Requirements Specifications Document

•Introduction - *This introduction is very important as it sets expectations that we will come back to throughout the SRS(software requirements specification).*

A healthcare insurance company is having trouble catering to customers' needs so they would like to use the big data ecosystem to analyze the customer of their competitors. They would receive that information from various sources, mostly by data scrapping and some from third parties. They would like to use those data to track the behavior, and conditions of the customers and overall to understand their customer's needs better so they can cater to their needs more prominently as well as calculate policies that the customers have purchased in the past.

•Intended Audience and Use - *Define who in your organization will have access to the SRS and how they should use it. This may include developers, testers, and project managers.*

The developers, testers and project manager will have access to the SRC. Our developers will be developing the pipeline, the customers will be used as test subjects while using it to create test cases, and the project managers will be seeing their progress as well as making sure that everything is running smoothly according to the business objectives.

•Product Scope - *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.*

The benefits will be enormous as achieving this will improve the revenue of the company as well as make the customers happy by catering to their specific needs. The long-term goal is to keep increasing the customers as well as the revenue with the customer royalties.

•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.*

SRC- Software Requirements specification.

•Overall Description - *Your next step is to give a description of what you're going to build. Why is this product needed? Who is it for? Is it a new product? Is it an add-on to a product you've already created? Is this going to integrate with another product? Understanding and getting your team aligned on the answers to these questions on the front end makes creating the product much easier and more efficient for everyone involved.*

• 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.*

The company will possibly implement the product to the customers by giving them a questionnaire and matching them with the insurance policy most fitted for them.

•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 - *Functional requirements are essential to your product because, as the name implies, they provide some sort of functionality. Asking yourself questions such as "does this add to my tool's functionality?" or "what function does this provide?" can help with this process. You may also have requirements that outline how your software will interact with other tools*

The system shall provide the functionality to filter subscribers based on their age and their subscriptions to any subgroup.

The system shall be able to find which group has maximum subgroups.

The system shall be able to find the hospital with the most number of patients and also which type of disease has max number of claims.

The system shall be able to find which subgroup subscribes the most number of times.
The system shall be able to find the total number of claims which were rejected.
The system shall find the city with the most claims.
The system shall be able to find the policies subscribers subscribe mostly to government or private.
The system shall be able to find the amount that the policy subscriber pays to the insurance company.
The system shall find which group brings them the most revenue.
The system shall find the patients below 18 who are admitted for cancer.
The system shall list all patients who have cashless insurance and have total charges greater than or equal to $50, 000.
The system shall list female patients over the age of 40 who 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- A company that will use the system to bring in more revenues.

•Hardware

•Software - SQL, S3, Databricks, Jupiter Notebook, Pyspark, Hive, AWS

•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:*

The system shall give reliable information.
The system shall be protected against malicious intent.
The system shall be easy to use.
The system shall bring in more revenues.