**Software Requirements Specification (SRS)**

**1. Introduction**

**1.1 Purpose**

This Software Requirements Specification (SRS) defines the functional and non‑functional requirements for building a Big‑Data analytics platform for **Acme Health Insurance** (pseudonym). The platform will ingest competitor health‑insurance data, cleanse and store it in AWS Redshift, generate business insights, and expose results to analysts and downstream systems. The goal is to equip business teams with actionable intelligence that increases revenue through targeted offers, optimized premiums, and customer loyalty royalties.

**1.2 Intended Audience and Use**

| **Role** | **Intended Use** |
| --- | --- |
| **Project Manager** | Scope validation, milestone tracking, risk management |
| **Data Engineers** | Design, develop, and deploy data‑pipelines that satisfy all requirements |
| **Data Analysts / Actuaries** | Consume Redshift output tables & visualizations to guide strategy |
| **QA Engineers** | Author and execute test plans that map to each requirement |
| **DevOps / Cloud Ops** | Provision & monitor AWS infrastructure (S3, Redshift, EMR, Databricks) |
| **Security & Compliance** | Verify HIPAA/PII controls, encryption, audit logging |

**1.3 Product Scope**

* Automates the collection, cleansing, and analysis of large volumes of insurance-related data using AWS big data services.
* Implements scalable ETL/ELT pipelines using PySpark on Databricks or AWS EMR.
* Ingests and processes seven source datasets (e.g., patients, subscribers, claims).
* Loads cleaned data into Redshift’s project\_staging schema.
* Derives over 12 business-critical metrics and stores results in the project\_output schema.
* Supports key business objectives:
* Identifying customer behavior trends
* Calculating policyholder royalties
* Generating personalized insurance offers
* Improving business decisions through insight-driven dashboards
* Enables CI/CD development using GitHub and agile tracking via Jira.
* Delivers visual dashboards and result snapshots through Databricks.
* Facilitates exportable reports and Redshift query access for business users.

**1.4 Definitions, Acronyms & Abbreviations**

| **Term** | **Definition** |
| --- | --- |
| **ETL / ELT** | Extract, Transform, Load / Extract, Load, Transform |
| **S3** | Amazon Simple Storage Service |
| **EMR** | Elastic MapReduce managed Hadoop/Spark service |
| **HIPAA** | U.S. Health Insurance Portability and Accountability Act |
| **PHI** | Protected Health Information |
| **PK / FK** | Primary Key / Foreign Key |
| **CSV** | Comma‑Separated Values file |

**2. Overall Description**

**2.1 Product Perspective**

The solution is a new, standalone analytics pipeline that interfaces with the following external systems:

* **AWS S3** – landing zone for raw CSVs (folder: *input-data*)
* **AWS Redshift** – enterprise data‑warehouse target
* **Databricks Community Edition / EMR Studio** – Spark runtime for development & scheduled jobs
* **GitHub** – source‑code repository & CI/CD
* **Jira** – agile project tracking

**2.2 Product Functions**

* Securely receive CSV files and store raw copies
* Automated schema inference & data‑type harmonization
* Data quality assessment (null counts, duplicate counts)
* Data cleansing (null replacement, duplicate removal)
* Dimension & fact table generation with PK/FK constraints
* Analytical query execution producing 13 business KPIs
* Persist KPI result sets in dedicated Redshift tables
* Visual dashboards for executive stakeholders

**2.3 User Needs**

* **Executives** need near‑real‑time insight into claim trends and profitability.
* **Underwriters** need subgroup‑level behaviour metrics to tailor policies and premiums.
* **Marketing** needs lists of young (<30) subscribers and cancer patients for campaign targeting.
* **Data Engineers** need reproducible, version‑controlled pipelines deployable to the cloud.
* **Business Analysts** require insights to support revenue-enhancing strategies.

**2.4 Assumptions and Dependencies**

1. AWS credentials and Redshift cluster (ra3 or better) are provisioned.
2. Sample CSVs are representative of production volume.
3. All PHI fields are tokenised in the sample; production data will require additional masking.
4. Stakeholders accept Databricks CE for prototype; production will migrate to EMR if required.

**3. System Features and Requirements**

**3.1 Functional Requirements**

* Upload sample datasets to AWS S3.
* Perform data cleaning: handle nulls, drop duplicates.
* Store cleaned data in AWS Redshift tables.
* Execute analytical queries to generate the required results.
* Store output of each use case into Redshift Project-Output schema.

**3.2 Use Case Requirements**

* Determine disease with the highest number of claims.
* Identify subscribers under 30 subscribed to any subgroup.
* Identify group with the maximum number of subgroups.
* Find the hospital serving the most patients.
* Determine the most subscribed subgroup.
* Calculate the number of rejected claims.
* Identify the city with the most claims.
* Determine whether subscribers prefer government or private policies.
* Calculate average monthly premium per subscriber.
* Determine the most profitable group.
* List patients under 18 admitted for cancer.
* List of patients with cashless insurance and charges >= 50,000.
* List female patients over 40 who underwent knee surgery in the past year.

**3.3 External Interface Requirements**

* **User Interface**: Databricks notebooks for development/testing.
* **Software Interface**: AWS CLI, Spark, Python.
* **Database Interface**: Redshift JDBC connection.

**3.4 Nonfunctional Requirements**

* **Performance**: Queries should return results within seconds for clean data.
* **Security**: Role-based access on AWS and GitHub.
* **Usability**: Code and results should be well-documented and reproducible.
* **Scalability**: Infrastructure should be scalable to process larger datasets.