Template - Requirements Specifications Document

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

## Purpose -*Define the purpose of these requirements here.*

## 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.*

## 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.*

## 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 –

A Health Care insurance company aims to boost revenue and improve customer understanding using Big Data. They plan to analyze competitor data from various sources via scraping and third-party channels. The analysis intends to monitor customer behavior, enabling customized insurance offers and royalties for policyholders, ultimately driving increased revenue. The project goal involves constructing data pipelines to facilitate strategic decision-making, leading to enhanced revenue through customer behavior analysis, personalized offers, and royalty distributions.

# *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.*

## 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 –

● Identify the disease with the highest number of claims.

● Locate subscribers under 30 who have joined any subgroup.

● Determine the group with the most subgroups.

● Identify the hospital treating the highest number of patients.

● Find the subgroup with the highest subscription frequency.

● Calculate the total count of rejected claims.

● Discover the city with the most incoming claims.

● Analyze whether subscribers prefer Government or private policy groups.

● Calculate the average monthly premium paid by subscribers.

● Find the most profitable policy group.

● List patients under 18 admitted for cancer.

● Display patients with cashless insurance and charges ≥ Rs. 50,000.

● Show females above 40 who had knee surgery in the last 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:*

● AWS S3

● AWS Redshift

● Databricks

● AWS EMR Studio

● Pyspark

● Jira

● GitHub

## 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

### Usability requirements

### Scalability requirements

## 