**Requirements Specifications Document**

# **Introduction**

In the rapidly evolving landscape of healthcare insurance, companies are constantly seeking innovative ways to deepen their understanding of their customer base and target market. Gaining a strategic advantage is crucial for staying ahead and increasing market share. Data, being a pivotal asset in today's business environment, is key to unlocking a roadmap for success, making it indispensable in today’s tech-driven world.

This project leverages the Big Data Ecosystem to analyze competitors' data, providing critical insights into market trends and customer behaviors for the healthcare insurance company. By understanding competitors' strategies and customer interactions, the company can refine its offerings, enhance customer satisfaction, reveal significant patterns, and identify inefficiencies overlooked by traditional methods.

Thus, by employing advanced data processing techniques and scalable cloud technologies, this project aims to transform raw data into actionable intelligence, empowering the company to make data-driven decisions and strengthen its competitive advantage.

1. **Purpose**

This project aims to establish a robust data pipeline infrastructure utilizing the Big Data Ecosystem. This system will facilitate the comprehensive analysis of competitor data, enabling the company to gain a deeper understanding of market trends and customer behavior. These insights will be instrumental in optimizing our market positioning. By leveraging this knowledge, we can strategically tailor insurance offerings, refine marketing campaigns, and ultimately drive customer acquisition and revenue growth.

1. **Intended Audience and Use**

The intended audience for this SRS includes developers, data engineers, data analysts, testers, and project managers. Developers and data engineers will use it to understand the system requirements and architecture. Data analysts will refer to it to understand the data processing and analysis requirements. Testers will use it to develop test cases, and project managers will use it to track project progress and ensure that the requirements are being met.

1. **Product Scope**

This project focuses on developing data pipelines utilizing the Big Data Ecosystem (AWS S3, AWS Redshift, Databricks, and Spark) to clean, process, and generate valuable insights.

**Goal**

The goal is to streamline data processing to encourage policy purchases, increase sales, and ultimately generate more revenue.

**Objective**

The objective is to create data pipelines that implement sophisticated data processing techniques to analyze competitors' data, identify customer behavior, market trends, and provide tailored services that meet customer needs more effectively.

**Strategy**

* Design and implement data pipelines using:
* AWS S3 (Amazon Simple Storage Service): Serves as a data lake for raw data obtained from scrapping and third-party sources.
* AWS Redshift: Function as the data warehouse to store cleaned and processed data, acting as the central repository for structured data to enable complex queries and analytics.
* Databricks: Utilize for data cleaning, transformation, and analysis. Its integration with Spark allows for scalable and distributed data processing.
* AWS EMR (Elastic Map Reduce): Manage and execute big data processing tasks, serving as a potential alternative to Databricks based on scalability requirements.
* Jira (Project Management Software): This will be used to create tasks, track progress, and manage workflow for development, data cleaning, and analysis activities.
* GitHub: GitHub will be used to manage the source for data cleaning, transformation, and analysis modules. Further, it ensures that changes are tracked, and different versions of the code are maintained, promoting collaboration and code quality.

1. **Definitions and Acronyms** **(Remaining)**

* SRS: Software Requirements Specification
* ETL: Extract, Transform, Load
* IP: Internet Protocol
* Pyspark: Python API for Apache Spark

1. **Overall Description**

This project introduces a new data analytics platform designed to empower a Healthcare insurance company to gain a competitive edge. By leveraging Big Data technologies and integrating with the company's existing AWS infrastructure, the platform will analyze competitor data collected from various sources. These valuable insights will be used to:

* Create targeted offers that attract new customers and increase policy purchases.
* Calculate royalties for past customers, potentially leading to higher customer satisfaction and retention.

1. **User Needs**

Understanding the different users and their needs is crucial for the success of this project. Here's a breakdown of the key user groups and how they will interact with the product:

* Business Analysts: They rely on detailed reports and interactive dashboards to gain insights into customer behavior, identify market trends, and track revenue growth.
* Data Engineers: Their primary need is robust data pipelines that efficiently process and analyze large volumes of competitor data.
* Developers: Clear system requirements and well-defined architecture are essential for them to build, maintain, and enhance the platform.
* Executives: They require high-level summaries and actionable insights to make strategic decisions that drive business growth.

1. **Assumptions and Dependencies**

**Assumptions:**

* The data will be provided in a semi-structured format (e.g., CSV, JSON) and stored in an existing AWS S3 bucket.
* The project will utilize AWS Redshift for data warehousing and Databricks for data processing and analysis.
* The development team has the necessary access and expertise to work with AWS services and Databricks.
* The data cleaning process will effectively address any inconsistencies present in the data.

**Dependencies:**

* Availability of AWS Infrastructure and Databricks: The successful operation of the platform relies on the continued availability of these cloud-based services.
* Timely Access to Data: The platform's effectiveness hinges on receiving data from various sources in a timely manner.
* Collaboration between Teams: Smooth collaboration between development, data engineering, and business analysis teams is critical for project success.

1. **System Features and Requirements**

System features and requirements outline the essential functions and capabilities that the system must have to achieve its objectives. They serve as the foundation for designing, developing, and deploying the system, ensuring it meets the needs of the stakeholders. These requirements are crucial as they define what the system must do (functional requirements) and how it should perform (nonfunctional requirements), guiding the entire development process and ensuring alignment with business goals.

1. **Functional Requirements**

* Data Ingestion: The system should be capable of ingesting raw data from various sources, including web scraping and third-party datasets, into AWS S3.
* Data Cleaning and Transformation: Utilize Databricks to clean and transform the ingested data. This includes handling missing values, removing duplicates, and formatting data appropriately.
* Data Storage: Store the cleaned and transformed data in AWS Redshift for efficient querying and analysis.
* Data Analysis and Reporting: Implement analytical queries in Redshift to generate insights, such as identifying customer behavior, market trends, and other critical business metrics.
* Pipeline Automation: Automate the data pipeline process to ensure timely updates and minimize manual intervention.
* Visualization: Integrate with visualization tools to present data insights in an understandable format for stakeholders.

1. **External Interface Requirements**

External interface requirements specify how a system will interact with external entities, such as users, hardware, software, and communication systems. These requirements ensure that the system can effectively communicate, exchange data, and function within its intended environment.

* 1. **User Interfaces**
* Clearly defined roles for data engineers, data analysts, and business analysts.
* Access levels and functionalities tailored to each user role.
* User-friendly interfaces for querying data and visualizing insights.
* Dashboards showcasing key metrics and trends, tailored to the needs of different roles.
  1. **Hardware Interfaces**
* Server Infrastructure: Provision AWS EC2 instances with sufficient computing power, memory, and storage to handle data processing workloads. Ensure that the infrastructure can scale dynamically based on demand.
* Network Requirements: Ensure reliable and high-speed network connectivity between different components of the Big Data Ecosystem (AWS S3, AWS Redshift, Databricks, EMR) and external data sources.
* Storage Solutions: Utilize AWS S3 for scalable storage of raw and processed data. Ensure that the storage solutions are optimized for performance and cost-efficiency.
  1. **Software Interfaces**
* Big Data Processing Tools: Use Databricks for data cleaning, transformation, and analysis. Integrate with PySpark to leverage distributed data processing capabilities.
* Data Warehousing: Implement AWS Redshift as the central data warehouse for storing cleaned and processed data. Ensure it is optimized for complex querying and analytics.
* Data Integration Tools: Utilize AWS Glue or similar ETL tools to automate the data ingestion and transformation processes.
* Visualization Tools: Integrate with visualization platforms like AWS QuickSight or Tableau to provide interactive dashboards and reports for stakeholders.
* Version Control: Use GitHub for version control of codebases, including data pipelines, ETL scripts, and analysis modules. Ensure that the repository is well-organized and documented.
* Project Management Software: Use Jira for project management, task tracking, and workflow management. Create user stories, tasks, and sprints to manage development and testing activities.
  1. **Communication Interfaces**
* APIs for Data Integration: Implement secure APIs to facilitate seamless data transfer between different system components (e.g., between AWS S3 and Redshift, Redshift and Databricks).
* Data Transfer Protocols: Use secure and efficient data transfer protocols such as HTTPS and SFTP for moving data between systems and ensuring data integrity and security.
* Notification and Alert Systems: Set up notification mechanisms to alert users and administrators about pipeline statuses, errors, and completion of data processing tasks. This can be done using AWS SNS (Simple Notification Service) or similar tools.
* Inter-Service Communication: Ensure robust communication between various AWS services (e.g., S3, Redshift, EMR, Databricks) to maintain data flow and process integrity.
* Collaboration Tools: Implement collaboration tools (e.g., Slack, Microsoft Teams) for effective communication among team members, especially for remote collaboration.

1. **System Features:**

The system will feature automated data pipelines for continuous ingestion, cleaning, and processing of data, ensuring seamless data flow. Real-time data analysis and reporting will be performed on the ingested data, providing timely insights. Scalability will be a core feature, allowing the system to adjust based on data volume and processing requirements. Robust security measures will be implemented to protect data and ensure compliance with relevant regulations. Additionally, the system will include role-based access control and comprehensive user management to maintain secure and organized user interactions.

1. **Nonfunctional Requirements**

Nonfunctional requirements are critical to ensuring the system's overall effectiveness, reliability, and user satisfaction. They define the quality attributes and operational constraints of the system, which are essential for achieving performance efficiency, security, usability, and scalability. Addressing these requirements is crucial for the project's success, as they impact user experience and system robustness.

1. **Performance Requirements:**

* Latency: Ensure low latency in data processing and querying to provide near real-time insights.
* Throughput: Handle large volumes of data efficiently without performance degradation.

1. **Safety Requirements:**

* Data Backup: Regular data backups to prevent data loss.
* Failure Recovery: Implement mechanisms for quick recovery in case of system failures.

1. **Security Requirements:**

* Data Encryption: Encrypt data at rest and in transit to ensure data privacy and security.
* Access Control: Implement strict access control policies to protect sensitive data.
* Compliance: Ensure compliance with industry standards and regulations, such as HIPAA.

1. **Usability Requirements:**

* User Interface: Design an intuitive and easy-to-use interface for data access and visualization.
* Documentation: Provide comprehensive documentation and user guides for system users.

1. **Scalability Requirements:**

* Elasticity: Enable elastic scaling of resources based on data processing needs.
* Modularity: Design the system in a modular way to allow easy addition of new features and components.