**Requirements Specifications Document**

# Introduction - *This introduction is very important as it sets expectations that we will come back to throughout the SRS.*

## Purpose - *The goal of this project is to create data pipelines for the Health Care insurance company that will enable the company to make appropriate business strategies to enhance their revenue by analysing customers behaviours and send offers and royalties to customers respectively.*

## Intended Audience and Use - *Project Manager, Software developers, Data Engineers, Data Scientists, Business Analysts, Test Engineers.*

## Product Scope - *NA*

## Definitions and Acronyms - *NA*

# Overall Description - *Your next step is to give a description of what you’re going to build.*

* 1. DATASET CREATION: As a first step, we will upload the given sample data on AWS S3 in a folder. We will start by creating a bucket which will act as a dedicated repository for our data that will be used in our project. Within this bucket we shall create multiple folders as per our needs and requirement.
  2. DATA CLEANING: In this process we will check the given dataset for any duplicate, incomplete, mislabeled, incorrect and corrupted data. Check for null values. Check for incorrect file formats. We will identify these issues and rectify them using pyspark on Databricks. Then we will write the cleaned data files back into AWS S3 and run crawlers on these data to create tables to visualize them using AWS Athena and check if the uploaded data are consistent, correct and visible.
  3. DATA ANALYSIS AND VISUALIZATION: Next we will generate a pyspark code to create a dataframe on these cleaned data sets on Databricks notebook. We will analyze these tables as per the given use-cases, transform and visualize as per the specified requirements.
  4. RESULT CREATION ON REDSHIFT: The resultant output of the above transformation are saved as tables in the Redshift database for business analysis. These datas will be used by the Health Care insurance company that will enable the company to make appropriate business strategies to enhance their revenue by analysing customers behaviours and send offers and royalties to customers respectively.

# System Features and Requirements -*In order for your development team to meet the requirements properly, we must include as much detail as possible. This can feel overwhelming but becomes easier as you break down your requirements into categories.*

## Functional Requirements –

## We will be using the following use-cases as our functional requirements:

## *Which disease has a maximum number of claims.*

## *Find those Subscribers having age less than 30 and they subscribe any subgroup*

## *Find out which group has maximum subgroups.*

## *Find out hospital which serve most number of patients*

## *Find out which subgroups subscribe most number of times*

## *Find out total number of claims which were rejected*

## *From where most claims are coming (city)*

## *Which groups of policies subscriber subscribe mostly Government or private*

## *Average monthly premium subscriber pay to insurance company.*

## *Find out Which group is most profitable*

## *List all the patients below age of 18 who admit for cancer*

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

### Hardware: NA

### Software: For this project we will be using below listed softwares:

1. AWS S3: for storing initial datasets, processed data and the other resultant output
2. AWS Redshift: for data warehousing and data loading
3. Databricks: use databricks notebook for data analysis and visualization.
4. Pyspark: for coding purpose
5. Jira: To track the progress of the project
6. GitHub: as a data repository for project related documents, resources, progress reports, etc.

### Communications: NA

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

## Nonfunctional Requirements - *Nonfunctional requirements, which help ensure that a product will work the way users and other stakeholders expect it to, can be just as important as functional ones. These may include:*

### Performance requirements:

### Safety requirements

### Security requirements: We shall need following users and roles access:

USER: IAM user : Access key , Secret key

ROLE: redshift\_admin: login credentials (user name, password)

### Usability requirements

### Scalability requirements

## 