Requirements Specifications Document

# Introduction

The Health Care Insurance Industry needs to implement a Big Data Ecosystem that can analyze competitor company data received from different sources through scrapping and third-party services. We plan to track down the behavior and condition of customers so that we can offer customers insurance policies, and those customers who bought policies in the past calculate royalties, which enhances the revenue.

a. Purpose -This document aims to build a big data analytics solution for the healthcare industry.

b. Scope: The project includes creating data pipelines to process large volumes of data from different sources, such as web scraping and third-party data. The process data should generate insight into the business strategies. We will use AWS S3, Databricks, PySpark, Redshift, and Jira tools.

c. Definition and Acronyms

AWS S3- Amazon Web Services Simple Storage Service

Databricks- Cloud platform that is used for big data analytics

Pyspark- Use for Python API for Apache Spark

Jira- Project management tool

1. Overall Description

The sole purpose of this project is to integrate the Big data ecosystem in healthcare insurance companies by using cloud services like AWS S3, AWS Redshift, and Databricks. We plan to build a data pipeline and clean and process data before loading it into Redshift.

* + - * 1. User Needs

The product will be used by data engineers, data analysts, business analysts, and project managers(stakeholders). Data Engineer uses this product to build, create, and maintain pipelines. Data Analysts utilize this product by analyzing processed data to generate business insights. Business Analysts use this document to gain insights and make business and strategic decisions. Project Managers oversee the progress and ensure deadlines are met.

b. Assumptions and Dependencies –

The operating environment used for the products is:

* AWS Cloud Services (S3, Redshift)
* Databricks community edition
* PySpark
* Jira
* GitHub

# System Features and Requirements -

## a. Functional Requirements – The requirement add value to the product and make sure it meets the need of user expectations.

1. Data Ingestion and Storage-

* The system shall support data uploads to AWS S3.
* The system shall categorize data, for example, patients, subscribers, and claims.
* The system shall store clean data in redshift with proper schemas.
* The system shall ensure data integrity during the loading process.

2. Data Cleaning-

* The system shall find count and null values in the dataset.
* The system shall replace null values with NA when applicable.
* The system shall identify and remove duplicate records
* The system shall clean dataset for patients, subscriber, claims and subgroups.

1. Data Analysis-

* The system shall identify the disease with the maximum number of claims.
* The system shall find subscribers under 30 who subscribe to any subgroup.
* The system shall identify the group with the maximum number of subgroups.
* The system shall determine the hospital that serves the most patients.
* The system shall identify the most frequently subscribed subgroups.
* The system shall calculate the total number of rejected claims.
* The system shall identify the city from which most claims are submitted.
* The system shall determine whether government or private policies are more subscribed.
* The system shall calculate the average monthly premium paid by subscribers.
* The system shall identify the most profitable group.
* The system shall list patients under 18 years of age admitted for cancer.

1. Data Visualization

* The system shall generate visualizations for each use case.
* The system shall allow exporting visualizations for reporting purposes.

## 4. External Interface Requirements -

### User- Databricks notebook interface for data processing and visualization

### Hardware – The system shall operate on the cloud infrastructure, requiring no specific hardware interfaces.

### Operating System – Windows, Mac, Linux

### Browser- Chrome, Firefox, Edge, Safari

### Software- AWS S3, AWS RedShift, Databricks, Pyspark

### Communications- Microsoft Teams.

### 5. System Features -

##### **Data Pipeline**

1. **Data Ingestion:** Data uploaded to AWS S3.
2. **Data Cleaning**: Data cleaned using PySpark on Databricks.
3. **Data Storage:** Cleaned data stored in AWS Redshift.
4. **Data Analysis:** Analysis performed using SQL queries in Redshift.
5. **Data Visualization:** Results visualized in Databricks.

**Database Schema Design**

* **Patients Table:** Primary key: patient\_id
* **Subscribers Table**: Primary key: subscriber\_id
* **Claims Table:** Primary key: claim\_id, Foreign keys: patient\_id, subscriber\_id
* **Group Subgroup Table**: Primary key: group\_id, Foreign keys: subgroup\_id
* Detailed schema design to be provided in a separate document.

## 6. Nonfunctional Requirements -

### Performance requirements

### The system should process data within 3 seconds

### The system should load across multiple servers, prevent bottlenecks, and perform consistently.

### The system should be available all the time

### Safety requirements

* The system should use the VPN (Virtual Private Network) to protect data from malicious attackers.

### Security requirements

* The system should have data privacy about health care regulations.
* The health care data security must comply with data governance.
* The system should have access to only authorized users.

### Usability requirements

* The system should be easy and user-friendly to use.
* If any crash is encountered, need to immediately report to the development team

### Scalability requirements

* The system should scale resources on a single machine to handle loads.
* The system should be able to handle network traffic as the number of users and data volume grows.
* The database should be able to handle complex queries as data volume grows and maintain query performance.
* The implementation of auto scaling mechanism should be adjusted based on demand, cost-efficiency, and performance.

#### **7. Project Management**

##### 7.1 **Sprint Planning**

* **Week 1**: Documentation and solution design
* **Week 2**: Implementation and testing

##### 7.2 **Jira User Stories and Tasks**

* User stories and tasks will be created for each use case and test case in Jira.
* Tasks will include data ingestion, cleaning, storage, analysis, and visualization.

##### 7.3 **GitHub Repository**

* Code will be version-controlled using GitHub.