**Machine Learning Using Deep Learning and Natural Language Processing**

**Peraton**

**www.peraton.com**

Architecture

**Date:**

10/08/2021

**Team Members:**

Rolando Gonzalez

Elshaday Alemayehu

Yusuf Siddiqui

Arial Carvalho

Eduardo Heredia

# Abstract

This document details the agreement between the sponsor and the development team. The architecture style and model of our system represent the design decisions related to overall system structure and behavior. The defined architectures will help stakeholders understand and analyze how the system will achieve essential qualities such as modifiability, availability, and security. Explicit description of software architecture provides a foundation for understanding and reasoning about the functionality and quality of our software system at the very early stages in the development process as well as a high level of abstraction. In this deliverable, we will see the formulation of our architecture style used and how it supports various features of our application. This document is not final and is subject to change as needed by the sponsor.

The architectural styles used define the structure of the system we intend to build for the sponsor. It focuses on how we intend to organize the project and the relationship between the various layers of the project. The architectural model presents a visual depiction of this organization. The technology, software, and hardware used show what products and libraries are utilized by the team to develop this project. The rationale indicates the reasoning of the development team’s though process when choosing the architecture pattern.

Table of Contents

[Abstract 2](#_Toc84593634)

[List of Figures 4](#_Toc84593635)

[Architectural Styles Used 5](#_Toc84593636)

[Three-tier Architecture Pattern 5](#_Toc84593637)

[Data Access Object (DAO) Design Pattern 5](#_Toc84593638)

[Façade Design Pattern 5](#_Toc84593639)

[Architectural Model 6](#_Toc84593640)

[Technology, Software, and Hardware Used 7](#_Toc84593641)

[Rationale for Your Architectural Style and Model 8](#_Toc84593642)

[Evidence the Document Has Been Placed Under Configuration Management 9](#_Toc84593643)

[References 10](#_Toc84593644)

# List of Figures

[Figure 1: Three-Tier Architectural Model 6](#_Toc84592814)

# Introduction

The Architecture Document provides an architectural overview of the project. The document describe the architectural style, such as Three-Tier Architectural Patterns, Data Access Object (DAO) Design Pattern, and Facade Design Pattern, the architectural model, and the technology, software, and hardware used to support the project. The purpose of the documentation is to ensure there’s a qualitative view of the project and provide measurable criteria to meet.

The Architecture Document structure is as follows:

1. The architectural styles implemented that’ll contribute to the communication, user interface, and visibility of the system.
2. The architectural model to provide a high level visual model of the architecture.
3. The technology, software, and hardware used to implement the project
4. The rationale supporting the architectural style and model implemented.

# Architectural Styles Used

Architectural pattern is a design strategy that contributes to the communication, documentation, and the software quality attributes. For this project, we have selected the following architectural patterns:

## Three-tier Architecture Pattern

The three-tier architecture design divides the software into three layers: Presentation, Application, and Data tier. For this system, there will be a dashboard that provides several options for the user. These controls are implemented by the back-end architecture, which is hidden from the presentation tier. For any data that needs to be persisted, the data tier will be used to store information in the database.

The presentation tier is the user interface and communication layer of the application, where the end-user interacts with the application. Its main purpose is to display information to and collect information from the user. This top-level tier can run on a web browser (UI) that will send content to a browser that will allow client communication.

In the application layer, information collected in the presentation tier will be processed against other information in the data tier using relevant business logic and a specific set of business rules.

The Data Layer will be used as the back-end where the information processed by the application is stored and managed. This will be storage such as a SQL database management system that provides access to application data.

## Data Access Object (DAO) Design Pattern

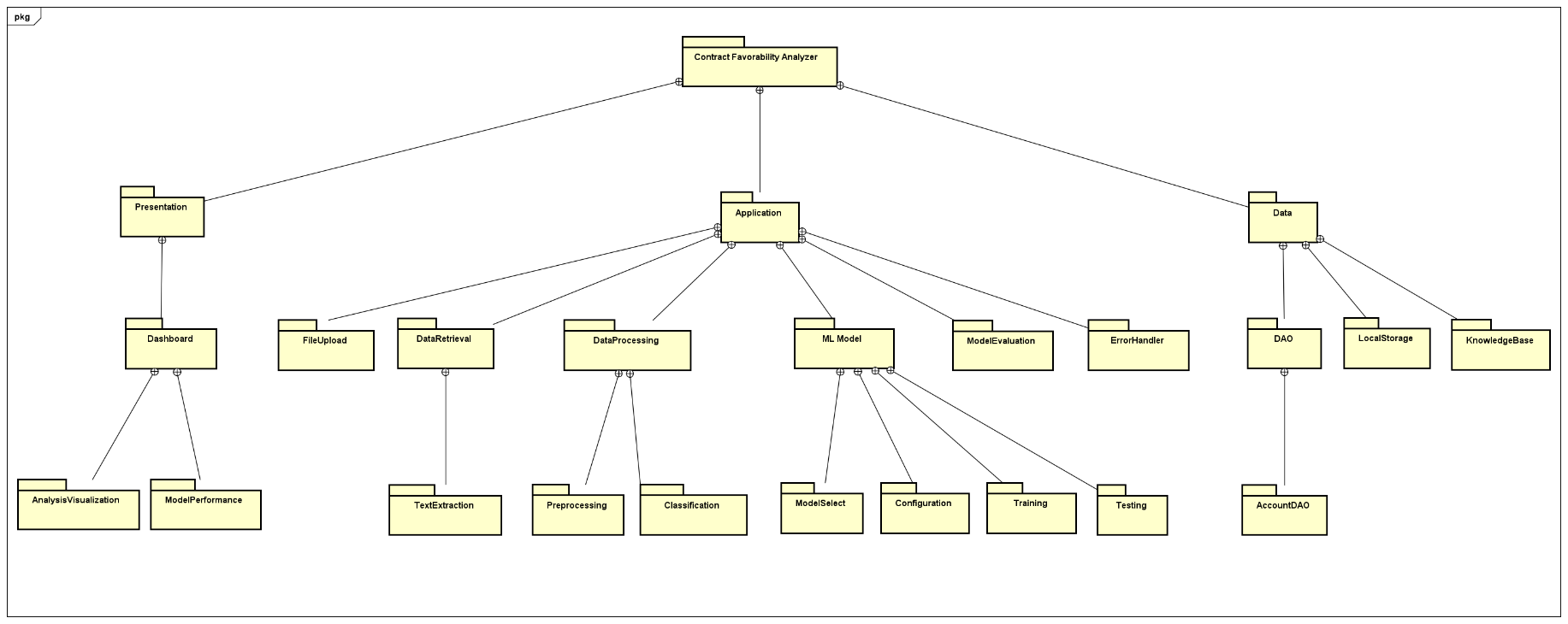
The DAO pattern is a structural pattern that isolates the operations (Create/Retrieve/Update/Delete) on the data from the business tier. For this system there will be user’s data to be stored on the database. The DAO class will handle any incoming/outcome to the database..

## Façade Design Pattern

The Facade pattern hides the complexity and simplifies the client’s view of a service. This design helps in the understanding of the architecture workflow

# Architectural Model

Figure 1: Three-Tier Architectural Model



# Technology, Software, and Hardware Used

**Hardware:** Windows laptop environment with Wi-Fi connectivity

**Software:** Python, Anaconda, Jupyter Notebook

* All team members will be operating in a Windows environment to avoid conflicts
* We will be using Python as it is suitable for collaborative implementation within our team members. Since Python is a general-purpose language, it can do a set of complex machine learning tasks and enable us to build prototypes quickly that allow us to test our product for machine learning purposes. Python also has a lot of ML and NLP  frameworks and libraries that can be suitable for our project.
* We will also be using Jupyter notebooks for all sorts of data tasks including data cleaning and transformation, data visualization, machine learning, deep learning, and NLP.

**Text Extraction:**

* Python-docx: python library that will be used for extraction words in paragraphs from Microsoft Word documents [1].
* PyPDF2: python library that will be used to extract words and document information from PDF documents [1].

**Data Processing**:

* Natural Language Toolkit (NLTK): python library that will be used for natural language processing on extracted text to clean and prepare the data through tokenization and the removal of ‘stop words’ [2].
* Scikit-learn: python library that provides tools for classification. It provides various unsupervised and supervised learning algorithms that may be used such as support vector machines[4].
* Pandas: open source python package for data processing and analyzing data [3].
* MongoDB - open source cross-platform database that uses no-SQL design and stores entries in XML/JSON format.
* PyMongo - Python framework to connect to MongoDB.

**Communication between application server and database server:**

For accessing the dashboard, the user will have to verify their credentials which are hashed and stored in a database. We will have an account DAO which will allow for creation, updating, and deleting of user accounts in the database.

For our knowledge base we will have some neural network based classification for the data so we can determine words that have some sort of favorability aspect and then add them to our knowledge base in the database. We can then use our knowledge base and context to predict whether the word is favorable or unfavorable.

# Rationale for Your Architectural Style and Model

We will be using the three-tier architecture pattern to divide the work based on the skill sets of team members and because of the ease of scalability and maintainability aspect provided with the architecture pattern. This architecture pattern allows us to quickly redesign the front-end of the application based on the needs of the sponsor.

Three-tier architecture improves data integrity. Since all the updated information goes through the second tier, in which the second tier can ensure that only important information is allowed to be updated in the database and the risk of unreliable client applications corrupting information is removed. This will help improve security as clients will not have direct access to the database.

The Database Access Object (DAO) design pattern will allow us to have an interface that can perform CRUD (create, update, retrieve, delete) operations making the account implementation simple.

# Evidence the Document Has Been Placed Under Configuration Management

Graphical user interface, text, application, email

Description automatically generated

# References

1. A. Sweigart, “Chapter 13 – working with PDF and word documents,” *Automate the Boring Stuff with Python*. [Online]. Available: https://automatetheboringstuff.com/chapter13/. [Accessed: 08-Oct-2021].
2. “Natural language toolkit,” *Natural Language Toolkit - NLTK 3.6.3 documentation*. [Online]. Available: https://www.nltk.org/. [Accessed: 08-Oct-2021].
3. “Pandas,” pandas. [Online]. Available: https://pandas.pydata.org/. [Accessed: 08-Oct-2021].
4. “scikit-learn,” scikit-learn. [Online]. Available: https://scikit-learn.org/stable/. [Accessed: 08-Oct-2021].