**Requirements Specifications Document**

**Introduction**

**a. Purpose**: The purpose of this document is to outline the requirements for the implementation of a Big Data ecosystem to analyze competitors' company data for a Health Care insurance company. The goal is to enhance revenue generation and customer understanding through data analysis.

**b. Intended Audience and Use**: This document is intended for developers, testers, project managers, and other stakeholders involved in the implementation and testing of the proposed solution.

**c. Product Scope**: The project aims to create data pipelines and analysis mechanisms using AWS services, Databricks, and Pyspark to analyze customer behavior, customize offers, calculate royalties, and enhance revenue.

**d. Definitions and Acronyms:**

* AWS: Amazon Web Services
* S3: Simple Storage Service
* EMR: Elastic MapReduce
* NA: Not Applicable

**Overall Description**

1. **User Needs**: The Health Care insurance company requires a solution to analyze competitor data, track customer behavior, and create customized offers and royalties to enhance revenue. Data analysts can use the output table to analyze the results of the pipelines created. Data scientists can use this data to predict the outcome from the output table. Fellow data engineers working in the future in this pipelines can read and know how the data is being collected, stored, cleaned as well as help them analyze the output from the pipelines that are created.

**b. Assumptions and Dependencies:**

* The project assumes access to AWS services (S3, Redshift, EMR), Databricks, Pyspark, Jira, and GitHub.
* The project assumes that all the data has already been collected and can be moved to S3.

**System Features and Requirements**

**a. Functional Requirements**:

* Identify the disease with the maximum number of claims.
* Find subscribers below the age of 30 who have subscribed to any subgroup.
* Determine the group with the maximum number of subgroups.
* Identify the hospital serving the highest number of patients.
* Determine the subgroups with the most subscriptions.
* Count the total number of rejected claims.
* Identify the city with the highest number of claims.
* Determine whether subscribers prefer government or private policies.
* Calculate the average monthly premium paid by subscribers.
* Identify the most profitable group.
* List patients under 18 admitted for cancer.
* List patients with cashless insurance and charges greater than or equal to Rs. 50,000.
* List female patients over 40 who had knee surgery in the past year.

**b. External Interface Requirements:**

* **User Interface:** The system will provide user interfaces for data analysis and visualization using Databricks. At the end the data created by the pipelines will be dumped to the output table for each use cases/functionality listed above.
* **Hardware Interface:** The system will interact with AWS services (S3, Redshift) and EMR.
* **Software Interface:** The system will utilize Pyspark for data processing and analysis. Data will be transferred between AWS services, Databricks, and Pyspark. All of the code will be pushed to the GitHub using Pyspark and the ETL will be running on EMR.
* **Communications Interface**: As this project is completed with the help of team of 7 members with team lead, manager and developers most communications were done through Microsoft teams, email as well as various zoom meetings with the stakeholders.

**c. System Features:**

* **Data cleaning modules:** The data loaded in the RedShift must be cleaned as there should not be any null values as well as duplicated values.
* **Schema Design modules:** There should be a schema design document for each target table showing the relationship between two or more tables with their primary and foreign key identified.
* **Result generation modules:** There should be a separate redshift table for each use case output in a red shift schema.

**d. Nonfunctional Requirements**:

* **Performance:** The system must handle large volumes of data efficiently.
* **Safety:** Data security and privacy must be maintained during data processing.
* **Security:** Access to AWS services, Databricks, and data repositories must be secure.
* **Usability:** Databricks visualizations should be intuitive for users.
* **Scalability:** The solution should scale to accommodate growing data.