DOUGLAS COLLEGE

COMMERCE & BUSINESS ADMINISTRATION
COMPUTING STUDIES & INFORMATION SYSTEMS
COMPUTER AND INFORMATION SYSTEMS (PBD)

CSIS 4495-050: APPLIED RESEARCH PROJECT

Project Proposal:

End-to-End Data Engineering Solution for HR Analytics

Team: Bruno do Nascimento Beserra | 300392300
Jay Clark Bermudez | 300380540 (Team Leader)
Matheus Filipe Figueiredo | 300389657

Instructor: Dr. Bambang Sarif

NEW WESTMINSTER/BC
FALL/2025

Contents

1.0 Introduction

This project looks at a challenge with Dayforce, a SaaS platform used to manage HR data like employee information and payroll. The problem is that Dayforce does not keep historical records. When an employee leaves, their data is deleted, and when updates are made, older records are replaced. This makes it hard to do historical analysis, track workforce trends, or study issues like employee turnover (Dayforce, 2024). This is not just a Dayforce issue, but a common limitation with HR SaaS platforms (Solutions, 2025).

Platform3 Solutions (Solutions, 2025) notes that not keeping payroll and HR records can lead to compliance issues, problems during audits, and even legal trouble. They stress that companies need a clear plan for keeping and archiving data so it stays available when needed and costs stay under control.

Research shows that without strong historical archives, companies struggle with workforce planning and decision-making (Madden, 2025). Using data engineering techniques like Slowly Changing Dimensions Type 2 and platforms like Databricks and Delta Lake can fix this problem by allowing data to be captured, stored, and analyzed over time (WJARR, 2025).

To solve this problem, the project will build a data pipeline that automatically collects, processes, and saves historical HR data. The pipeline will run in Databricks, using Python and PySpark for transformations and Delta Lake for reliable storage. On top of this, a web app will be built with Django and React to show the results of the analysis.

The finished system will help organizations keep and study HR history in a more efficient way. It will make storage use better, allow faster queries, support long-term workforce analysis, and improve decision-making by giving insights that are not possible with the current setup.

2.0 Project Goal

The goal of this project is to design and implement an end-to-end data engineering solution that preserves and enables analysis of historical HR data. The project addresses a key limitation in current HR data management, where employee records are deleted after termination or overwritten when changes occur, making historical analysis impossible.

To solve this, the team will develop a data pipeline that ingests daily data from Dayforce and applies data engineering techniques such as Slowly Changing Dimensions (Type 2) to track changes over time, the medallion architecture to structure data into quality layers, and Kimball data modelling to reduce redundancy and simplify queries.

The solution will be built in Databricks using Python, PySpark, and Delta Lake, and will be complemented by a Web application that highlights key workforce metrics using the data

available. Together, these components will demonstrate how historical HR records can be effectively preserved, organized, and analyzed to support long-term workforce insights.

3.0 Methodology

As outlined in the project goal, this study will develop an end-to-end data engineering solution to preserve and manage historical HR data. In the corporate world, businesses typically obtain their HR data from platforms such as Dayforce or other human resource management systems. When reports are needed, this data is often exported as a CSV file and analyzed using visualization tools. Since our team does not have direct access to such corporate data, we will use a Kaggle dataset containing information on 2 million employees (Kaggle, 2025), from which we will extract a subset as our sample dataset for this project. This data will be updated monthly, allowing us to track trends over a period of seven years. The pipeline will follow the medallion architecture (Databricks, 2020), with Bronze, Silver, and Gold layers ensuring both quality and traceability.

Figure 3.1 System Architecture

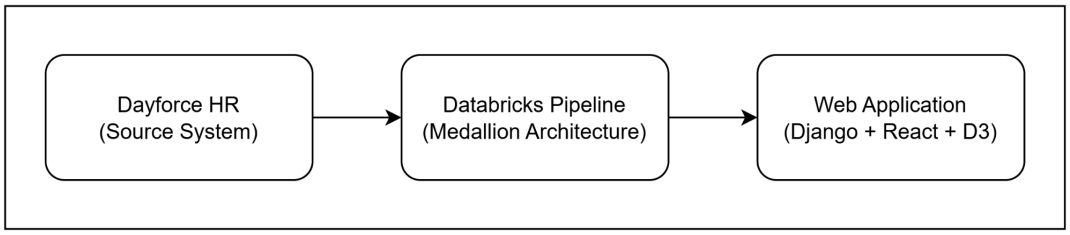

Figure 3.1 shows the overall system architecture. There are three (3) major stages. The raw data from Dayforce will be processed through a Databricks pipeline. Finally, one of the outputs of the system is a CSV file, which will then go through the web application for visualization.

Figure 3.2 Data Engineering Pipeline

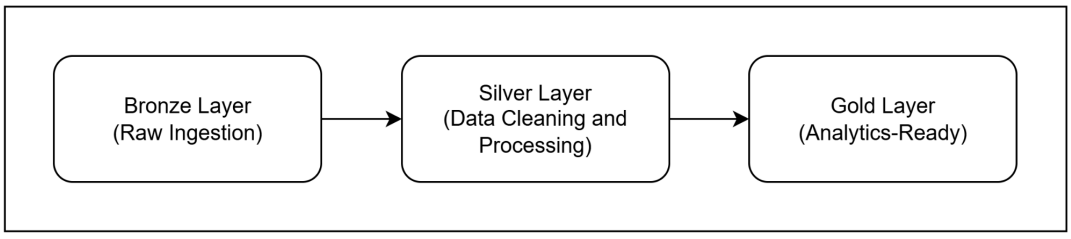

Figure 3.2 shows the data pipeline utilizing Databricks. The process begins with the ingestion of raw employee data from Dayforce (Dayforce, 2020) into the Bronze layer, capturing daily snapshots of the system. Data is then stored in a parquet format. The Silver layer will apply Slowly Changing Dimensions Type 2 (Asanka, 2021) to track historical changes, such as promotions, transfers, or terminations. Data cleaning will address missing values, duplicates, and inconsistent formats, while verifying key identifiers such as employee IDs. Each record will include timestamps and active/inactive flags to maintain historical accuracy.

The Gold layer will structure data for analysis using Kimball modelling (Nguyen, Pham, and Chin, 2020), creating fact and dimension tables to reduce redundancy and simplify queries. Implementation will be carried out in Databricks notebooks with Delta Lake features such as incremental ingestion, schema enforcement, and merge operations to ensure consistency.

Additionally, the output from Gold layer can be readily used for any analytics reports that can be utilized by Tableau, PowerBI, other business needs, and the custom web application developed by the team. This web application will serve as the primary interface for end users to access and explore the visuals.

Figure 3.3 Web Application Architecture

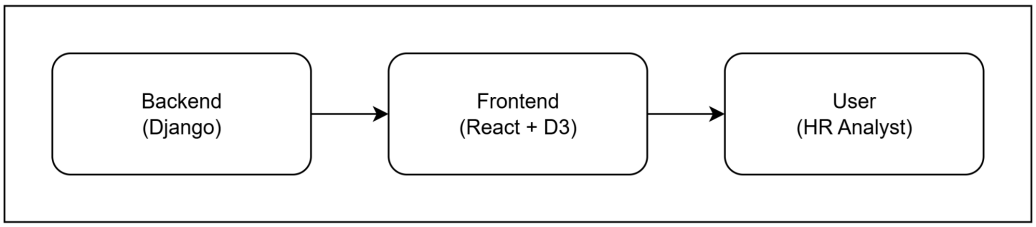

Figure 3.3 shows the architecture of the simple web application that complements with the output from the data engineering pipeline. The backend will use the Django framework, which supports quick development, strong security features, and an organized structure for handling data (Django Software Foundation, 2018). Django will take in the CSV files produced by the Gold layer of the pipeline, process the data, and provide results through APIs that the front end can access.

On the frontend, the app will be developed with ReactJS, since its component-based design makes it easy to update the interface whenever new data comes in (W3Schools, 2017). To make the results more meaningful, we will add D3, a library that specializes in interactive charts and graphs (D3 by Observable, 2020). This allows us to go beyond static visuals and create features like filters, drill-down options, and side-by-side comparisons.

The visualizations will highlight important HR metrics, such as headcount trends, departmental transfers, and payroll history. Presenting this information in an interactive and easy-to-read format will help managers and decision-makers spot patterns, track changes over time, and make better choices using historical data.

Development will follow best practices such as version control in Git (Microsoft, 2022) and thorough documentation. This methodology ensures the solution is robust, maintainable, and scalable, providing a framework that supports both current and future HR data analysis needs.

4.0 Technical Requirements

- **Data Platform:** Databricks
- **Data Pipeline and Processing:** Delta Lake, Python PySpark
- **Data Storage:** Delta Lake Bronze, Silver, Gold layers
- **Data Modelling:** Kimball dimensional modelling, Slowly Changing Dimensions (Type 2)
- **Backend:** Django (Python)
- **Frontend:** ReactJS, D3.js
- **Version Control:** Git

5.0 Project Plan and Timelines

5.1 Phase Overview and Milestones

Figure 5.1 Gantt Chart

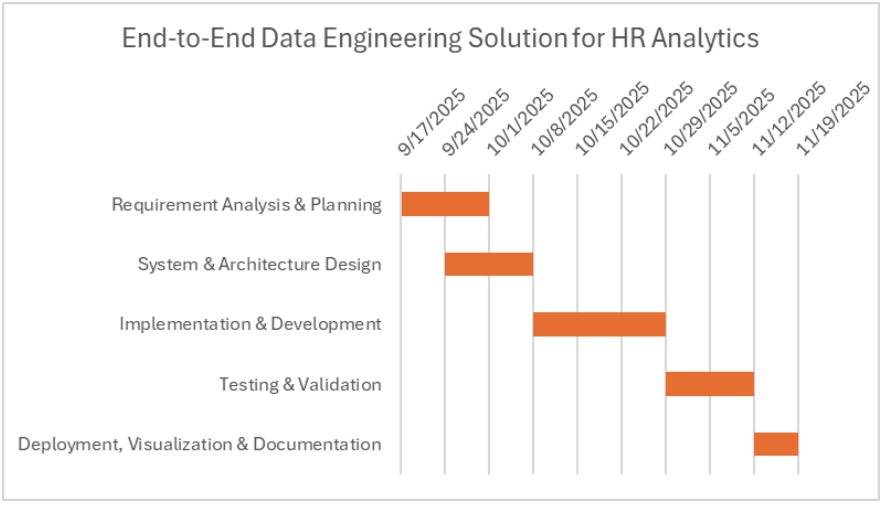

Figure 5.1 shows the Gantt chart, which outlines the five phases of the project: Requirement Analysis, System Design, Implementation, Testing, and Deployment, along with their timelines and dependencies. It shows how the earlier phases provide the foundation for development and testing, which ensures readiness for final deployment.

5.1.1 Requirement Analysis and Planning

The team will assess Dayforce's data limitations, focusing on record loss after updates or employee exits. From these observations, requirements such as daily ingestion, historical record preservation, and trend analysis will be defined. The scope will outline Databricks, PySpark, Delta Lake, and Django as core tools, with acceptance criteria based on accurate ingestion, retention, and longitudinal querfying.

5.1.2 System Design and Modelling

The system will be designed using the medallion architecture with Bronze, Silver, and Gold layers. SCD Type 2 logic will be specified to capture employee history, and Kimball-style modelling will define fact and dimension tables for analysis. Deliverables include schema definitions and a system design document. Additionally, the web application design will be planned, with Django handling backend processes and React/D3.js enabling interactive dashboards for visualization.

5.1.3 Implementation and Development

A Databricks pipeline will be built to ingest simulated Dayforce data into the Bronze layer, transform and apply SCD Type 2 in the Silver layer, and organize fact/dimension tables in the Gold layer. Delta Lake features such as incremental ingestion, schema enforcement, and merge operations will ensure reliability. The web application will be developed with the backend processing data and the frontend displaying HR metrics such as headcount trends and payroll history.

5.1.4 Testing and Validation

Testing will cover both the pipeline and the web application. Unit tests will confirm that ingestion and transformation processes work correctly, while integration tests will ensure data flows smoothly across all layers. User acceptance testing (UAT) will check that historical records are preserved accurately and that analysis tasks return correct results. The web application will also be tested for responsiveness and accuracy of visualizations.

5.1.5 Deployment, Visualization, and Documentation

The pipeline will be deployed in Databricks and connected to a custom web application for dashboards displaying metrics such as workforce trends, employee retention and turnover, and other key metrics. Final deliverables include technical documentation, versioned code in Git, a project report, and a presentation of the solution.

5.2 Weekly Work Plan and Schedule

Phase	Duration	Due Date	Key Deliverables
Requirement	2 weeks	Oct 1	• Approved project proposal

Analysis & Planning			● Draft use case diagram ● Initial dataset collection and exploration
System Design & Modelling	2 weeks	Oct 8	● System architecture design ● Data engineering pipeline design ● Web application architecture design ● UI/UX design ● Draft test cases ● Setup of GitHub repository ● Setup Databricks workspace
Implementation & Development	3 weeks	Oct 29	● Databricks ingestion and transformation pipeline ● Lakehouse configuration for Bronze, Silver, Gold layers ● Backend development ● Frontend development ● Unit testing of pipeline and components
Testing & Validation	2 weeks	Nov 12	● Test execution ● Regression testing for pipeline and web app ● Bug tracking and resolution ● Test summary report
Deployment, Visualization & Documentation	1 week	Nov 19	● Final pipeline deployment ● Web application deployment ● Project documentation ● Final project presentation and demo

5.3 Roles and Responsibilities

Jay Clark Bermudez - Project Manager & Back-end Developer

- Coordinate team meetings, progress, and ensure deadlines are met
- Serve as the main point of communication between the team, instructor, and other stakeholders
- Design and implement the backend system and APIs for the web application

Bruno do Nascimento Beserra - Data Engineer

- Extract, clean, and transform raw HR data for pipeline integration
- Implement the medallion architecture
- Collaborate with the backend developer to align data processing with the system requirements

Matheus Filipe Figueiredo - Front-end Developer & QA/Documentation

- Build the user interface for visualizing reports and analytics
- Develop and execute test cases to validate functionality and performance
- Track and document issues, prepare a user guide, and finalize project documentation

6.0 Project Contract

Project Agreement Contract:

This Project Agreement Contract was created on August 9th, 2025, referring to the project **End-to-End Data Engineering Solution for HR Analytics**, between the following Team:

1. Agreement to Project Scope and Timelines:

All team members agree to the project scope and timeline presented in *Section 5* of the project proposal.
All team members also agree on the project timeline and Gantt Chart as presented in *Section 5* of the project Proposal.

2. Meeting format and frequency:

All team members agreed to meet weekly every *Thursday, 6:30 pm to 9:30 pm*, during our *Applied Research Project* class.

All team members also agreed that if, for any reason, a member is unable to attend the class, they must notify the other members at least 5 days in advance, unless it's an unforeseen reason, such as sickness, family tragedy, or accident.

3. Responsibilities

All team members agreed to the responsibilities outlined in Section 5.3 of this proposal.

4. Agreement:

By signing this document, every member commits to working on the project, follows the scope, helps and assists each other, is present during the team meetings, understands its responsibilities and delivers the project on the due date.

5. Member Signatures:

Jay Clark Bermudez - 300380540 (Team Leader)
Date: 09/13/2025

Matheus Filipe Figueiredo - 300389657
Date: 09/13/2025

Bruno do Nascimento Beserra - 300392300
Date: 09/13/2025

7.0 Work Hours

Student Name: Bruno do Nascimento Beserra

Date	Number of Hours	Description of Work Done
09/09/2025	2	Initial Research on the topic (ResourceMatch)
09/09/2025	0.5	Write about the Project Goal (ResourceMatch)
09/10/2025	1.5	Write part of the Methodology (ResourceMatch)
09/12/2025	1.5	Initial Research on the new project,
09/11/2025	1	Create the project git Repository, Readme and folder Structure (ResourceMatch)
09/13/2025	4.5	Update the Project Goal, Tech Requirements, and Methodology for the new project
09/22/2025	1	Create new project git Repository, Readme and folder Structure
09/21/2025	1	Create databricks workspace
09/20/2025	1	Search for dataset to use in the project

Student Name: Jay Clark Bermudez

Date	Number of Hours	Description of Work Done
09/09/2025	1	Further research about the topic (ResourceMatch)
09/09/2025	0.5	Possible tech stacks to be used (ResourceMatch)
09/10/2025	2	Adding external partners, tech requirements, and the project timeline (ResourceMatch)
09/11/2025	0.5	Update document format (ResourceMatch)
09/13/2025	1	Initial research on the new project
09/14/2025	1	Update project Timeline for the new project
09/15/2025	0.5	Finalize project proposal for submission
09/21/2025	1	Changed PowerBI to custom web App on methodology and timelines
09/22/2025	1	Update methodology figures and description

09/23/2025	2	Specify roles and responsibilities, update milestones and deliverables
09/24/2025	1	Make changes in methodology figures and texts
09/24/2025	2	Final updates in the project plan and timelines before submission

Student Name: Matheus Filipe Figueiredo

Date	Number of Hours	Description of Work Done
09/09/2025	1.5	Initial work on the contract (ResourceMatch)
09/09/2025	0.5	Updated the contract (ResourceMatch)
09/11/2025	1.25	Initial introduction added
09/14/2025	0.5	Updated Introduction and Contract
09/18/2025	2	Updated the contract once again and changed the text to paragraphs, corrected text
09/22/2025	1	Updated the Introduction
09/24/2025	2.5	Searched for more references and updated the introduction with such references.

8.0 Acknowledgement

We would like to acknowledge our instructor, Dr. Bambang Sarif, for guidance and support throughout the project proposal.

9.0 References

Asanka, D. (2021). Implementing slowly changing dimensions (scds) in data warehouses. Retrieved from https://www.sqlshack.com/implementing-slowly-changing-dimensions-scds-in-data-warehouses/

Databricks, T. (2020). Medallion architecture. Retrieved from https://www.databricks.com/glossary/medallion-architecture

Dayforce. (2020). Dayforce main website. Retrieved from https://www.dayforce.com/

Microsoft. (2022). What is git? Retrieved from https://learn.microsoft.com/en-us/devops/develop/git/what-is-git

Nguyen, H., Pham, H., & Chin, C. (2020). The analytics setup guidebook. Holistics. Retrieved from https://www.holistics.io/books/setup-analytics/

D3 by Observable. (2020). What is d3? Retrieved from https://d3js.org/what-is-d3

Django Software Foundation. (2018). Django: The web framework for perfectionists with deadlines. Retrieved from https://www.djangoproject.com/

W3Schools. (2017). React introduction. Retrieved from https://www.w3schools.com/react/react_intro.asp

Kaggle, T. (2025). Hr dataset (multinational company). Retrieved from https://www.kaggle.com/datasets/rohitgrewal/hr-data-mnc

Dayforce. (2024). Personal data retention policies. Retrieved from https://help.dayforce.com/r/ImplementationGuide/Dayforce-Implementation-Guide/Personal-Data-Retention-Policies

HRBrain. (2024). Hr saas platform analytics and reporting. Retrieved from https://hrbrain.ai/blog/hr-saas-platform-analytics-and-reporting/

Madden, S. (2025). Combating turnover using people analytics. Retrieved from https://www.scottmadden.com/insight/combating-turnover-using-people-analytics/

Solutions, P. (2025). Payroll data retention management: How to do it right. Retrieved from https://platform3solutions.com/blog/solution-for-keeping-payroll-records-data-retention-compliance/

WJARR. (2025). Developing scalable hr analytics platforms for smes with scd2. Retrieved from https://journalwjarr.com/sites/default/files/fulltext_pdf/WJARR-2025-2920.pdf