Template - Requirements Specifications Document

# Introduction - *A health insurance company is having trouble increasing its revenue and better understanding its clientele. As a result, it is looking to the Big Data Ecosystem for assistance in analyzing competitor firm data that it has obtained from a variety of sources, including third-party sources, and scraping. Through this analysis, the company will be able to monitor customer behavior and conditions and tailor offers to encourage clients to purchase insurance policies. The insurance company will research on the data and will forecast the possible solution through the help of data analyst.*

## Purpose – *The project aims to build data pipelines for the health care insurance firm so that it may analyze customer behavior, deliver offers and royalties to clients, and make suitable business decisions to increase income.*

## Intended Audience and Use – *The audience for the project are developers, tester, project manage and data analyst.*

## Product Scope - A Health Care insurance company

## ( *What are the benefits, objectives, and goals we intend to have for this product? This should relate to overall business goals, especially if teams outside of development will have access to the SRS.)*

## Definitions and Acronyms -*Clearly define all key terms, acronyms, and abbreviations used in the SRS. This will help eliminate any ambiguity and ensure that all parties can easily understand the document.*

# Overall Description –

# *The project will stats as follows,*

* + - 1. *Dataset creation*
      2. *Data cleaning using pyspark in Databriks*
      3. *Table creation using AWS Redshift*
      4. *Mapping multiple relational tables into dimensional tables using schemas design doc*
      5. *Creation of final dimensional dataset using Databriks*
      6. *Publishing the dimensional tables into AWS Redshift*
      7. *Uploading all the documents related to the project into GitHub*

## User Needs - *Describe who will use the product and how. Understanding the various users of the product and their needs is a critical part of the SRS writing process.*

## Assumptions and Dependencies - *What are we assuming will be true? Understating and laying out these assumptions ahead of time will help with headaches later. Are we assuming current technology? Are we basing this on a Windows framework? We need to take stock of these technical assumptions to better understand where our product might fail or not operate perfectly.*

# 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 – *The functional requirements are:*

## *find out the disease has a maximum number of claims, those subscribers having age less than 30 and they subscribe any subgroup, which group has maximum subgroups, hospital which serve most number of patients, which subgroups subscribe most number of times, 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, Which group is most profitable, List all the patients below age of 18 who admit for cancer, list patients who have cashless insurance and have total charges greater than or equal for Rs. 50,000, List female patients over the age of 40 that have undergone knee surgery in the past year.*

## 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:*

### User

### Hardware

### Software (AWS S3, AWS Redshift, Databricks, AWS EMR Studio, Pyspark, Jira, GitHub)

### Communications

## 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

*Security requirements are IAM role, IAM Access and Security keys.*

### Usability requirements

### Scalability requirements

## 