Product Requirements and Specification Document (PRD)

Project Name

HealthTrends - Distributed Health Data Analyzer

Description

HealthTrends is a distributed data pipeline for analyzing anonymized health records. The system ingests data via Kafka, stores it in HDFS, and processes it with PySpark to extract trends and generate summary CSV reports. The project emphasizes practical big-data skills in the healthcare domain, using Python for orchestration and data processing.

---

1. Objectives

Objective	Description
Distributed Ingestion	Efficiently ingest anonymized health data streams using Kafka.
Scalable Storage	Store large volumes of health records in HDFS.
Trend Analysis	Use PySpark to process data and extract health trends.
Automated Workflow	Orchestrate the pipeline with Python scripts.
Data Artifacts	Output summary reports in CSV format.

---

2. Scope

- **In Scope:**
  - Data ingestion (Kafka)
  - Distributed storage (HDFS)
  - Data processing (PySpark)
  - Workflow orchestration (Python)
  - CSV summary report generation
- **Out of Scope:**
  - User interfaces
  - Real-time dashboards
  - Non-specified technologies

---

3. Functional Requirements

ID	Requirement

FR1	Ingest anonymized health records via Kafka topics.
FR2	Store ingested data in HDFS in a structured format (e.g., Parquet/CSV).
FR3	Process stored data with PySpark to extract key health trends.
FR4	Generate summary reports (CSV) with trend statistics (e.g., averages, counts).
FR5	Orchestrate the pipeline using Python scripts for end-to-end automation.
FR6	Ensure data privacy by handling only anonymized records.

---

4. Non-Functional Requirements

ID	Requirement
NFR1	Pipeline must handle at least 1 million records per batch.
NFR2	Processing time per batch should not exceed 30 minutes.
NFR3	System must be fault-tolerant and recover from node failures.
NFR4	All data must remain anonymized throughout the pipeline.
NFR5	Use only Python, Kafka, HDFS, and PySpark.

---

5. System Architecture

```
[Kafka Producer] --> [Kafka Broker] --> [Python Consumer] --> [HDFS] --> [PySpark J
```

---

6. Data Flow

1. **Kafka Producer**: Publishes anonymized health records to Kafka topics.
2. **Python Consumer**: Consumes records from Kafka, writes to HDFS.
3. **HDFS**: Stores raw health data files.
4. **PySpark Job**: Reads from HDFS, processes data, extracts trends.
5. **CSV Output**: PySpark writes summary reports to HDFS/local storage.

---

7. Data Specifications

Data Element	Type	Example/Description
patient_id	String	Anonymized unique identifier
age	Integer	Patient age
gender	String	‘M’, ‘F’, ‘O’
diagnosis_code	String	ICD-10 code
visit_date	Date	YYYY-MM-DD

treatment_code	String	Procedure code
outcome	String	‘recovered’, ‘ongoing’, ‘deceased’

---

8. Processing & Reporting Requirements

Trend/Metric	Description
Top diagnoses	Most frequent diagnosis codes
Age distribution	Patient count by age group
Gender breakdown	Patient count by gender
Treatment outcomes	Outcome statistics per diagnosis
Visit frequency	Number of visits per time period

---

9. Implementation Guidelines

- **Kafka**: Use Python (e.g., `kafka-python`) for producer/consumer.
- **HDFS**: Store data in partitioned directories by date.
- **PySpark**: Use DataFrame API for processing and aggregation.
- **Python Orchestration**: Use scripts to automate each pipeline stage.
- **CSV Output**: Ensure reports are well-structured and documented.

---

10. Acceptance Criteria

ID	Criteria
AC1	Pipeline ingests, stores, processes, and outputs data as specified.
AC2	All summary reports are generated in CSV format and match requirements.
AC3	System handles specified data volume and performance targets.
AC4	Only anonymized data is processed and output.
AC5	Uses only approved technologies.

---

11. Milestones

Milestone	Description	Due Date (suggested)
Environment Setup	Kafka, HDFS, PySpark configured	Week 1
Data Ingestion Complete	Kafka to HDFS pipeline operational	Week 2
Processing Logic Ready	PySpark trend extraction working	Week 3
Reporting Complete	CSV summary generation	Week 4

Final Testing	End-to-end validation	Week 5

---

12. Risks & Mitigations

Risk	Mitigation
Data volume exceeds capacity	Test with sample data, scale cluster as needed
Data privacy breach	Use only anonymized datasets
Technology integration issues	Use standard libraries, document interfaces

---

13. Glossary

Term	Definition
HDFS	Hadoop Distributed File System
Kafka	Distributed event streaming platform
PySpark	Python API for Apache Spark
CSV	Comma-Separated Values file format

End of Document