Data Cleaning**: From S3 import the data into Databricks for data cleaning. Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled. If data is incorrect, outcomes and algorithms are unreliable, even though they may look correct. There is no one absolute way to prescribe the exact steps in the data cleaning process because the processes will vary from dataset to dataset. For this project, we will be following below mentioned general procedure:

* + 1. First check if there are null values in the dataset.
    2. Count the total Null values for each column.
    3. And then replace the null values for specific columns by NA.
    4. Check if there are duplicate records.
    5. If there are duplicates then drop duplicates.

### Clean data for datasets like Patients, Subscriber, Claims and Group\_Subgroup.

### **Data Transformation**:

### Transform the cleaned data in Databricks and load it into Redshift tables for further analysis.

### **Data Analysis and Reporting**:

### Analyze data based on specific business requirements, such as claims analysis, subscriber demographics, policy popularity, etc.

### Generate Redshift tables under Project-Output schema for each specific output, based on use cases.

### **Visualization**:

### Use Databricks to create visualizations for presenting results.

### **External Interface Requirements**: We may also have requirements that outline how our software will interact with other tools There are several types of interfaces we may have requirements for, including:

### **User Interface**:

### Databricks for data visualization and insights.

### **Software Interface**:

### Integrate with AWS Redshift for data storage and querying.

* + 1. Use Databricks notebook for data cleaning and loading to Redshift.

### AWS EMR and PySpark for data processing and analytics.

### **Communication Interface**:

### Connect to S3 for data storage and retrieval.

1. **System Features:** System features are a type of functional requirements. These are features that are required in order for a system to function.

* **Data Collection from Multiple Sources**: Enable the system to collect data from various sources, including web scraping and third-party APIs, to gather competitor and customer information.
* **Automated Data Loading to AWS S3**: Automatically ingest raw data into an AWS S3 bucket in a structured format for further processing.
* **Data Processing with PySpark on Databricks**: Process and analyze large datasets using PySpark on Databricks, allowing for scalable big data computations.
* **Databricks Visualizations**: Generate visual representations of the results (e.g., charts and graphs) within Databricks to assist in data analysis and insights sharing.
* **Data Upload to AWS Redshift**: Load cleaned and transformed data into AWS Redshift tables for structured storage and efficient querying.

### **Nonfunctional Requirements**:

### **Performance Requirements**:

### The system should handle large datasets efficiently using Spark on Databricks.

### **Safety Requirements**:

### Data should be stored securely in AWS Redshift and S3 with restricted access.

### **Security Requirements**:

### Implement AWS Identity and Access Management (IAM) roles for secure access to AWS resources.

### **Usability Requirements**:

### The platform should be accessible for end-users through Databricks for easy visualization.

### **Scalability Requirements**:

### The system should scale to accommodate an increase in data volume and additional analytical requirements.

### 

## 