Software Requirements Specification

for

HR Data Analytics Platform

Version 1.0 approved

Prepared by Ebrahim Mamawala

Vellore Institute of Technology, Vellore

16th January 2025

Table of Contents

Table of Contents ii

Revision History iii

1. Introduction 1
   1. Purpose 1
   2. Document Conventions 1
   3. Intended Audience and Reading Suggestions — separate audience and reading suggestion 1
   4. Product Scope 2
   5. References 2
2. Overall Description 2
   1. Product Perspective 2
   2. Product Functions 2
   3. User Classes and Characteristics 2
   4. Operating Environment 3
   5. Design and Implementation Constraints 3
   6. User Documentation 3
   7. Assumptions and Dependencies 3
3. External Interface Requirements 3
   1. User Interfaces 3
   2. Hardware Interfaces 4
   3. Software Interfaces 4
   4. Communications Interfaces 4
4. System Features 4
   1. System Feature 1 4
   2. System Feature 2 (and so on) 5
5. Other Nonfunctional Requirements 5
   1. Performance Requirements 5
   2. Safety Requirements 5
   3. Security Requirements 5
   4. Software Quality Attributes 5
   5. Business Rules 6
6. Other Requirements 6

Appendix A: Glossary 6

Appendix B: Analysis Models 6

Appendix C: To Be Determined List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

This document specifies the software requirements for the HR Data Analytics Platform project.

- Write all the use cases of this project and mention that this project will focus on two use cases

The platform aims to provide HR professionals with actionable insights into employee attrition and organizational diversity. This SRS focuses on end to end implementation of the data analytics platform, including data pipeline development, dashboard creation, and AI-driven analytics. The described scope includes essential features for attrition prediction and diversity analysis.

## Document Conventions

* High-level requirements inherit the priority of their detailed counterparts unless explicitly stated.
* Key terms are highlighted in **bold** for emphasis.
* Agile development principles guide the organization and prioritization of tasks.

## Intended Audience and Reading Suggestions — separate audience and reading suggestion

* **Developers:** Focus on sections detailing the data pipeline and AI model requirements.
* **Project Managers:** Review the timeline, milestones, and measurable outcomes.
* **Testers:** Refer to testing and validation phases for quality assurance guidelines.
* **HR Professionals:** Utilize the dashboards to derive actionable insights.
* **Documentation Writers:** Ensure clarity and completeness in user guides and technical documentation.
* **Business Leaders**: Executives and department heads for strategic workforce planning.
* **Data Analysts**: For performing in-depth custom analyses on HR data.

It is recommended to begin with the overview and scope sections, followed by detailed requirements specific to each audience’s role.

## Product Scope

The HR Analytics Platform is designed to assist HR teams in decision-making by leveraging data-driven insights. Key objectives include improving employee retention and fostering workplace diversity. By aligning with corporate goals of enhancing workforce stability and inclusivity, this platform delivers both predictive analytics and real-time reporting.

## References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

# Overall Description

## Product Perspective

The product perspective for an HR Data Analytics Platform outlines how the platform fits into the overall ecosystem of an organization and interacts with existing systems, users, and processes. While not a follow-on or replacement for any existing system, it complements existing HR processes by offering advanced analytics. The platform integrates Python-based AI models for predictions, and Power BI and custom visualization libraries for visualization. The product interacts with existing HR databases via micro-services based APIs and is modular to ensure seamless future integrations.

Diagram—————— pending

## Product Functions

The platform’s primary functions are:

* **Data Ingestion and Processing**: Automates ETL tasks for structured HR data.
* **Attrition Prediction**: Identifies employees likely to leave the organization.
* **Diversity Analysis**: Offers insights into workforce gender and demographic diversity.
* **Interactive Dashboards**: Displays metrics and trends using Power BI for real-time insights.

Diagram—————— pending

## User Classes and Characteristics

The anticipated user groups are:

* **HR Managers**: Non-technical users relying on dashboards for decision-making. HR Managers require ease of use.
* **Data Analysts**: Technical users refining and analyzing datasets. Data Analysts need configurable filters and advanced analysis options.
* **IT Administrators**: Technical users maintaining the platform’s backend systems. IT Administrators focus on performance, security, and uptime.

## Operating Environment

The platform will operate in:

* **Hardware**: Cloud or on-premise servers with minimum configurations of 16 cores and 64 GB RAM.
* **Software**: Databricks (latest version), Python 3.9 or higher, Power BI, and SQL-based HR databases.
* **Operating Systems**: Windows Server 2019 or later, Linux (Ubuntu 20.04+).

## Design and Implementation Constraints

* It will utilize pre-existing HR databases with defined schemas.
* Development tools are limited to Python, Databricks, and Power BI.

## User Documentation

The following documentation will be provided:

* User Manuals: Step-by-step guides for HR managers.
* Technical Documentation: Details on APIs, data pipelines, and AI models for developers.

## Assumptions and Dependencies

* Dependencies include third-party libraries (e.g., Scikit-learn, Pandas) and Power BI licenses.
* The project assumes stable cloud services for hosting.

# External Interface Requirements

## User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

## Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

## Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

## Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

# System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

## System Feature 1

<Don’t really say “System Feature 1.” State the feature name in just a few words.>

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## System Feature 2 (and so on)

# Other Nonfunctional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

## Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>