Software Requirements Specification

Credit Card ETL Pipeline Manager

Version 1.0

Prepared by: Kenneth Copas

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to define the software requirements for the Credit Card Data Engineering Pipeline project. This system processes, transforms, and loads credit card data into a MySQL database using Apache Spark and a modular CLI interface.

## 1.2 Scope

This application provides an end-to-end data engineering pipeline that ingests raw financial data from files and APIs, cleans and transforms the data using PySpark, and stores it into a structured relational database. It includes a command-line interface for interaction.

## 1.3 Definitions, Acronyms, and Abbreviations

• CLI - Command Line Interface  
• SRS - Software Requirements Specification  
• PySpark - Python API for Apache Spark  
• JDBC - Java Database Connectivity

## 1.4 References

• IEEE Std 830-1998 - IEEE Recommended Practice for Software Requirements Specifications  
• https://spark.apache.org/docs/latest/api/python/  
• https://dev.mysql.com/doc/

## 1.5 Overview

The remainder of this document includes a detailed description of the system, including functional and non-functional requirements, use cases, and design constraints.

# 2. Overall Description

## 2.1 Product Perspective

The software is a standalone data engineering tool intended to operate on a local system. It interacts with MySQL through JDBC and handles data processing using Apache Spark.

## 2.2 Product Functions

• Read structured data from JSON/CSV files  
• Ingest API-sourced loan data  
• Transform and clean the data  
• Load the processed data into a MySQL database  
• Provide a CLI for executing and managing these workflows

## 2.3 User Characteristics

The intended users are data engineers and analysts with basic familiarity with Python, Spark, and SQL. Users will interact with the system via a command-line interface.

## 2.4 Constraints

• Python 3.10+ must be installed  
• MySQL Server must be accessible  
• System must include required dependencies: pandas, PySpark  
• Limited to single-machine operation in its current version

## 2.5 Assumptions and Dependencies

• The data input formats are consistent and follow expected schemas  
• Users have appropriate permissions to access local files and MySQL server  
• JDBC driver is correctly installed and referenced in the config

# 3. Specific Requirements

## 3.1 Functional Requirements

• The system shall read input data from `application\_data/` directory  
• The system shall validate and transform raw data into clean, structured formats  
• The system shall store transformed data in a MySQL database using user credentials  
• The system shall provide a CLI for executing transformations and managing data loading

## 3.2 External Interface Requirements

• User Interfaces: Command-line prompts and options via `cli\_manager.py`  
• Hardware Interfaces: Standard x86 machine running MacOS/Linux/Windows  
• Software Interfaces: PySpark, JDBC, MySQL Server  
• Communication Interfaces: File system and localhost database

## 3.3 Performance Requirements

The system should process datasets up to 100MB in under 30 seconds on a typical workstation. Data loading and transformation are expected to complete with minimal delay for most single-batch workflows.

## 3.4 Design Constraints

• Implementation language is Python  
• PySpark is used for all transformation logic  
• MySQL is the target relational database  
• Configuration is stored in a local JSON file

## 3.5 Software System Attributes

• Reliability: Assumes stable data and schema inputs  
• Availability: CLI can be launched as needed; not a long-running service  
• Maintainability: Codebase is modular and readable  
• Portability: Can run on any OS with Python and Spark installed